TAILORED TRACEABILITY AND PROVENANCE DETERMINATION IN MANUFACTURING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Adam Dachowicz

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Jitesh H. Panchal, Co-Chair

School of Mechanical Engineering

Dr. Mikhail Atallah, Co-Chair

Department of Computer Science

Dr. Karthik Ramani

School of Mechanical Engineering

Dr. Ilias Bilionis

School of Mechanical Engineering

Approved by:

Dr. Nicole Key

School of Mechanical Engineering Associate Head for Graduate Studies

For Mom, Dad, and Alyssa.

ACKNOWLEDGMENTS

It is impossible to thank everyone who has supported me through grad school, but I am going to try my best. First and foremost, I must thank my loving and patient family, including my Mom, Dad, sister, Grandma, Aunt Sue, Chris, and everyone else back home. You all have provided so much love and support over the years, and I can never repay you. I would not be where I am today without your lessons in perseverance, courage, and kindness.

I am extremely grateful to my advisors, Dr. Jitesh Panchal and Dr. Mikhail Atallah, for their guidance and support throughout my graduate studies. Dr. Panchal, your constant scientific, professional, and personal guidance have been my rock since day 1 of grad school. You are a fantastic teacher and mentor, and your students will be forever thankful. Dr. Atallah, you have taught me so much throughout my time at Purdue, both technically and professionally. Thank you for helping this ME student branch out into CS, and for all your insightful feedback and suggestions for my work. Every meeting with you both is a valuable learning experience, whether the topic is cryptography, steganography, economics, paper writing, patent writing, philosophy, or one of hundreds of other subjects. Sincere thanks as well to my committee members, Dr. Karthik Ramani and Dr. Ilias Bilionis, for your continued support and feedback throughout my studies. I am also deeply indebted to my undergraduate advisors, Dr. Farrokh Mistree and Dr. Janet Allen, and the members of the Systems Realization Laboratory @ OU, for nearly a decade of mentorship, advice, and guidance. Without your support, my grad school journey would never have happened.

Thank you to all members of the Design Engineering Lab @ Purdue. Ashish Chaudhari and Murtuza Shergadwala, you two have been a constant from my first day at Purdue to our back-to-back defenses, and I will forever treasure our friendship. I have always been able to turn to you two whenever I needed help in lab or in life.

Lessons from our conversations about research, careers, the future, and life in general will stay with me long after grad school. Siva Chaduvula, I have grown so much through collaborating with you, and I will always consider you a close friend and mentor. Thank you as well to Sharmila Karumuri, Karim ElSayed, and Atharva Hans, and past lab mates Joseph Thekinen, Parth Paritosh, Piyush Pandita, Naman Mandhan, and Zhenghui Sha. It is a pleasure to work with, learn from, and hang out with all of you. I will keep the memories of DELP with me forever. I must also thank my friends outside Purdue for keeping me sane these past few years. I am especially thankful to Mark Roundtree and Austin Savala. Our late-night conversations and gaming and D&D sessions provide a constant source of joy, and grad school would have been impossible without your support.

Finally, I would like to thank all my professors in Mechanical Engineering, Computer Science, Materials Science, Mathematics, and Electrical Engineering for your patient teaching and for helping me grow as an engineer and researcher. I would also like to acknowledge everyone else at Purdue and the School of Mechanical Engineering that make graduate study possible, especially the incredibly patient grad office staff. I also appreciate the support provided by the Purdue Graduate School Doctoral Fellowship program, as well as NSF grant CMMI 1329979. Thank you again to everyone I mentioned here and to those I missed. None of this was possible without you.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABSTRACT

Dachowicz, Adam Ph.D., Purdue University, August 2020. Tailored Traceability and Provenance Determination in Manufacturing. Major Professors: Jitesh H. Panchal, Mikhail Atallah.

Anti-counterfeiting and provenance determination are serious concerns in many industries, including automotive, aerospace, and defense. These concerns are addressed by ensuring traceability during manufacturing, transport, and use of goods. In increasingly globalized manufacturing contexts, one-size-fits-all traceability solutions are not always appropriate. Manufacturers may not have the means to re-tool production to meet marking, tagging, or other traceability requirements. This is especially true when manufacturers require high processing flexibility to produce specialized parts, as is increasingly the case in modern supply chains. Counterfeiters and saboteurs, meanwhile, have a growing attack surface over which to interfere with existing supply chains, and have a leg up when implementation details of traceability methods are widely known. There is a growing need to provide solutions to traceability that i) are particularized to specific industrial contexts with heterogeneous security and robustness requirements, and ii) reliably transmit information needed for traceability throughout the product life cycle.

This dissertation presents investigations into *tailorable traceability schemes* for modern manufacturing, with a focus on applications in additive manufacturing. The primary contributions of this dissertation are frameworks for designing traceability schemes that i) achieve traceability through recovery of manufacturer-specified signals, from simple identity information to more detailed strings of provenance data, and ii) are tuned to maximize information carrying capacity subject to the available data and intended use cases faced by the manufacturer.

In the vein of physically unclonable function (PUF) literature, these frameworks leverage the intrinsic information present in material structure, such as phase or grain statistics. These structures, being functions of largely random and uncontrollable physical and chemical processes, are by their nature uncontrollable by a manufacturer. According to the frameworks proposed in this dissertation, anti-counterfeiting and traceability schemes are designed by extracting large libraries of features from these properties, and designing methods for identifying parts based on a subset of the extracted features that demonstrate good utility for the present use case. Such schemes are customized to handle specific material systems, metrology, expected part damage, and other concerns raised by a manufacturer or other supply chain stakeholders.

First, this dissertation presents a framework that leverages this intrinsic information, and models for damage that may occur during use, for designing schemes for genuinity determination. Such schemes are useful in contexts like anti-counterfeiting and part tracing. Once this framework is established, it is then extended to design schemes for dynamically and securely embedding manufacturer-specified messages during the manufacturing process, with a focus on implementation in additive manufacturing. Such schemes leverage both the intrinsic information inherent to the material / manufacturing process and extrinsically introduced information. This extrinsic information may include cryptographic keys, message information, and specifications regarding how an authorized user may read the embedded message. The resulting embedding schemes are formalized as "malleable PUFs."

The outcome of these investigations are *frameworks for designing, evaluating, and implementing traceability schemes* that can be used by manufacturers, academics, and other stakeholders seeking to implement secure and informative traceability schemes subject to their own unique constraints. Importantly, these frameworks can be adapted for a range of industrial contexts, and can be readily extended as new methods for in-situ measurement and control in additive manufacturing are developed.

1. INTRODUCTION: THE TRACEABILITY PROBLEM

Counterfeiting, in a general sense, is the business of producing "fakes." The counterfeiting trade is as old as the idea of personal and intellectual property (IP). From fake ancient Roman denarii [1] to misrepresented wine to fake designer handbags, stealing the looks, name, and reputation of one product for the benefit of another is a common criminal business practice.

In the modern era, counterfeiting is a growing problem in many markets. A growing body of literature has developed to study the issue of counterfeiting, and to better understand its causes, operation, and weaknesses. The most commonly cited reasons for increasing counterfeit activity include: the rising prevalence of multinational corporations and geographically disparate product development teams [2], the increasing ease of international trade, the rise of e-commerce, and increasing societal importance of complex supply chains with huge infiltration attack surfaces [3–6]. These factors contribute to a global environment that, at best, makes the job of legitimate vendors looking to protect their IP more difficult, and at worst encourages IP infringement as a reliable business strategy among a sizable population of potential suppliers. Even if counterfeiting becomes a non-issue, the task of ensuring the correct goods are being used in the correct way can be challenging. In the context of emerging manufacturing technologies, such as additive manufacturing, where potentially every part is custom-made for a specific use case, assuring this correctness becomes more pressing.

Observation of IP-related trends in several high-tech sectors gives evidence for these concerns, although the exact impact can be difficult to correctly capture and corroborate. The electronics industry is estimated to lose about $100 billion in sales per year to counterfeit products [7], while as of 2011 the American automotive industry was estimated to introduce $3 billion worth of counterfeit components into service [8] through supply chain infiltration and after-market repair. In the Ameri-

can defense industry, the number of electronics systems compromised by counterfeits has been on the rise since 2005, reaching almost 10,000 confirmed cases in 2010 and prompting the Senate Armed Services Committee and the Department of Defense to formally investigate the issue [9].

The numbers above are estimates, of course, as hard statistics regarding the extent of counterfeiting are hard to come by. More certain statistics come from law enforcement agencies tasked with enforcing existing IP law. For instance, according to the International Anticounterfeiting Coalition, a Washington DC-based international organization of over 250 member IP holder businesses and agencies, approximately $1.2 billion in IP-infringing goods was seized during 55.7 million seizures conducted by US homeland security in 2014 alone [10].

These trends are worrying, and beg several questions. For instance, industries facing increasing rates of counterfeit part infiltration should seek to understand the forces contributing to the problem, both within their own organizations and in broader economic, national, and international terms. How should we model the issue, and what should such models capture about a given supply chain and the actors and forces involved? How are existing remedies performing, andards answers how might these be tweaked to produce better results? What solutions in anti-counterfeiting and traceability make sense for today's highly globalized manufacturing ecosystem, and what solutions will fit with accelerating technologies like additive manufacturing? I invite the reader to consider the last question intently, as this dissertation serves as a modest step towards achieving those solutions.

1.1 Traceability, Counterfeiting, and the Goals of this Dissertation

Specifically, in this dissertation techniques are presented to address the challenge of counterfeiting, and *traceability* more generally, in modern (global, highly complex) manufacturing supply chains. This is done through two primary investigations: First, *traceability through leveraging intrinsic information* is considered through the case

studies and the resulting traceability schemes presented in **Chapters 3 through 5**. Second, *traceability through secure embedding schemes for additive manufacturing* is considered through exploration of the concept of malleable physically unclonable functions (PUFs) presented in **Chapters 6 and 7**. The outcome of both investigations are *frameworks for designing, evaluating, and implementing traceability schemes* that can be used by manufacturers, academics, and other stakeholders seeking to implement secure, informative traceability schemes subject to their own unique constraints. With the existing literature given proper context in **Chapter 2**, a thorough discussion of the research questions guiding these investigations is provided towards the end of this chapter. The goals of each chapter are presented graphically in Figure 1.1.

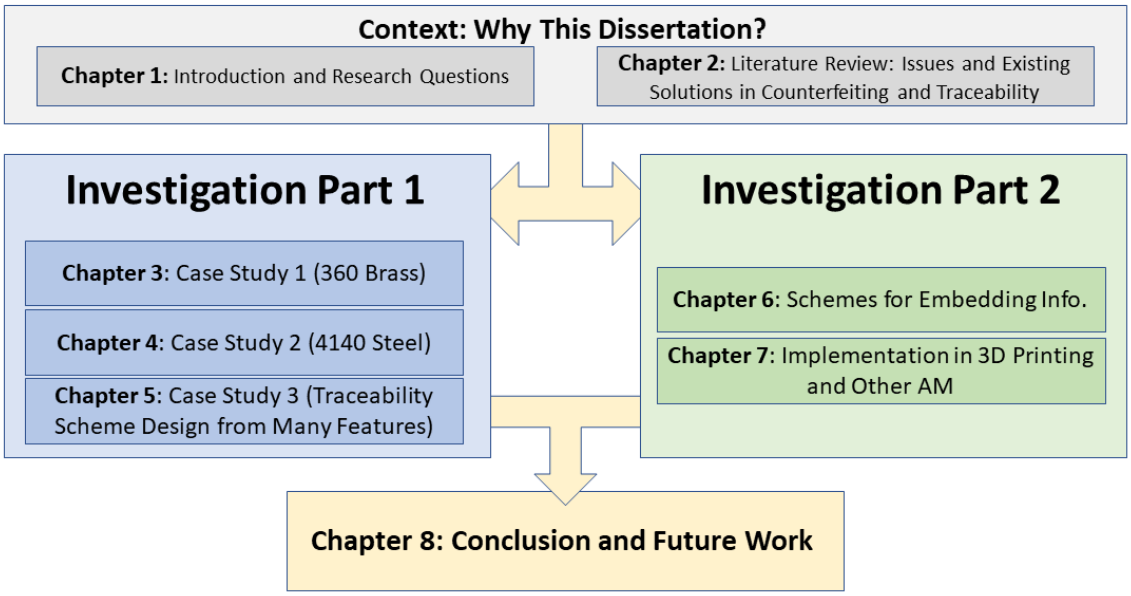

Fig. 1.1. Outline of this dissertation.

To motivate the research presented in this dissertation, **this chapter** and **Chapter 2** attempt to introduce fundamental issues in traceability by examining the problem of counterfeiting, and to present a critique of current anti-counterfeiting best practices. Chapter 2 also introduces some modest proposals for increasing complex

supply chain resilience to counterfeit part infiltration. As an *illustration* of the complex problems facing modern supply chains, counterfeiting is particularly well-suited: the success of counterfeiting attempts are impacted by decisions made by part designers, manufacturers, shippers, buyers, and maintainers; many vulnerabilities exploited by counterfeiters may also be exploited by Nature (through chance, human error, bad weather, and so on) to prevent proper traceability; and counterfeiting itself is an intuitive idea with identifiable impacts on society. Much of the work presented in this dissertation, especially work presented in **Chapters 3 through 5**, attempts to address counterfeiting explicitly, and as Chapters 1 and 2 attempt to make clear, counterfeiting is indeed a serious issue facing many important industries.

However, I do *not* wish to imply anti-counterfeiting is the only, or even the main, issue in traceability in manufacturing. Challenges exist across the manufacturing and supply chain pipeline, from credit and blame attribution in design, to product and product metadata tracking, to ensuring the correctness of parts in an assembly, to tracking maintenance and end-of-life data. These challenges span the fields of cybersecurity, cyber-physical systems security, additive and subtractive manufacturing, materials science, logistics, and design. Indeed, given recent advances in additive manufacturing (AM) and its growth in the automotive [11], aerospace [12], and medical industries [12, 13], there is significant opportunity to connect many topics in these fields to "bake in" notions of security and traceability from the very first steps of the design process, so that they carry throughout the entire product life cycle. These AM-flavored ideas are discussed in more detail in **Chapters 6 and 7**. If you are already convinced these issues exist and are worth studying, feel free to skip to the end of this chapter for a more thorough discussion of the goals and research questions guiding this dissertation.

The remainder of this chapter is organized as follows. First, definitions of "traceability" and "counterfeit parts" are presented, in order to scope the discussion in later chapters. Then, some closing thoughts on the motivation for tailored traceability in modern manufacturing are provided.

1.2 Defining "Traceability" and "Counterfeit"

Since "traceability" appears in the title of this dissertation, yet is underlined in red for quite a few spell-checking apps, I first offer a definition for this concept that will follow throughout this dissertation. Interestingly, though for public health reasons perhaps unsurprisingly, the definition of traceability sees the most high-volume discussion in agriculture and food supply chain literature. Good traceability in agriculture is needed to comply with various governmental regulations on food safety, and to assist with recalls, studies, and other logistical concerns facing the industry [14–16], but is not the focus of this dissertation, so I mention it here only for completeness.

In manufacturing, product "traceability" is often understood either i) as tracking material and processing *while that material is being used to manufacture goods*, not considering information before (source of raw materials) or after production (movement of goods to customers and beyond), or ii) as tracking chain of custody and processing information as a part moves through a supply chain. The former, "internal traceability," often applies to one process or production component and may actively be used to track or even control part quality if traceability data is of high enough resolution [16–18]. The latter is referred to as "chain traceability," and as set by the ISO 9001:2000 standard requires the ability to "trace the history, application or location of an entity by means of recorded identifications throughout the entire supply chain" [16, 17, 19].

Definition: While the literature provides many definitions for traceability, particular to the resolution (internal or chain) and the industry, throughout this dissertation I will use the following:

> *"Traceability" is the ability to track a product's chain of custody during production, distribution, use, and disposal, and the ability to recall the movement and processing of the product throughout its history.*

Things change somewhat if the focus narrows to the particular traceability issue of counterfeiting. Ideas about what constitutes a "counterfeit" good cover a broad range

of artefacts and can be, perhaps surprisingly, difficult to formalize. The definition is sensitive to existing notions of intellectual property, assumptions about what constitutes a "fake" among a specific population of products (a "counterfeit" wine may still be enjoyed by the majority of the population, after all), and to some extent the attitudes of the IP holder and the target market. In the case of the complex supply chains considered in this dissertation, and restricted in this chapter and Chapter 2 to those of the automotive and American defense industry for consistency and simplicity, the target market tends to be other members of the chain (i.e., business-to-business or "B2B" interaction).

In an attempt for consistency in the following chapters, I will construct a definition of "counterfeit" that captures the most relevant features of "counterfeit" goods entering these supply chains. This definition is lightly paraphrased from Merriam-Webster [20] and expanded to include other useful concepts discussed in literature. In particular, Staake and coauthors [4] propose a strong, detailed definition of counterfeiting with respect to unauthorized signaling of a source, quoted here: *"... goods that, be it due to their design, trademark, logo, or company name, bear without authorization a reference to a brand, a manufacturer, or any organization that warrants for the quality or standard conformity of the goods in such a way that the counterfeit merchandise could, potentially, be confused with goods that rightfully use this reference"* [4]. This definition is good, but (at the risk of throwing stones from a glass house) a bit long-winded, and does not explicitly include parts from an original source that may be out-of-specification.

An alternate three-part definition for use in this dissertation is proposed below:

Definition: *A "counterfeit" part, in the context of parts infiltrating a manufacturing supply chain, is:*

1. *a fraudulent imitation of another part knowingly signaling incorrect origin, or*
2. *an object knowingly represented as having properties that it does not possess, regardless of origin, or*

3. an object otherwise sold in violation of IP laws.

Note that Points (1) and (3) are intended to capture the intuitive notion of counterfeit and Staake's IP-intensive definition, while Point (2) is intended to capture any parts, regardless of the represented origin, that are knowingly introduced to the chain with unknown or inferior performance characteristics. This definition captures all of the counterfeiting scenarios discussed in this dissertation, although given the bulk of literature it is not exhaustive.

1.3 The Case for Tailored Traceability

Finally, I do not want to end this introductory chapter without highlighting the word "tailored" in "tailored traceability." As I show in **Chapter 2**, traceability is not a new field, and plenty of researchers have devoted significant time to addressing internal and chain traceability across many industries. Literature in anti-counterfeiting, especially with respect to technical solutions like physically unclonable functions (or PUFs, see Chapters 2 and 3), leverage the intrinsically random-like behavior of material systems for determining uniqueness. In such a paradigm, the fact that humans have little to no control over the specific features of interest of the material is a feature, for it is this unique response that serves as a mark of genuinity. However, such a system requires an explicit design that may be costly or inappropriate for a given manufacturing process.

One could also consider the task of information hiding, as discussed in steganography literature. In this paradigm, the user has complete control over the information being written into the artefact (in literature, almost always some file like an image or video), and relies on the fact that an adversary either does not know a message is being hidden, or if they do, that they cannot decrypt the message without additional information like a key known only to authorized users.

In this dissertation, I take the view that a *combination* of these two paradigms is needed for anti-counterfeiting and traceability. This is especially true in areas where

parts are manufactured with a high degree of customization, as is the case in additive manufacturing. By leveraging the *intrinsic* information gifted to us by physics as essentially natural keys (for instance, microstructural information like phase distributions or grain statistics in metals), we can design strong schemes for securely transmitting some *extrinsic* information regarding the knowledge we as manufacturers care about (identity, processing information, customer information, and so on). If the framework for such an approach is sufficiently general, then these schemes can be tuned to the most appropriate intrinsic information and the most extrinsic embedding method for the manufacturing scenario, hence *tailored traceability*.

To justify this, I demonstrate that i) such intrinsic information actually exists and can be tailored for traceability in specific use cases, and ii) that this information can be leveraged in processes like additive manufacturing for tailored information embedding schemes for traceability. I address point i) in **Chapters 3 through 5**, and ii) in **Chapters 6 and 7**. The specific research questions guiding this investigation are given in the following section.

1.4 Frameworks for Traceability Schemes: Research Questions

The primary question driving this dissertation is:

> *How can manufactured goods be traced throughout the product lifecycle with minimal to no impact on existing manufacturing and procurement processes?*

The first research question addresses the construction of tailored traceability schemes that leverage a manufactured part's intrinsic features:

RQ1 : **Traceability Framework:** What data derived from manufactured goods, when collected and processed, yields features that can be used to achieve robust traceability?

This data must come from some measurement, and this measurement must be carried out both at time of manufacture and at some time downstream, when the good's genuinity is under question. Further, "robust" traceability must both i) be robust to corruption from expected wear-and-tear or other damage taken during transport or use, and ii) have high enough information carrying capacity to transmit the desired signal. The information channel is subject to complex operations that introduce noise through damage after manufacture, and perhaps even corruption during manufacture introduced through imperfect data acquisition. The channel itself, encompassing the data being measured, the measurement equipment, the data processing steps, and all relevant manufacturing parameters, must be carefully designed to account for this corruption. In this dissertation, optical data is used as the primary information source.

The secondary research questions in support of RQ1 are therefore:

RQ1.1 : What optical data may be leveraged for information carrying capacity?

RQ1.2 : How can features constructed from this data be evaluated for robustness in the context of manufacturing?

RQ1.3 : How can features constructed from this data be evaluated for information carrying capacity?

If the above RQ's are answered, then those answers can be channeled directly to generate a **traceability framework** for designing traceability schemes built on these intrinsic signals.

The second research question addresses the construction of tailored traceability schemes that combine these unique, intrinsic features with some extrinsically introduced message to embed information in additive manufacturing:

RQ2 : **Embedding Framework:** Given an existing additive manufacturing process, how can information be embedded reliably and securely for identification and message transmission?

Attacking this question requires a few additional pieces of information. For "reliable" and "secure" transmission, there must be some source of random-like information that could be used as some sort of key. In fact, the intrinsic features discussed in the previous RQ may be excellent candidates for this. Then, there will need to be some way of intermingling this intrinsic information with some extrinsic message to perform the embedding, subject to processing parameter constraints, information carrying capacity constraints, and robustness requirements.

The secondary research questions in support of RQ2 are therefore:

RQ2.1 : How could intrinsic features of manufactured goods be used to reliably embed extrinsic information during manufacturing?

RQ2.2 : How can additive manufacturing processes be used or modified to perform information embedding in parts?

RQ2.1 is primarily focused on the formulation of the embedding schemes and the **embedding framework** for their design. RQ2.2 assumes the existence of such schemes, or a framework for their design, and instead focuses on implementation.

The remainder of this dissertation represents my investigation of these questions. RQ1 is addressed through the investigation of anti-counterfeiting schemes for a variety of material systems. These are presented in Chapters 3 through 5. RQ2 is addressed through the investigation of tailorable schemes for information embedding in additive manufacturing. These are presented in Chapters 6 and 7. By investigating these RQs, frameworks for designing tailored traceability schemes incorporating intrinsic and extrinsic information can be developed for future use. The specific research questions addressed by each of the remaining chapters of this dissertation are presented in Figure 1.2.

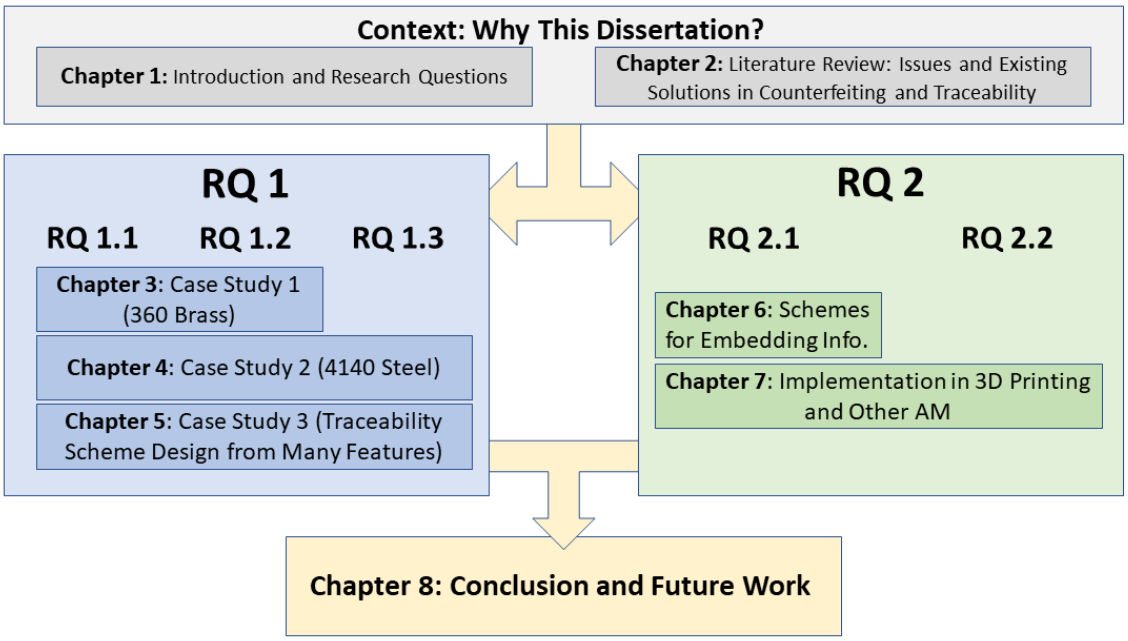

Fig. 1.2. Outline of this Dissertation. Chapters 3-7 are each mapped to the respective Research Questions investigated.

2. LITERATURE REVIEW: ANTI-COUNTERFEITING AND TRACEABILITY IN MANUFACTURING

This chapter aims to provide a literature review to set the proper context for the work presented in this dissertation. This review is broken into four sections. The first presents a discussion of two relevant supply chains that are highly vulnerable to physical part counterfeiting attempts, namely global automotive supply chains and the US Department of Defense (DOD) supply chain. The second section discusses existing approaches to modeling industrial counterfeiting operations. The third section discusses legal, organizational, and technical strategies to mitigate the risk of counterfeit part infiltration based on these models, and modest recommendations regarding these strategies. The final section discusses physically unclonable functions (PUFs) and related technologies, and their implications for tailored traceability. The following chapters of this dissertation build on the ideas discussed in this section, and are aimed at furthering the work in the field of anti-counterfeiting and traceability. These goals are discussed in greater detail in Chapter 1.

Countermeasures for counterfeiting, in all contexts, generally consider legal, organizational, economic, cultural/social, or international factors contributing to the counterfeiting business. Effective strategies, intuitively, should consider all of the above concerns when applied to international markets and supply chains. Where actors are incapable of addressing all these concerns at once, cooperation (often through intergovernmental or international business organizations) is occasionally used, although in carrying out this review it seems that cooperative approaches are less common than may be necessary. Existing strategies are discussed at several organizational levels, from individual companies to international organizations. Technological remedies are also discussed.

In the end, this chapter is intended to provide context for today's growing industrial counterfeiting problem, and give a survey of the surrounding issues, literature, and best practices. This topic represents a highly active field of research, and as the body of knowledge grows we should look forward to better IP protections and the increased innovation to follow.

Classifying Types of Counterfeit Parts. Before continuing with this review, it may be helpful to scope the problem by expanding on specific examples of "counterfeit" parts discussed in related literature. Guin and coauthors [7] discuss seven kinds of counterfeit parts, which may be further segregated into four categories for the purposes of this chapter: salvaged parts, taken or stolen goods, sabotaged goods, and illicitly produced goods. These categories capture all of the relevant illicit parts discussed in this chapter. These categories, and the original counterfeit types presented by Guin, are summarized in Figure 2.1.

Recycled	→ Salvaged after-market
Remarked	→ Taken from source and sold illegally
Overproduced	
Out-of-Specification	
Tampered	→ Sabotaged part
Cloned	→ Produced by illicit manufacturer
Forged Documentation	

Fig. 2.1. The seven counterfeit types presented by [7], segmented into four broad counterfeiting categories.

2.1 Industrial Counterfeiting and Relevant Supply Chains

In this section, two large industries with highly complex, global supply chains are discussed: the automotive industry in general and the American defense industry, under the umbrella of the US Department of Defense (DOD). This discussion is intended to provide additional real-world context to the issue of counterfeit manufac-

tured goods and to motivate the following discussion of modeling and mitigating the counterfeiting problem.

2.1.1 The Automotive Supply Chain

The automotive industry remains one of the larger sectors in American manufacturing. Over 17 million automobiles were sold in the US, with American companies General Motors and Ford taking the first- and third-largest market shares, respectively (Toyota took the second-largest) [21]. the industry has enjoyed nearly a century of technological development, and modern cars and trucks are without exception assemblies of thousands of components. The industry in general relies heavily on a large number of small, simple-looking parts, as well as specialized electronics components. These fall under the umbrella of "high-tech-low-cost" parts that are particularly vulnerable to cloning by an illicit manufacturer [4] attempting to ride off the reputation of another without providing the expected performance guarantees.

In the 21st century, globalization has allowed for better economies of scale: specialized manufacturers may now focus on a small subset of these components, perhaps even just one, to produce at high quality and very low cost. This, in turn, allows automotive manufacturers to drive development costs down and may enhance flexibility when sourcing components as the supply chain continues to grow. As an example, according to Simchi-Levi and coauthors [22], Ford's network of suppliers and suppliers-to-suppliers has expanded over four continents. This supply chain development has become increasingly important as markets become more competitive, and customers demand better customization.

Large supply networks are further encouraged to expand by the wide-spread adoption of lean or "just-in-time" manufacturing thinking within automotive and other manufacturing companies. In fact, automotive manufacturing company Toyota pioneered the concept in its production before widespread adoption. Melton [23] provides a good summary of the concept of lean manufacturing; the main idea is to leverage

advances in logistics, technology, supply sourcing, and processing to reduce as much waste as possible during production. In the context of supply chain management, this often takes the form of sourcing parts in small batches (compared to volumes necessary for mass production) as cheaply as possible, hence the preference to large and robust supply networks.

This method has clear benefits: it reduces labor, material, and other overhead costs that may then be passed on to the customer, and encourages competition among existing suppliers to lower cost and increase quality. However, in the context of counterfeit activity, downsides may include difficulty in tracking the original location of parts through the chain (traceability), difficulty enforcing quality standards, and reduced control over the production of outsourced sub-systems. Again, turn to Ford for an example: A 2007 disclosure states that Ford estimates about $1 to 2 billion in losses per year to counterfeit parts [24]. These costs are introduced from repairs and warranty costs, recalls, and after-market infiltration through routine repair work. Modeling and mitigating such illicit activity will be discussed in the following sections.

2.1.2 The US Department of Defense Supply Chain

The American Department of Defense (DOD) supply chain faces many of the same challenges listed in the automotive industry, as highlighted in a 2013 report[1] on the issue of counterfeit components entering DOD systems [9]. For instance, the DOD supply chain is also highly globalized, with suppliers from many different nations providing critical components, design input, and sub-system manufacture and assembly services.

The major American defense contracting companies (including Boeing, Raytheon, Lockheed, and a few others) and DOD itself also employ lean manufacturing principles when necessary, such as the controversial "Lowest Price Technically Available" sourcing principle, which awards contracts to the contractor providing the best price

[1] As a side note, I highly recommend reading this report if you enjoy a good scare.

at the minimum technical guarantee [25]. This contracting method constitutes a severe risk of moral hazard among DOD contractors, especially smaller contractors with insecure membership in the DOD supply chain. Often, the DOD is a company's biggest customer, and such companies may calculate correctly that costs incurred by preventing counterfeit infiltration may not outweigh the risk * costs of being caught passing on illicit parts, especially when tracing those parts would lead investigators to another country of origin (such liability concerns, it should be noted, are not unique to DOD, and extend to any complex manufactured system). As an example, consider a 2012 investigation into counterfeit electronics, specifically semiconductors and night vision equipment, making their way onto Boeing aircraft [26, 27]. The investigation led a Senate Committee on a whirlwind tour of several states, bouncing from supplier to supplier, until eventually the trail led to China, where the effort was stymied due to a lack of cooperation from Chinese officials. Although the DOD has increased requirements on vetting incoming components and reporting suspected counterfeits or out-of-specification parts, these standards are hard to enforce and make it harder for licit suppliers to compete with those who spend less on ensuring quality sourced components.

As in the automotive industry, the 2013 report highlights in particular e-commerce, large international supply chains, disruption caused by global economic upheavals (mainly during the 2008-2009 crisis) and dependence on overseas manufacturing as major liabilities. There are, however, interesting differences that highlight additional concerns when fighting counterfeiting attempts. The first is the high level of defense contractor integration in the US. This integration of large defense contractors over the past few decades, coupled with a deep reliance on global suppliers and talent to build and maintain complex defense systems, has dramatically increased the DOD's vulnerability to counterfeit part infiltration. This trend is expanded upon in the 2013 DOD report [9], and mimics concerns raised in the automotive and electronics industry as well.

The long service life of many critical defense systems (often spanning several decades) is also a point of concern. Such systems undergo numerous cycles of use, repair, and updating. As will be discussed in the next section, common counterfeit infiltration methods involve selling illicit components to companies that provide repair or refurbishing services, where traceability is less of a concern; these components may then compromise a product during its useful lifetime. In fact, this is one concern mentioned in the recent report with regard to counterfeit electronic components entering critical aerospace devices [9].

In the following section, models of counterfeiting activity are discussed, primarily to motivate the pros and cons of counterfeitng mitigation methods.

2.2 Models of Counterfeit Activity in Complex Supply Chains

Modeling the motivations and behaviors of counterfeit enterprises has seen less study among academics and stakeholders when compared to the economic, social, and safety effects of industry-specific counterfeit attempts [4–6]. Staake and coauthors [4] highlight the difficulty in studying illicit ventures directly, which of course try to stay hidden, while Stevenson and Busby [5] cite the need for data from both the supply and demand sides of the counterfeit market to collect actionable data on the methods of counterfeiters (clearly, neither side is typically willing to cooperate, and may not even know they are complicit in illegal activity). Hoecht [6] discusses potential strategies original equipment manufacturers (OEMs) and downstream manufacturers could adopt in addressing counterfeit products from unknown sources, and recommends grounding any study of the phenomenon in the assumption that counterfeiters are rational, knowledgeable business actors responding to existing incentives in a given industry, and must be treated first and foremost as a competitor. This view is adopted in this dissertation; in all cases considered here, counterfeiters serve as suppliers within existing supply chains, and should be treated as such even if they are undesired.

The robustness of many counterfeiting operations is also a concern. As Stevenson and Busby acknowledge, counterfeiting businesses often have well-financed backers and are highly tolerant to large attrition rates. Any effective strategy must cope with the fact that completely eliminating counterfeiters from a market is close to impossible [5].

It is also important to consider the price of reputation in manufacturing supply chains. For many industrial scenarios, a supplier is not really selling bolts, or screws, or plating, or whatever other crucial and potentially life-saving equipment they list on the invoice. The real value is the source's name. That name encodes a great deal of information: manufacturer experience, customer testimonials, country of origin, environmental impact, and relations (or lack thereof) to other entities. Given a steady stream of counterfeit parts, counterfeiters' attempts to infiltrate licit supply chains revolve around this issue: *How do I convince any supplier in a supply chain that my parts come from a respected source?*

2.2.1 Adapting The Chaudry Model

Chaudry and coauthors [3] present an environment-based model for studying the counterfeit trade in a variety of scenarios. Although Chaudry's work focused on counterfeit consumer goods, the model proposed is quite general, and it is interesting to extend the concepts to other scenarios like industrial part supply chains. A modification of the model for use in this dissertation is presented in Figure 2.2. The key theme of this model is that counterfeiters should be treated as members of the supply chain that respond rationally to their environment just as licit members do.

Note that this model is broken into three major components, which loosely correspond to a point in the production cycle supply chain members are considering. The first, *Supplier Environment*, encompasses the consideration of legal, operational, and cultural concerns that pressure licit and illicit participants in a supply network to tolerate or participate in counterfeiting activity. The second, *Modes of Entry*, consider

the relative ease and difficulty of entering the supply chain, both through licit means (starting a business, earning customers, etc.) and illicit means (manufacturing or obtaining counterfeit parts, etc.). This component also involves studying infiltration methods specific to a given industry. The third, *Participant Actions*, consider the organizational, legal, and international responses to known or suspected counterfeit activity. the main concern for licit supply chain members and stakeholders is to address what can be controlled directly (white in Figure 2.2) to pressure what can only be influenced indirectly (gold in Figure 2.2).

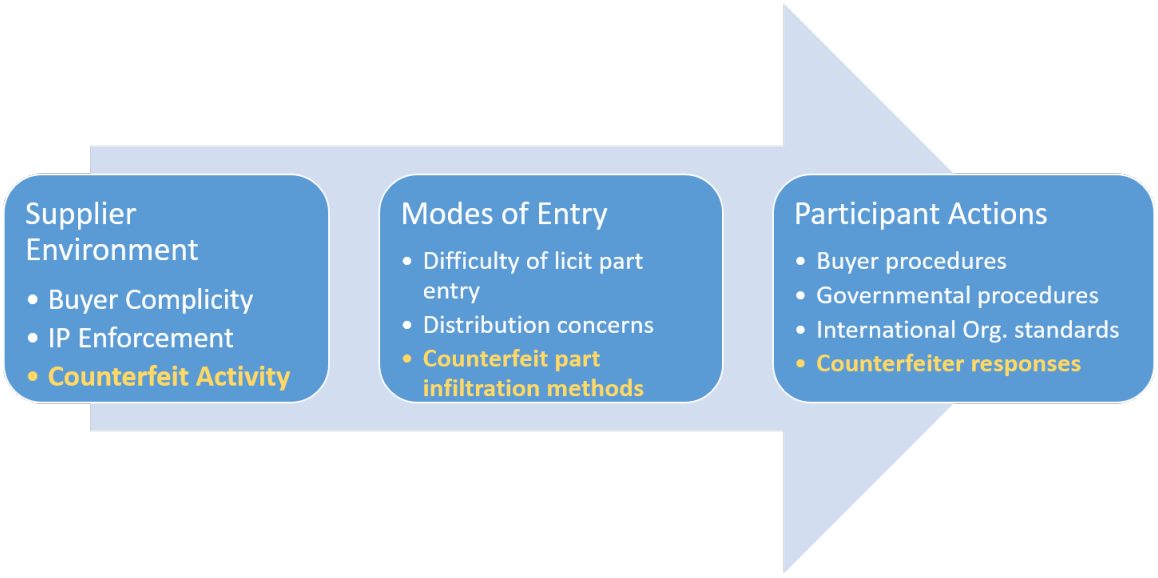

Fig. 2.2. Three-stage model of licit-illicit supplier activity in complex supply chains, modified from [3].

The following subsections expand on two critical components of this model, namely otherwise-licit members' complicity with counterfeiting activity in the supply chain and modes of entry for counterfeiters looking to infiltrate the automotive, defense, or similar supply chains.

2.2.2 Complicity in the Supply Chain

Given the high incidence of counterfeiting activity in large supply chains, it is reasonable to assume that there is at least moderate complicit behavior among otherwise upstanding suppliers, contractors, and manufacturers. This could be considered both a cultural and economic factor contributing to the prevalence of counterfeit parts in all markets, not just manufacturing supply chains. As this behavior is undesirable, and possibly illegal, there is little if any direct study of this complicity, although literature does propose reasonable hypotheses given the state of different markets [4,6,7,9]. Several of these motivations are discussed here.

Perhaps the most obvious, and most benign, reason for complicity is ignorance. The rise of e-commerce gifts counterfeiting enterprises with a large crowd of vendors to hide in, and many mid-stream suppliers may lack the resources or experience needed to properly spot illicit sellers on the Internet. Certainly, the widespread use of unvetted suppliers among DOD contractors has motivated the Senate Armed Services Committee to recommend mandatory risk detection training for American DOD suppliers [9].

Another reason: momentum to do little to nothing to prevent counterfeit part infiltration. As mentioned above, competitive contract awards common to the DOD encourage vendors and mid-stream suppliers to cut costs wherever necessary, and known difficulties in traceability lower the risk of getting caught when passing along bad parts, even if up-stream members of the supply chain catch on. This highlights another danger of large supply chains: moral hazard may become multiplied through the network, especially in cases of poor traceability or a lack of potential consequences if counterfeits simply pass from one supplier to another. If there are consequences (and that is often a big "if"), they are diluted along a large chain of hand-offs with the original perpetrator often out of reach of the interested authorities [27].

A third, more sinister motivation is to pressure competition out of the market. If counterfeiters are abstracted as simple members of a supply network that produce

goods at a very low cost, then well-established suppliers in the same business can leverage that competition to pressure competing licit suppliers out of the market. Such behavior can be waved off as the "do nothing" strategy above, or could be considered a form of co-opting as described by Hoecht and Trott [6].

Finally, it is worth noting other potential benefits of at least some counterfeit activity in a given market. Counterfeiting, especially when it involves cloning attempts, often provides a good source of work and technological development to the host region, and promotes technology transfer that may increase the rate of innovation over the long run [4]. Counterfeits introduce a low-cost low-quality alternative to licit goods, which encourages a minimum bar of quality among licit suppliers to justify their membership in established supply chains. Finally, the presence of counterfeits falsely claiming to be from your brand may, if high enough quality, increase your brand's recognition in some markets, which may help your business in the future. These must also be weighed against the costs of limiting or eliminating counterfeit activity.

2.2.3 Modes of Entry for Counterfeiters

Literature has spent comparatively more attention studying how counterfeiters enter markets and supply chains. Stevenson and Busby propose signaling as an approach to modeling counterfeiter behavior and assessing anti-counterfeiting strategy effectiveness, and provide a detailed study of existing strategies among counterfeit enterprises [5]. According to such an analysis, counterfeiters attempt to signal quality and genuity, obscure any signals of counterfeiting, or a combination[2]. By predicting such signals, thwarting attempts may be made and evaluated. On the other hand, factors like those described in the previous subsection increase the risk of missing such signals as infiltration attempts are made along the supply chain.

[2]And of course, such signaling is *exactly* the goal of well-designed traceability schemes: "I'm me! I'm not a duplicate! Nobody has tried to change me!" This is discussed in greater detail in Chapters 6 and 7, in the context of information embedding in additive manufacturing.

Stevenson and Busby describe several common tactics common in supply chain infiltration, including 1) bundling, which involves including a small population of counterfeit goods with a larger population of licit goods (common with electronics components), 2) corruption via spending significant resources in co-opting a trusted member of a supply chain, and 3) infiltrating a market through after-market services like auto repair or aerospace refurbishing [5]. Counterfeiters may also take advantage of black-swan events like massive recalls and the resulting disruption, attempting to pass on counterfeit or out-of-specification goods to unsuspecting or sloppy repair shops.

Consider the following illustrative example. In 2014, Aston Martin, a luxury car company, was forced to recall almost 18,000 vehicles manufactured since 2007, due to counterfeit materials provided by a Chinese contractor [28]. Specifically, the contractor sourced a counterfeit plastic material component critical to the vehicle's accelerator pedal from a separate Chinese source, and passed the material on to Aston Martin. With limited ability to seek damages from the Chinese companies, Aston Martin simply ate the cost of the recall and switched to a different accelerator pedal supplier in Britain. Similar cases can be found involving counterfeit Toyota airbag components installed during manufacture or (more commonly) during repair services [29], or surrounding the Takata airbag recall and several counterfeiting attempts made while car owners attempted to replace the faulty equipment [30]. Cases like these highlight 1) how difficult it can be to establish the presence of counterfeits in a complicated system (in this case, these parts were being supplied for seven years before getting caught), 2) how easy it can be for a duplicitous supplier to infiltrate an existing chain, and 3) how little consequence a compromised supplier may be expected to face (in this case, the loss of business may have been significant, but it came with no legal consequences).

In the following section, current open issues and recommendations on the topic of counterfeiting are discussed.

2.2.4 Open Issues in Counterfeiting

The issue of counterfeiting in globalized, complex supply chains is a challenging topic. So far in this chapter, features of several such supply chains were discussed, along with methods for modeling counterfeit behavior and the forces that encourage or mitigate that behavior. Literature, case studies, and existing organizational best practices suggest numerous methods exist for dealing with the issue. However, in the face of increasingly complex supply chains feeding into high-tech products, and the rising incidence of counterfeit-related issues in many supply chain contexts, it is clear that more work needs to be done. Contributions in this area should come from a variety of angles, including technical, managerial, legislative, and diplomatic.

With regard to open questions, it seems clear that more research is necessary to establish how remedies applied at different levels (organizational, technical, industry, national, international) interact with each other. This suggests a more complicated problem than perhaps some researchers realize. For instance, the best part-tagging technical solutions will be most effective only if the corresponding managerial culture accepts the practice, and the solution integrates well with the existing industrial supply chain layout. Similarly, legislators and diplomats must coordinate with managers, engineers, and businesspeople to ensure policy does not overly hamper business and innovation.

From a technical aspect, the current range competing anti-counterfeiting products suggests that there is little consensus towards the best methods for preventing counterfeit part infiltration and ensuring traceability. Trade-offs in economic feasibility, ease of supply chain integration, disruption to existing manufacturing methods, and ease of circumvention by counterfeiters deserve more study. It is also possible that the "correct" technical approaches are highly sensitive to the given industry, which would further complicate the issue.

However, this literature survey does provide room for some modest recommendations while these issues are further researched. In particular it seems that, in large

supply chains, issues of traceability and supplier trust seem to dominate instances of counterfeit part infiltration. While it is impossible to fight e-commerce now that it is so central to business, it is increasingly important to encourage smart e-commerce practices at all levels of participation. Use of industry portal sites for supplier enrollment will give companies at all levels of the supply chain more confidence while sourcing and selling, and as they grow the portals' influence will attract more companies to the platform. Such sites, coupled with large and robust international industry organizations like SAE, can also set widely accepted standards and course-correct entire technology sectors when new infiltration issues arise. As tracing technologies like physically unclonable functions (PUFs) mature, industries may also look to standardize traceability technology to provide better guarantees to end manufacturers regarding part origin. Finally, international and intergovernmental organizations should continue investigating issues surrounding counterfeiting and counterfeit part infiltration, and work to ensure governments cooperate in rooting out IP theft and enforcing international IP standards.

2.3 Existing Solutions in Anti-Counterfeiting and Traceability

This section presents a survey of existing organizational, legal, and technological approaches to mitigating counterfeiting and traceability concerns.

2.3.1 Company-led Remedies

At the single organization level, strategies often involve signaling "good member" status in the given supply chain. For instance, the 2013 DOD report emphasizes that focusing on "best value, as opposed to lowest cost" encourages suppliers to ensure quality and legitimacy, and signals to customers your integrity [9]. Of course, this objective competes with an ever-present need to keep costs down, and many companies may not be willing or able to hold up to the desired standards when forced to compete with i) well-established competitors that benefit from cost-saving

infrastructure and ii) less scrupulous competitors that save costs via cutting corners when vetting suppliers or providing assurances to customers.

On the other hand, more aggressive approaches in dealing with suppliers and customers may also be used. Typically, this involves requiring quality or traceability guarantees among suppliers or holding to these guarantees when negotiating with customers [5,9]. In instances where the identities of counterfeiters are known, surveillance and eventual litigation may also be pursued, albeit usually at great cost to the IP holder [6]. Enforcing traceability generally requires large-scale consensus, or market dominance in a particular step in the supply chain. Otherwise, legislation could be used to coerce better traceability policies. Of course, this approach may involve expensive lobbying, annoy up-stream members of the supply chain, and have limited benefit if counterfeit parts are entering the the supply chain out-of-country and local suppliers have plausible deniability regarding any part's true origin (thus circumventing the "knowing participation" rule in most anit-counterfeiting legislation).

2.3.2 Technology-based Remedies

Technology-based anti-counterfeiting efforts generally involve developing and deploying methods for ensuring traceability and identity verification for parts moving through a supply chain. In the simplest case, manufacturers may rely on serial numbers or other stamps or etchings on their components to signal genuinity. Unfortunately, these markings are often easy for a determined counterfeiter to copy, and do not prevent the re-sale of out-of-specification parts in after-market scenarios.

More advanced tracing technology includes radio-frequency identification systems (RFID), which provide a more reliable way to electronically monitor parts moving through a supply network or a warehouse [31, 32]. RFID has enjoyed increasing popularity over the past decade or so, and is commonly used to track large shipments. This method, however, tends to be expensive if applied at a per-part (per-instance)

basis, may be invasive during manufacture, and requires some infrastructure at the sending and receiving ends.

Further tagging methods may involve Physically Unclonable Functions (PUFs), which may take many forms as described by Suh and Devadas [33] (see Section 2.4). Common implementations include optical chips that produce random but repeatable speckle patterns, capacitance or coating based methods, and other various electronic methods useful for validating the identity of computers and chips. One interesting approach adopted by several DOD suppliers involves tagging parts with plant DNA, which is robust to many damage types and nearly impossible to reproduce [34].

Many traceability concerns are eliminated with per-instance validation techniques implemented by the original part manufacturer and end part customer, but success assumes both sides are actively interested in achieving good traceability. This may not be a good assumption in many cases, such as automobile repair or in cases where suppliers wish to put pressure on competitors through counterfeit parts entering the market. A per-instance method would require large-scale adoption that legislation or international cooperation would have to back.

Another promising development is the increasing popularity of industry-specific supplier portals. Such sites essentially serve as a database of suppliers that may be vetted by an industry, and that by participation in the portal agree to industry-specific guarantees of quality, genuity, and traceability that may be as loose or as strict as the community, local government, or international law permits. For example, Covisint is a portal site serving numerous automotive companies and their suppliers [35].

2.3.3 Legal and International Organization-based Remedies

Most nations have IP-protection related legislation. In the US, the most important law regarding counterfeit goods is *US Code: Title 18, Part I, Chapter 113, Section 2320: Trafficking in Counterfeit Goods or Services* [36]. Section 2320 provides federal penalties for producing, selling, or trafficking counterfeit goods in the US. The law

provides for up to ten years in prison and $2 million in fines for knowingly producing or trafficking in counterfeit goods as an individual, and fines up to $5 million for "[persons] other than an individual." Unfortunately, this law has limited to no impact when dealing with counterfeiters operating in other countries, and it is often difficult to prove that US-based suppliers "knowingly" participate in trafficking counterfeit goods. Similar laws exist in other nations where US industries have deep connections (China, India, various European nations), but similar difficulties arise there too, and often nations are reluctant to cooperate in international counterfeit investigations (see the case studies from Section 2.1, [26, 27]). In general, national laws seem to have a small impact on counterfeiting activity in international supply chains, compared to efforts on the company, industry, and international level.

At the widest level, one can also consider efforts made internationally, implemented through organizations representing groups of companies and/or nation states. The most visible example of such an organization is probably the International Anti-Counterfeiting Commission (IACC), a coalition of about 250 member companies and forty member nation aimed at combating product counterfeiting and piracy around the world [37]. The organization is valuable in that it raises the visibility of the issue, and organizes several helpful initiatives aimed at protecting intellectual property. IACC's most prominent initiatives at time of publication include the RougeBlock and MarketSafe programs. RougeBlock allows for easier reporting of illicit merchant activity with a focus on shutting down merchant accounts on common digital storefronts, and facilitates information flow between participating members. The MarketSafe program is a strategic partnership with Alibaba to track and shut down counterfeiter accounts on the storefront.

SAE International is also a valuable organization in combating industrial counterfeiting. The SAE issues non-governmental standards aimed at informing supply chain members about best practices and expected risks. In particular, the *Counterfeit Electrical, Electronic, and Electromechanical (EEE) Parts; Avoidance, Detection, Mitigation, and Disposition Standard* of 2013 [38] is intended to inform international

community-based counterfeit mitigation, and intended to be enforced through contracts between, for instance, the DOD and US-based defense contractors [9]. Terms laid out in the standard include training personnel in proper sourcing and receiving of parts, ensuring safe after-market sourcing during repair, mandatory reporting of suspected parts as they are discovered, and mandatory SAE audits.

Finally, the World Intellectual Property Organization (WIPO), an agency of the United Nations, is the official intergovernmental assembly that oversees IP policy and discussion. Established in 1967, WIPO contains 189 member states and facilitates cooperation between members when IP related issues come up at the UN [39].

2.4 Traceability through Physically Unclonable Functions

Physically unclonable functions, or PUFs, are functions applied to some input component, the *challenge*, that produce an output *response* that serves to identify the input as genuine or otherwise [33, 40]. In particular, it should be difficult for an attacker to determine the response a challenge should output without seeing the challenge first, regardless of the number of challenge-response pairs the attacker may know. PUFs leverage inherent or externally introduced randomness of a part to generate a discriminating, yet robust, identifying string for that part. What constitutes adequate discrimination and robustness is a decision made by the entity implementing a PUF.

There is a need to extract randomness from a part and form some response that acts as an identifier for that part. This is similar to the "fuzzy extractor" methods used in biometrics and other cryptographic applications with noisy random sources [41, 42], although without the error correction implied in fuzzy extractor schemes; indeed, a measure of difference between expected and realized response could encode not just identifying information, but information on damage sustained by the part, previous use, and other valuable information[3].

[3]Fuzzy extractors, error correction, and encoding such information is explored in greater detail in Chapter 6 through the concept of a *malleable PUF*.

The most common PUFs involve evaluating integrated circuit (IC) responses to a given challenge [33] that vary randomly between instances of the IC. Such PUFs have varied operating principles, including, for example, measuring the spatially varying capacitance of a coating or the delay between different gates of an IC [43]. Optical PUF designs, for instance leveraging the random distribution of particles suspended in a transparent media as proposed by Pappu [40], are also well-studied. Existing counterfeiting solutions in manufacturing all draw from the principles of PUFs. For instance, the use of physical attachments (e.g., RFID tags or holographic markers) for counterfeit detection [44] and paper- or packaging-based methods of counterfeit detection [45,46] draw on IC- and optically-based PUF designs. Recent work has also pointed out the rising potential for nano-scale PUF development [47].

In this dissertation, instance-specific surface analysis is proposed to extract the necessary randomness from instances of a metal part. Analysis of metal surfaces for anti-counterfeiting has shown some promise. For instance, Takahashi has shown the potential identifying characteristics of surface finish features [48]. Cao presented an analysis of microscopic responses to nano-scale features [49]. However, to the best of the authors' knowledge, a generalized framework for leveraging surface microstructural features of metallic parts for anti-counterfeiting in the face of expected part damage or imaging inconsistencies has not been presented.

2.4.1 Common Designs for PUFs

There are many PUF designs that are non-intrinsic, i.e., whose randomness must be explicitly "injected" into the object during the manufacturing process [50–52], as well as PUF designs leveraging randomness *intrinsic* to the part under inspection. Both concepts are relevant for the anti-counterfeiting and traceability schemes discussed in later chapters. Common examples include integrated-circuit PUFs, optical PUFs, coating PUFs, and paper PUFs. These are briefly highlighted below.

- The most common PUFs involve evaluating *integrated circuit (IC) responses* to a given challenge [33]; the responses are chosen such that they vary randomly between instances of the IC even under identical manufacturing conditions. Examples include measuring the spatially varying capacitance of a coating or the delay between different gates of an IC [43].

- *Optical PUFs* leverage visually-acquired features to determine authenticity. The original optical PUFs are based on the speckle pattern caused by shining a laser on a transparent material containing randomly inserted scattering particles [40, 53, 54]. Several patents and articles present methods for extracting optical surface features from objects for anti-counterfeiting purposes [55–57]. Related to this are various tagging methods such as RFID [31,32] and hologram tagging [58]; and tagging parts with plant DNA, which is robust to many damage types and nearly impossible to reproduce [27, 34].

- The *coating PUF* is based on the capacitance between metal lines in a silicon chip, one layer of which consists of a coating into which dielectric particles were randomly inserted [59].

- The *phosphor PUF* proposed by Chong et al. [53, 60] relies on randomly blending phosphorus particles into a material, which makes it possible to challenge using standard optical equipment currently deployed in, say, automated visual inspection systems.

- Many proposals exist for optical *paper PUFs* that make use of paper microstructure, most of which are for preventing the counterfeiting of paper currency notes. For example, the approach by Bulens et al. [61] relies on specially manufactured paper in which ultraviolet-reflecting fibers have been randomly inserted. Buchanan et al. [62] use regular paper but require the external measurement to use a focused laser beam.

2.4.2 Computer Vision for Anti-Counterfeiting

The computer vision literature covers a wide array of feature extraction algorithms, as well as accepted methods for analyzing the performance of classifiers built on the features extracted. For example, the scale-invariant feature transform (SIFT) is a well-known algorithm for extracting local feature vectors for object recognition and related computer vision tasks, and has been used for biometric applications like fingerprint matching [63]. Binary feature extraction algorithms like ORB and BRIEF have seen success recently, and are often faster to compute and require less storage than competing algorithms [64, 65]. While such algorithms are effective in object recognition tasks, the features extracted are hard to intuitively relate to the structure of input images without clear target objects, as is often the case in random-field-like micrograph images [66].

2.4.3 Use of Statistical Micrograph Descriptors in Materials Science

Optical PUF schemes can be designed to leverage micrograph descriptors for determining authenticity [40]. For example, I present studies demonstrating the use of volume fraction and grain size descriptors in metal part anti-counterfeiting schemes [67, 68] in Chapters 3 and 4, while paper fiber features have been used to develop schemes for packaging traceability [46]. Work presented in Chapters 3 and 4 show that grain size and volume fraction descriptors have been shown to provide reasonably robust, small identifying strings for polygonal structures (360 brass microstructures) and multiphase alloys (4140-steel microstructures).

Grain boundary descriptors also make possible candidates for feature extraction, and have been used with success in materials characterization and evolution literature [69]. Geometric data or data on edges or vertices of grain boundaries, if they are expected to remain after damage, could serve as useful reproducible features for PUF design.

2-point auto-correlation and cross-correlation statistics are physically meaningful features that can be efficiently extracted from micrograph images where each pixel is assigned to one of N phases [70]. These statistics represent a good vein for constructing high-information features, and are discussed in greater detail in Chapter 5. Indeed, the study presented in Chapter 5 centers such features for use in anti-counterfeiting.

2.5 The Case for Tailored Traceability, Revisited

To summarize, several observations on the state of modern manufacturing motivate the work in this dissertation. **First**, manufacturing today is *highly decentralized*, and so finished products often contain parts from many different sources. **Second**, this decentralization opens a *large attack surface*, where bad actors or simply bad luck can impact one link in the chain (a specific part, or a specific manufacturing step) and severely damage the product's performance. **Third**, modern manufacturing processes and metrology lead to *highly unique, readable material structure* even within manufacturing batches, and these structures can serve as signals of genuinity. Indeed, the PUF literature leverages such structures for traceability already. A **fourth** observation, discussed in greater detail in Chapters 6 and 7, is that modern manufacturing involves many processes that can be tailored to *embed signals in manufactured parts directly*, and such signals may serve as extrinsic sources of information that can carry more complex messages.

And while these points can be applied in many manufacturing contexts, it should be pointed out that additive manufacturing (AM) is a major motivating technology for this dissertation. AM is growing into a major (perhaps *the* major) disruptive technology in manufacturing. The topics discussed in this dissertation, especially those involving tailoring features to read and embed for traceability, are uniquely suited for this environment, since the structure of the part can be finely controlled by the engineer and, in principle, can even be tailored dynamically during the additive

manufacturing process. From a more long-term perspective, the rise of AM brings novel issues: the outsourcing of AM part manufacturing promises to reduce barriers for suppliers (trustworthy or otherwise) to enter the market, and the rise of bespoke small-batch manufacturing implies quality standards will be more difficult or impossible to enforce. These concerns need to be anticipated, and as a community we should spend time looking at these issues now so they don't sneak up on the industry. Security in AM is a growing field of research, with researchers scoping out existing issues in AM cyber-physical security [71,72]; this work is given greater attention in Chapter 6. This dissertation aims to add to this literature by proposing bespoke methods to "bake in" security and traceability throughout the product manufacturing and life cycle.

2.5.1 Research Gaps

Within the fields of traceability and anti-counterfeiting, there lacks knowledge about the use of intrinsic, unclonable manufacturing features for developing robust, context-specific anti-counterfeiting schemes. Specifically, manufacturing environments and the resulting structures, be they phase distributions, manufacturing surface artefacts, or others, provide a rich but under-utilized source of randomness for anti-counterfeiting and traceability. There exists a research need to study quantitatively the use of this randomness for context-specific anti-counterfeiting schemes: given a manufacturing process, a material system, available data acquisition method(s), and expected damage profiles, how may such schemes be automatically generated and quantitatively evaluated?

There is a need to develop a general framework for creating algorithms to *enroll* parts and their corresponding data (for example, micrograph inputs), produce a light-weight scheme for determining whether a challenge image indeed corresponds to the enrolled data, and consider the high volume and high damage expected in transportation or use.

3. A PUF DESIGN AND ANALYSIS FOR LIGHTLY ETCHED 360-BRASS

Towards addressing Research Questions 1.1 and 1.2 (see Figure 1.2), this chapter deals with the first of several case studies presented in this dissertation: a case study to illustrate the feasibility of traceability/anti-counterfeiting schemes built on microstructural feature data. Through these case studies, I aim to establish the feasibility of using such features for traceability scheme design, and to propose a general framework for designing such schemes.

In this chapter, I present an efficient PUF formulation for converting a micrograph into a bitstring that is a unique representation of that micrograph, with the goal of establishing the feasibility of using physically meaningful micrograph descriptors for robust anti-counterfeiting schemes. This chapter follows the work of my previous study of this PUF formulation problem for 360 brass [67]. I illustrate that the approach is robust to minor wear-and-tear during usage. Key advantages of the proposed approach are as follows: (i) this approach uses non-replicable surface microstructure as a PUF challenge captured using an optical microscope, and (ii) manufacturers need not modify their production processes with any sophisticated equipment. A manufacturer can discredit the genuinity of the product if the derived bit string cannot be verified.

Both to motivate the work in this chapter, and to lay the ground for additional extensions to the design framework discussed in Chapters 4 and 5 and extensions to information embedding discussed in Chapters 6 and 7, I will begin with a more targeted overview of the use of microstructural information for designing these PUF schemes. The proposed PUF formulation is discussed in the following section.

3.1 Using Material Microstructure to Design PUFs and Schemes for Information Embedding

As discussed in the previous chapter, the microstructure of materials plays a central role in understanding the behavior of materials. Material scientists have utilized microstructure to characterize material properties based on the microstructure at various scales, using structure-property relationships to design new materials [73]. Existing efforts in materials science are focused on identifying microstructure descriptors that can be (i) related to material properties such as strength, and (ii) controlled through manufacturing processes. Examples of such microstructural descriptors include n-point spatial correlations, lineal path functions, grain size distribution, shape distribution, and orientation distribution [74].

In this dissertation, I present a different use of material microstructure. I illustrate that the microstructure and its descriptors can be used (i) *to uniquely identify a manufactured metallic product*, and (ii) *in appropriate manufacturing scenarios, like additive manufacturing, can be leveraged to design secure and robust methods to embed information in an instance of a part.* A material's microstructure is a result of the manufacturing process, which by its nature is inherently random. A micrograph is a specific instance of the random microstructure [66], which is practically impossible to duplicate even if the manufacturing process and its control parameters are precisely known. Hence, the micrograph is a unique fingerprint of the product, and can be used as input to a properly designed Physically Unclonable Function (PUF) to determine the authenticity of the product.

A naïve approach to leverage this uniqueness is to capture a micrograph from each product and store it in a database for future comparison. However, direct comparison of micrographs is impractical due to associated computational cost, particularly when the number of products is large. Hence, the challenge is in establishing efficient representations of the micrograph that are efficient to compute and compare. This challenged is addressed by designing a PUF Π that takes as input a micrograph x and

outputs a string y, such that $y = \Pi(x)$, which is a unique, robust identifier for the microstructural region captured by the micrograph. Extensive work has been done in formalizing PUFs [43]; here the key features of well-designed PUFs, and consequently the schemes to be proposed in this dissertation, as presented by Maes and coauthors are summarized below:

1. **Evaluatable**. A PUF $y = \Pi(x)$ should be easy to evaluate for $x \in X$, where X is the "challenge space" of the PUF. Here, X is the space of all possible micrographs.

2. **Unique**. $\Pi(x)$ is a unique representation of $x \in X$ such that $\Pi(x_i) \neq \Pi(x_j)$, $\forall x \in X, i \neq j$ with high probability.

3. **Reproducible**. $\Pi(x)$ is consistent for small errors ϵ in x, that is, $\Pi(x \pm \epsilon) = \Pi(x)$.

4. **Unclonable**. Given $\Pi(x)$, it is hard for an adversary to construct some function $\Gamma(x) \approx \Pi(x)$.

5. **Unpredictable**. Given a challenge-response set $Q = \{(x_i, y_i)\}$ of arbitrary size, it is hard for an adversary to estimate $y_c \approx \Pi(x_c)$ for $(x_c, y_c) \notin Q$.

6. **One-Way**. Given y and Π, it is hard for an adversary to construct the initial x.

7. **Tamper-evident**. Any deliberate alteration to the physical part under consideration transforms $\Pi \rightarrow \Pi'$ such that $\Pi'(x) \neq \Pi(x)$, $\forall x \in X$ with high probability.

3.1.1 Optical PUFs for Anti-Counterfeiting of Metallic Goods

Many techniques for extracting features from two-dimensional data structures like images have been developed by researchers in the field of computer vision. These

techniques are useful for tasks such as object recognition, object tracking, texture analysis, image matching, and others.

For the purposes of anti-counterfeiting in this dissertation, it is necessary to extract robust descriptors from a micrograph in a way reproducible under part damage and illumination changes. Well-known algorithms for image keypoint detection and description under such conditions include SIFT, SURF, and ORB [64, 65]; such techniques identify a large volume of keypoints, often "corner" points, and describe them in a rotation- and scale-invariant way.

The output descriptors from an input image using these methods tend to be large: on the order of 512 bytes per keypoint descriptor for SIFT, or more for SURF and ORB, where dozens to hundreds of keypoints may be necessary to correctly identify a challenge image. Also, SIFT and SURF are patented algorithms, making implementation in practice more difficult. Finally, such algorithms are often applied to images with easily-defined objects, textures, and corners, not the polygonal Voronoi- or random-field like micrographs. However, these techniques may still be useful when combined with other micrograph features, and should be considered in future work. In this dissertation, the focus is on leveraging established microstructural descriptors for generating these features, such that for a given system their behavior will already have been characterized, or can be readily characterized, using standard techniques of materials science. This is the approach followed in this chapter.

3.2 Approach

This section serves to illustrate the approach using 360-brass, which under common manufacturing conditions exhibits an equiaxed space-filling grain structure that allows straightforward calculation of features like mean grain intercept length, L_3. Although this Chapter focuses on one material and feature, I emphasize that the framework proposed is intended to generalize easily for general microstructures and their relevant features. This section discusses the relevant image capture and pre-processing

details, as well as the feature extraction and bit string construction methodology. This is followed in later sections with a discussion on the results of this analysis on several brass samples.

3.2.1 Image Capture and Pre-Processing

Several 360-brass samples from the same manufacturing batch were prepared for optical microscopy prior to analysis. The samples were polished using successive fine grit paper, diamond suspension, and alumina powder polishing. Samples were then etched by submersion in a 50% solution of nitric acid for five seconds. 50 micrographs were taken at various locations at 100X magnification, as shown in Figures 3.1 and 3.2(a). Each micrograph was captured using an AmScope MU500 5.1MP digital camera. Captured images (Figure 3.2(b)) were then pre-processed for further analysis using boundary-preserving median filtering (radius=5) and the moments-preserving auto-thresholding algorithm proposed by Tsai [75]. Image processing was carried out using the ImageJ Fiji distribution [76]. These images are then used to extract the relevant features and build the subsequent bit-string.

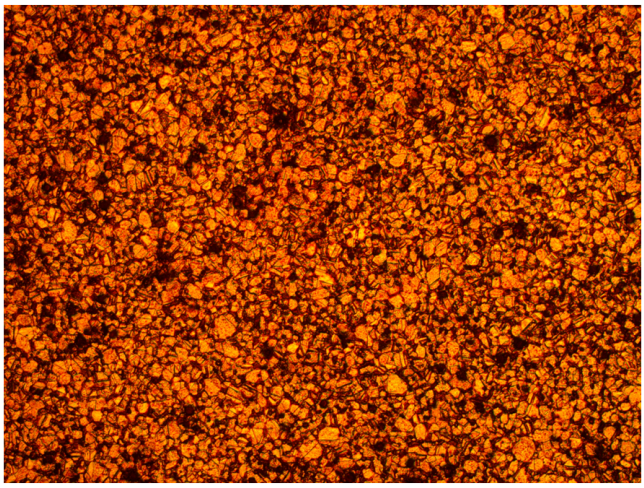

Fig. 3.1. Micrograph of 360 Brass etched with 50% Nitric Acid solution, 100x.

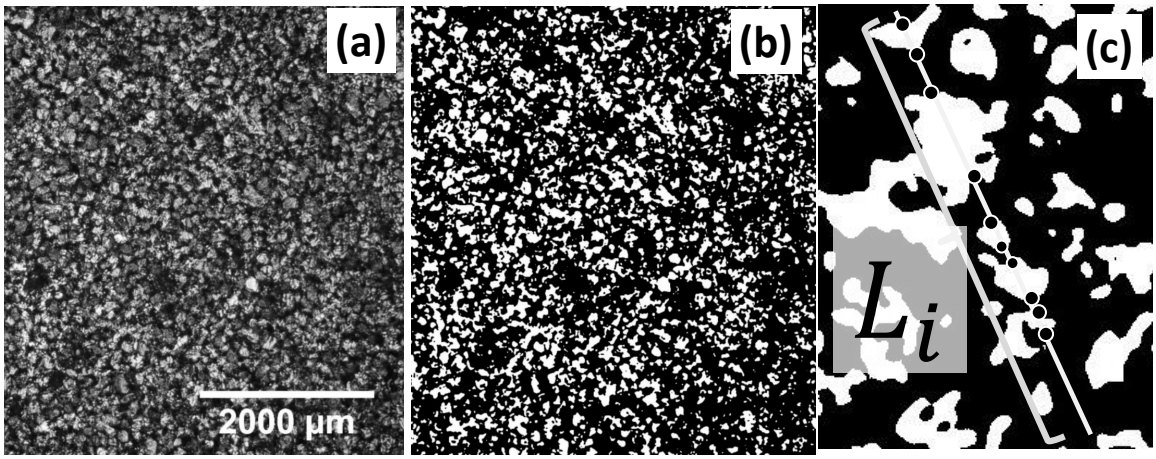

Fig. 3.2. (a) Micrograph of 360 Brass etched with 50% Nitric Acid solution, 100x. (b) Same image after pre-processing. (c) Example test line L_i, with $P_i = 10$.

3.2.2 Feature Extraction and Bit String Construction

Many well-established microstructure properties may be leveraged in gathering discriminating information from the processed micrographs [74, 77, 78]. Here, I focus on the 3-dimensional mean intercept length of the grains L_3, calculated via performing automated lineal analysis on the processed micrographs.

Mean Intercept Length Calculation

The mean grain intercept length is given for space-filling grain structures by Tomkeieff's Equation [79] using experimental measures of the normalized intercept count N_L:

$$N_L = \frac{\sum_{i=1}^{m} P_i}{\sum_{i=1}^{m} |L_i|} \tag{3.1}$$

Where P_i is the number of intersection points on line L_i, and $|L_i|$ is the length of L_i, $1 \leq i \leq m$. For each image considered, $m = 1,200$ lines were applied with

random lengths and orientations to the processed image result. Here m was chosen to be high enough to guarantee convergence to a common L_3 regardless of the random seed applied to the line generation algorithm. Discontinuities in each line profile were taken as intercepts, with the total intercept count taken as P_i. An example L_i test line and the corresponding intercepts are illustrated in Figure 3.2(c). The mean grain intercept length for the given image is then, by Tomkeieff:

$$L_3 = \frac{1}{N_L} \tag{3.2}$$

This calculation is then used to formulate an image bit string by the following method.

Image Tiling and Bit String Calculation

The L_3 calculation may be carried out for each micrograph to get a characteristic value for that sample. However, this measure varies in different *subregions* of the micrographs as a consequence of the random field nature of the grain structure at the relevant length scale. The proposed bit string extraction method leverages this inherent randomness to create discriminating strings over a family of images taken from the same brass sample.

Each image is segmented into several length levels, each of which consists of *Regions of Interest* (ROIs) in which the L_3 statistic is calculated following the procedure described above. I refer to the level constituting the entire image in one ROI as the characteristic level, and index subsequent levels according to their relative ROI size. The level containing the largest ROIs by area is indexed 0, with each subsequent level indexed 1, 2, and so on. This tiled segmentation approach is illustrated in Figure 3.3.

For each ROI across all levels, L_3 is calculated. Then, each ROI result is compared to the median result across all ROIs, and is assigned a bit of 1 if above the median and 0 else. The median comparison increases the robustness of this method when considering images taken at a later time with possibly damaged parts or slight

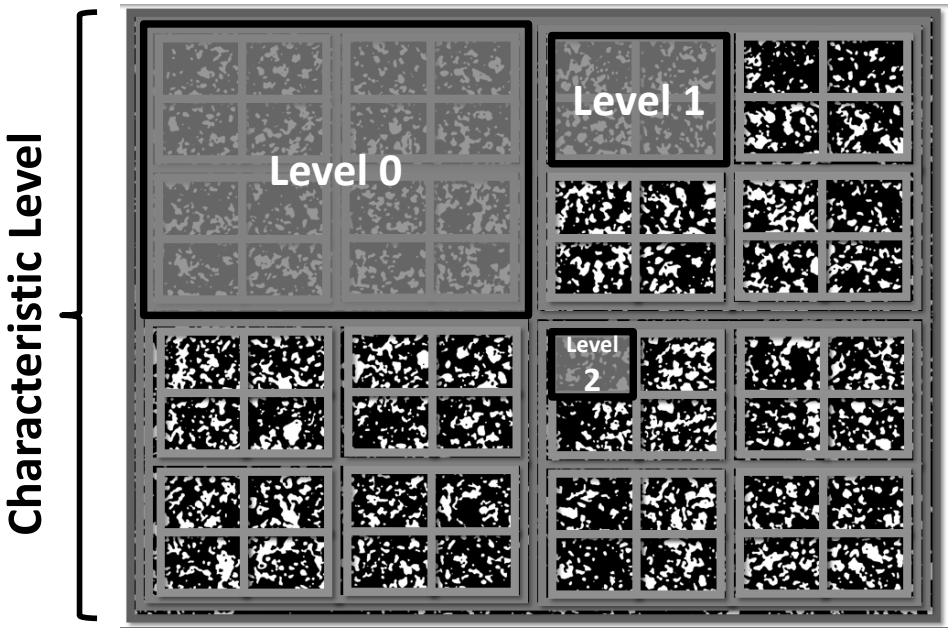

Fig. 3.3. Example of a quad-tiled pre-processed micrograph.

microstructural changes. A histogram of these calculations for one sample image is provided in Figure 3.4. As an illustration, a segment of one possible bit string constructed using three ROI levels is:

(level 0) - (level 1) - (level 2):

0011-1000000110110011-11010110101111000001....

Where the first four bits correspond to the results from the Level 0 ROIs, the following sixteen from the Level 1 ROIs, and the remaining 64 (not all shown) from the Level 2 ROIs.

Now, it is clear that for each low-level bit, there are four corresponding bits in the higher level, which should vary with greater probability as the microstructure changes over a part's life cycle. Thus, to calculate the difference between bit strings generated by this method, I propose a modified Hamming Distance metric [80] which penalizes deviations in lower-level bits more than those of high-level bits. For two bit strings

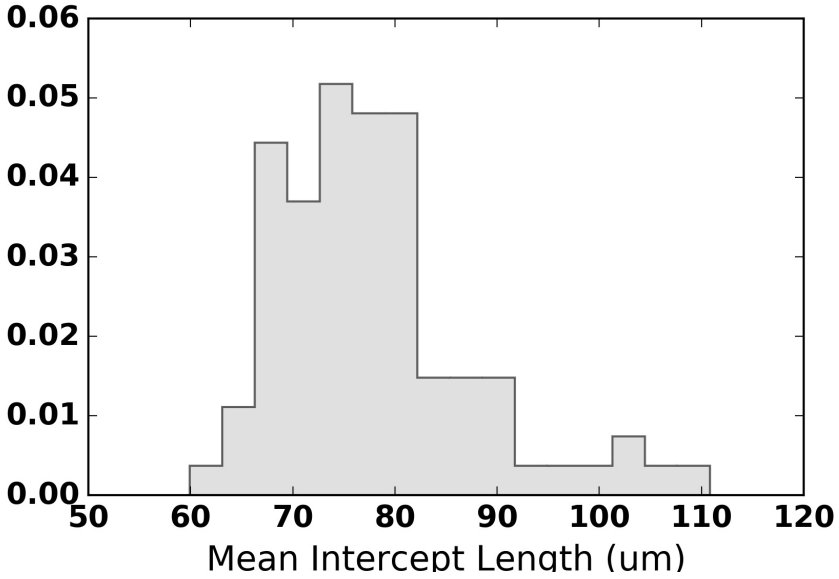

Fig. 3.4. Histogram of L_3 across 84 ROIs for a representative brass micrograph.

a, b of length L and entries a_i, b_i, $1 \leq i \leq L$, taken from an image segmented with a maximum K level, the proposed Modified Hamming penalty function H is given by:

$$H(a, b, K) = \frac{\sum_{i=1}^{L} q(a_i, b_i, k_i)}{\sum_{i=1}^{L} 4^{K-k_i}} \tag{3.3}$$

Where $q(a_i, b_i, k_i)$ is the bit-wise comparison function on the bit string entries, k_i is the corresponding tile level of entry i, $0 \leq k_i \leq K$, and $\sum_{i=1}^{L} 4^{K-k_i}$ is a normalizing term. $q(a_i, b_i, k_i)$ is given by:

$$q(a_i, b_i, k_i) = \begin{cases} 4^{K-k_i}, & a_i \neq b_i \\ 0, & a_i = b_i \end{cases} \tag{3.4}$$

Thus, the penalty function penalizes each bit discrepancy based on the relative size of the corresponding level tile. A distance value $H(a, b, K) = 0$ indicates complete agreement between strings a and b, while $H(a, b, K) = 1$ indicates complete reversal of bits between a and b. Next, I discuss the results of applying this analysis to the brass data.

3.3 PUF Protocol Analysis

In this section, the performance of this PUF protocol is analyzed by considering the reproducibility, evaluatability, and uniqueness of the response bit strings. The remaining PUF features, although important, are not considered here as further experimentation is required to provide complete analyses. However, I briefly comment on these features now for completeness:

1. **Unclonability, Unpredictability, and One-Wayness**. The PUF response is highly dependent on (i) the parameters of the part's manufacturing process and (ii) inherent randomness in the formation of local microstructure. Thus, it is highly unlikely an adversary could reliably reproduce a part with the necessary local microstructural properties to emulate an authentic instance, or predict the response ahead of time.

2. **Tamper Evidence**. Severe alteration of a part would destroy the local microstructural regions of interest in the part with high probability. It is unlikely such a damaged part would then be used in practice.

3.3.1 Reproducibility and Evaluatability

For each of the 50 micrographs, an 84-bit string was constructed by analysis of Level 0, 1, and 2 ROIs for each tiled processed image. The distance between each bit string and the strings for all other images was calculated according to the distance metric defined above in order to evaluate the inter-distance between non-identical image challenges. To test the robustness and reproducibility of this method, simulated "damage" was applied to each image in the form of dark striations on the raw images, simulating scratches of width $\approx 0.05mm$. The analysis was run with various damage severities (number of striations), and the resulting strings were compared to the original strings gathered for each image to get an intra-distance measure. Inter- and intra-distance histograms for the data set across several simulated damage sever-

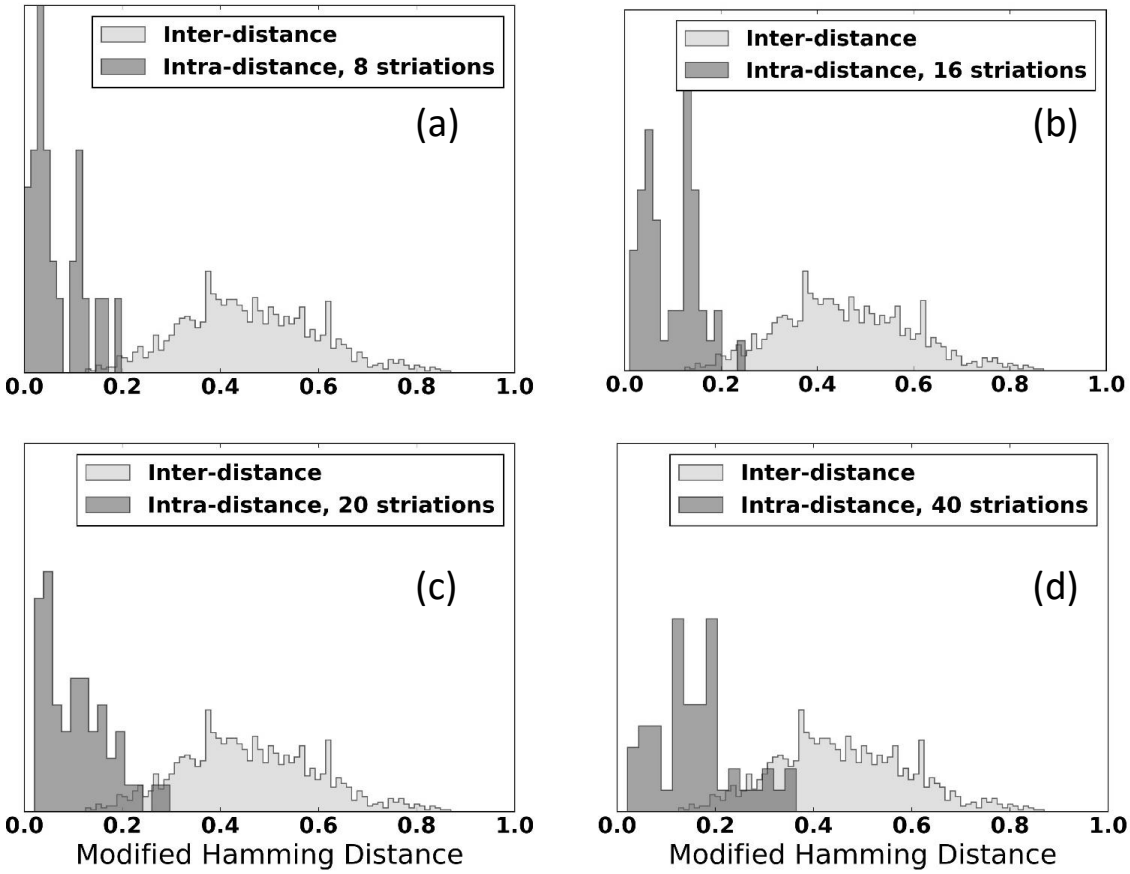

Fig. 3.5. Inter- and intra-distance results for 50 images with (a) 8, (b) 16, (c) 20, and (d) 40 dark parallel striations applied to each at random to simulate damage.

ities are plotted in Figure 3.5, which emphasize the PUF's reproducibility and the uniqueness of the output strings.

Timing data provides a method of studying the evaluatability of the proposed method. On a machine equipped with an Intel i5 3.2GHz CPU and 8GB RAM, the protocol took on average 12.1s to produce a bit string from a 2,592x1,944-pixel raw micrograph. Note this protocol is easy to parallelize across ROI calculations, and such parallelization will drastically reduce this calculation time. Individual ROI L_3 computations take on average 0.22s.

3.3.2 Uniqueness: Enhancing Discriminatory Ability

In practice, a user would compare a "challenge" image result taken from a suspect part with the "original" enrolled image bit string using the Modified Hamming distance measure. Note the small overlap in Figures 3.5 (a,b), which represent lesser scratch damage compared to Figures 3.5 (c,d). For challenges resulting in distances $H \leq 0.13$, across all sampled images, one can be confident that the challenged image is indeed genuine. Similarly for $H \geq 0.23$ for challenge images with light to moderate scratch damage, one can be confident that the challenged image is not genuine. Evaluating the performance of proposed schemes using the inter- and intra-distance histograms constructed from a training set of micrographs before and after damage is discussed in greater detail in Chapters 4 and 5.

Note that these results indicate significant uncertainty for a moderate range of H for more heavily damaged components, and thus a more discriminatory string construction may be desired for parts likely to become heavily damaged. The discriminatory ability can be further enhanced by collecting more feature measurements per image, rather than the single measure of L_3 considered here. Nevertheless, this analysis shows the reasonable discriminatory power of very small reliable bit strings derived from microstructural features. How best to extend this method and increase discriminatory ability while maintaining robustness is discussed in subsequent chapters.

3.3.3 Comments on an Improved Scheme

Here, I sketch a brief improvement to the scheme that makes it resistant against a future counterfeiting adversary who has the manufacturing capability to control, in any given small rectangular patch of material, the bit-string for that particular region. The improvement that thwarts such a future adversary consists of a simple modification to the scheme: Instead of generating the bit-string using the deterministic tiling of the micrograph with small squares of equal size (a tiling that is known

to the adversary), I now use a tiling with rectangles that is generated using a cryptographic key that is not known to the adversary; the rectangles I use have a bounded aspect ratio (hence are "square-like" and not arbitrarily thin). An adversary who does not have the cryptographic key is unable to produce the tiling, and therefore unable to determine which rectangular patches he should target for the use of his superior manufacturing prowess.

3.4 Integration in Manufacturing

This approach can be automated and need not involve experts. The genuinity of the product may be verified using the following steps:

1. Certain unique features of the microstructure are extracted from a captured micrograph, and a reliable bit string is extracted based on these unique features.
2. The bit string is signed by the manufacturer using a private key derived from production details such as batch number, part number, material supplier, etc.
3. All such signed bit strings are stored in a database.
4. This database is made available to the entire supply chain (including end-customers).
5. Any participant in the supply-chain can submit the derived bit string of the suspected product and verify it with the manufacturer database.

This protocol may be applied to any bit string extraction technique considered, such as the one presented in this chapter or the following two chapters. Economically feasible automation of this and similar protocols, however, should be a focus of future work and is not further discussed in this dissertation.

4. PUF DESIGNS AND COMPARATIVE ANALYSIS FOR LIGHTLY ETCHED 4140-STEEL

As another case study, in this chapter I consider PUF designs for 4140-steel, following the work of my previous study [68]. As in the previous chapter, this chapter addresses Research Questions 1.1 and 1.2 (see Figure 1.2). This case study also explores evaluating features for carrying capacity (Research Question 1.3) more thoroughly; this analysis is expanded to a larger library of features in Chapter 5.

Following from the previous chapter, I build on the bit string construction and analysis methods to more formally describe the formulation of a general optical scheme for anti-counterfeiting in manufacturing, and discuss methods for tailoring the design to fit data usage and performance constraints. The formalization in this section is built upon in the subsequent chapter, which considers a larger input library of micrographs and a much larger library of potential features for use in the scheme.

4.1 PUF Formulation

As discussed in the previous chapter, PUF formulation involves the complete specification of a PUF design Π that maps a discriminating, robust string b_i to a genuine instance i of a manufactured good. That is, formulate PUF $\Pi(i)$ that outputs a string b_i such that it is likely that the same part instance at a later time, $\widetilde{i}$, perhaps damaged in a way anticipated by the experimental and idealized data analysis, yields a similar string $\widetilde{b_i} \simeq b_i$, but that it is also likely to generate a highly dissimilar string when applied to an instance $j \neq \widetilde{i}$. In this subsection, notation is developed for formalizing PUF designs in the general case.

Notation and formulation. First, recall that a PUF $\Pi(i)$ is some function that takes as input an instance i of a part, and outputs an identifying string b_i for that

instance, $\Pi(i) = b_i$, which is used to classify the part as genuine or counterfeit. This process requires:

1. a procedure $\Gamma(i)$ for extracting some feature vector v_i from i, such that $v_i = \Gamma(i)$,
2. a map $g(v_i)$ that outputs a string b_i from the feature vector, such that $b_i = g(v_i)$,
3. a measure $H(b_C, b_i)$, denoted H for simplicity, quantifying the dissimilarity between a challenge response string b_C and an enrolled response string b_i, and
4. a decision rule $\Lambda(H)$ that classifies a *challenge instance* C as genuine ($C = \widetilde{i}$) or counterfeit ($C = j$) based on the dissimilarity H of response string b_C and enrolled string b_i.

Thus the complete PUF formulation is constructed by specifying first the string construction method given i,

$$\Pi(i) = g\big(\Gamma(i)\big) = g(v_i) = b_i, \tag{4.1}$$

and then specifying the decision rule for classifying any $\Pi(C) = b_C$, where if $\Pi(C = \widetilde{i}) = \widetilde{b_i}$ and $\Pi(C = j) = b_j$,

$$\beta = \Lambda\big(H(b_C, b_i)\big), \quad \beta \in \{\widetilde{i}, j\}. \tag{4.2}$$

Application. Constructing $\Gamma(i)$ requires specifying all aspects of feature extraction. In the optical case, for instance, $\Gamma(i)$ incorporates image capture, material pre-processing (if any), image pre-processing (if any), and the algorithms used for extracting relevant features from the images. Since v_i is not intended to be stored, the size of v_i is generally not a major concern. However, the evaluation time of $\Gamma(i)$ may be important if being applied to, say, an assembly line in real-time.

Constructing $g(v_i)$ requires specifying some map of v_i to the string b_i in a way that each instance is highly likely to produce a unique string with respect to other instances from the same manufacturing line, but that the string is robust to expected part damage. Since b_i, the enrolled string, will be stored, minimizing the size of b_i

may be important; thus analyzing the trade-off between string length and discrimination/robustness may be important in the experimental data collection and structure idealization phases of PUF design.

$H(b_C, b_i)$ should be constructed to properly capture the feature encoding of $g(v_i)$. For instance, some bits in b_i, if different in b_C, may encode a greater dissimilarity between C and i; in this case H should account for this difference. So, the construction of H is highly dependent on how the designer chooses to construct $\Gamma(i)$ and $g(v_i)$ and should be designed to minimize the classification error of $\Lambda(H)$.

4.1.1 PUF Evaluation

Given sets of possible Γ, g, H, and Λ for some implementation scenario, the problem becomes choosing a combination for the PUF formulation. Here, several PUF evaluation concerns are important. A PUF designer would choose the Γ, g, H, and Λ that best meets their most relevant concerns, subject to any constraints they may have. Such concerns include:

1. **Implementation feasibility.** What is the cost to implement the scheme? What is the cost of, say, image capture, material pre-processing, or consumables, if any?

2. **Data Storage.** How long are the strings being stored per instance? Is there a cost (per bit and/or feature) on discriminatory ability and/or robustness? How might this trade-off be quantified when selecting features to extract or bit string construction methods?

3. **Robustness to expected damage type(s).** How are these damage types determined and tested for? May the damage be simulated, may damaged samples be collected for analysis, or both?

In the following section, an example PUF formulation is provided to illustrate this methodology and how these concerns may manifest in practice.

4.2 Application to 4140-Steel Parts

In this section, a PUF formulation is developed for anti-counterfeiting of 4140-steel parts. The PUF takes as input a challenge instance C of the steel part, computes the resulting string b_C, and then compares this string to the instance's enrolled string b_i computed at enrollment time. For this analysis, it is assumed that micrographs serve as the raw inputs from each instance i to the PUF.

4.2.1 Defining $\Gamma(i)$: Data Collection

A 4140/4142 alloy steel rod, with composition according to standard ASTM A29, was sectioned perpendicular to its major axis, polished, and etched via submersion in a 5% nital solution for 30 seconds. 50 micrographs were taken at different locations on the etched sample surface for use in this analysis as the initial enrolled instances. Each micrograph was captured using an AmScope MU500 5.1MP digital camera at 200X magnification. An example micrograph is provided in Figure 4.1; note the distinct proeutectoid ferrite (bright) and pearlite (dark) phases, which serve as the primary source of randomness leveraged in this PUF formulation.

4.2.2 Defining $\Gamma(i)$: Image Pre-Processing

The micrographs were pre-processed using the open source image processing software, ImageJ Fiji [76]. Each micrograph was converted to an 8-bit greyscale image and then blurred using Guassian blurring ($\sigma = 2$ pixels) to remove excess noise and noticeable lamellae in the pearlite phase. The pre-processing procedure is illustrated in Figure 4.2.

The histograms of pixel intensities for the processed micrographs were leveraged to re-classify all pixels in the blurred image as belonging to one of four phases $\rho = \{\rho_0, \rho_1, \rho_2, \rho_3\}$: the dark pearlite phase ρ_0 (which appears as a dark phase at this magnification with a small Guassian blur), the bright proeutectoid ferrite phase ρ_3,

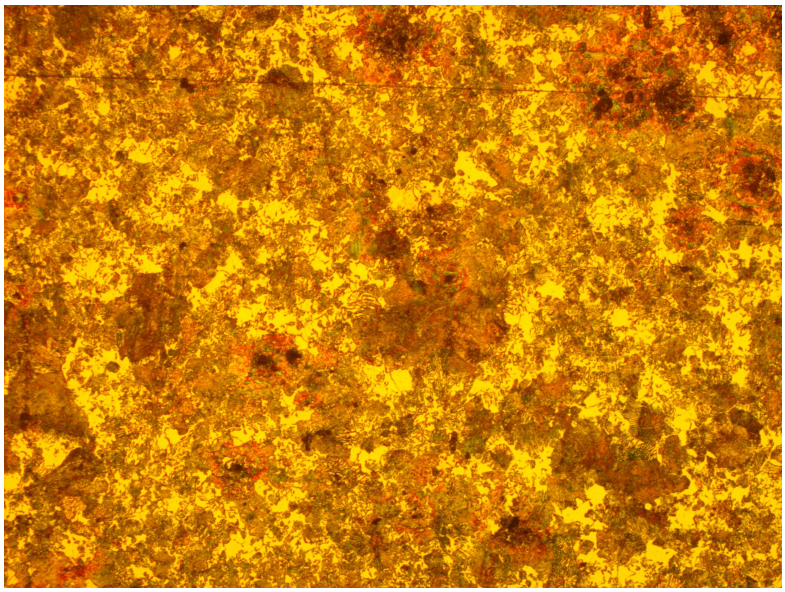

Fig. 4.1. 4140-steel micrograph at 200X. The bright areas correspond to the ferrite phase, while the dark areas correspond to the pearlite phase.

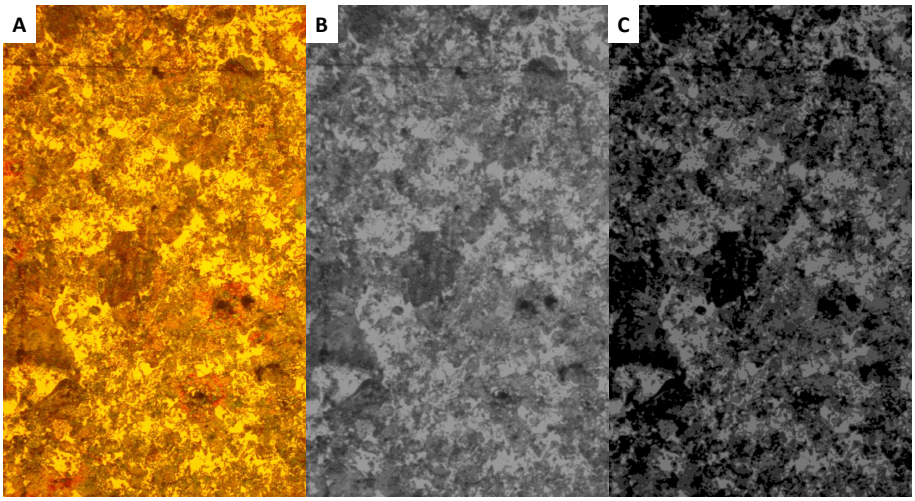

Fig. 4.2. A section of a 4140-steel micrograph at 200X: (A) the raw micrograph, (B) the micrograph after blurring, and (C) the micrograph after pixel classification.

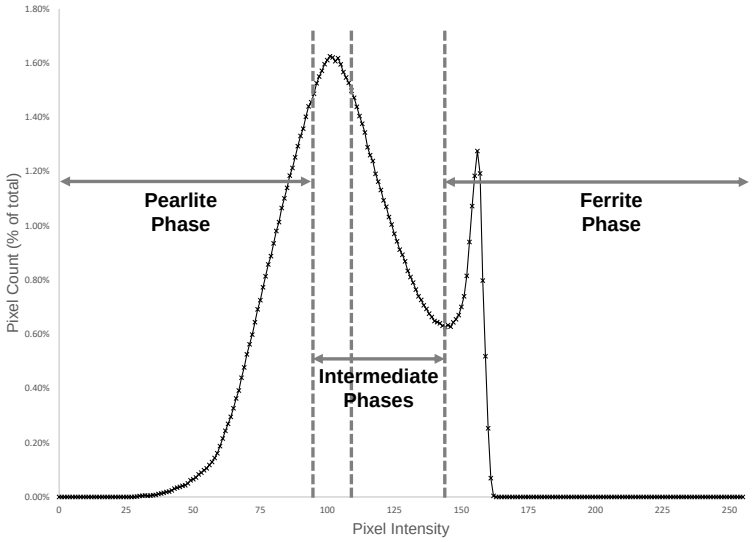

Fig. 4.3. A characteristic histogram of pixel intensities for an 8-bit micrograph after Gaussian blurring. Based on the locations of the two peaks of a micrograph's histogram, each pixel was classified as pearlite, ferrite, or one of two "intermediate" transition phases.

and two intermediate phases ρ_1 and ρ_2 that characterize the transition between the bright and dark phases in the post-processed micrographs. These intermediate phases were defined as those pixels lying within 10 intensity (where for each pixel, the 8-bit intensity $I \in [0, 2^8 - 1]$) of the first histogram peak, and those lying above 10 intensity of the first peak and 20 intensity below the second peak. This was done to more confidently distinguish between the pearlite and ferrite phases, as well as normalize for any illumination changes that may occur between image capture in practice. An illustration of this pixel classification scheme is given in Figure 4.3. An example of a micrograph after pixel classification is given in Figure 4.4.

4.2.3 Defining $\Gamma(i)$: PUF Feature Extraction

Four image slices corresponding to the four pixel classification phases were taken from each processed micrograph. An illustration of the resulting micrograph slices is

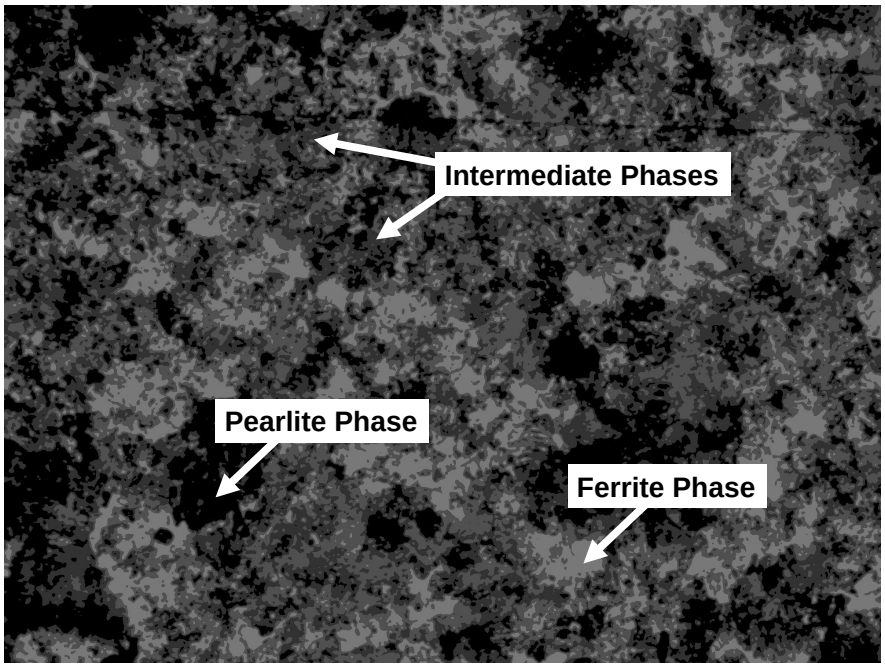

Fig. 4.4. A final processed micrograph with each pixel classified as one of four phases.

given in Figure 4.5. From each image slice, mean phase intercept length L_{3,ρ_n} and phase volume fraction V_{ρ_n} statistics were taken from individual quad-tiled regions of interest (ROIs) following the tiling protocol presented in Chapter 3 [67] with four levels of recursion. These responses were used to generate instance feature vectors v_i. The feature vector for this formulation has the form

$$v_i = \Big[\big((L_{3,\rho_n}, V_{\rho_n})_{\rho_n \in \rho}\big)_{r \in R}\Big], \tag{4.3}$$

where $r \in R$ denotes ROI r in the set R of all ROI's taken from i's micrograph. Since there are four phases for each micrograph, and as in [67] there are 85 ROI's taken per micrograph, $|v_i| = |R| \times |\rho| \times 2 = 680$ feature entries per micrograph.

Phase Intercept Length Extraction. The phase intercept length calculation follows from the lineal analysis approach given in the previous chapter [67] for a brass PUF. To each micrograph slice, apply m uniformly distributed, randomly oriented test lines, each having length L_k, $1 \leq k \leq m$. The mean intercept length is given by Tomkeieff's Equation [79] using the normalized intercept count N_L,

$$N_L = \frac{\sum_{k=1}^{m} P_k}{\sum_{k=1}^{m} |L_k|}, \tag{4.4}$$

where P_k is the number of on-phase–off-phase intersection points on line L_k, and $|L_k|$ is the length of L_k. For each micrograph slice considered, $m = 1,200$ lines were applied with random lengths and orientations. The mean three-dimensional intercept length L_{3,ρ_n} for the given micrograph slice is then

$$L_{3,\rho_n} = \frac{1}{N_L}. \tag{4.5}$$

This calculation was done for each ROI of each micrograph, for each pixel phase slice.

Volume Fraction Extraction. Lineal analysis also allows for the calculation of mean volume fraction. For each of the m lines $L_{k \leq m}$, let $|L_{k,\rho_n}|$ be the length of L_k falling on phase $\rho_n \in \rho$. Then the volume fraction of phase ρ_n, V_{ρ_n}, in the ROI under consideration is

$$V_{\rho_n} = \frac{1}{m} \sum_{k=1}^{m} \frac{|L_{k,\rho_n}|}{|L_k|}. \tag{4.6}$$

This calculation was done for each ROI of each micrograph, for each pixel phase slice.

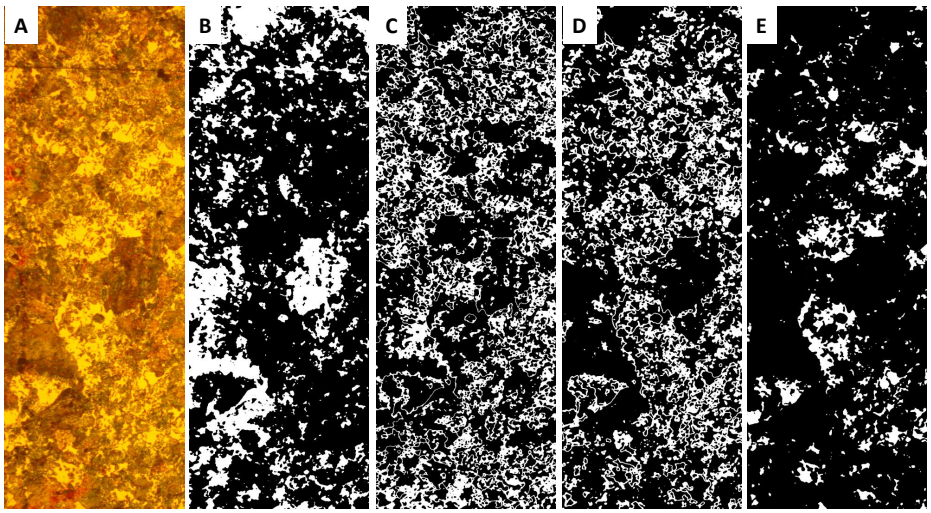

Fig. 4.5. Micrograph slices found through the pixel classification scheme: (A) the original micrograph, (B) the pearlite phase ρ_0 slice, (C-D) the intermediate phase ρ_1 and ρ_2 slices respectively, and (E) the ferrite phase ρ_3 slice.

4.2.4 Defining $g(v_i)$: PUF Bit String Construction

Given v_i, it remains to specify $b_i = g(v_i)$. In the naïve case, one could take v_i directly; that is, $g(v_i) = v_i$, or even simply take the complete micrograph. However, as b_i must be stored by the original manufacturer, or some other organization contracted by the manufacturer, there is an incentive to minimize the length of b_i. To this end, here $g(v_i)$ is specified such that the L_{3,ρ_n} and V_{ρ_n} feature responses for each ROI, for each phase $\rho_n \in \rho$ of i's micrograph, contribute only one bit to b_i. This is done by comparing each feature response to the median of all responses for that phase of that micrograph, as in Algorithm 1.

Note that, if a manufacturer desires a shorter b_i, not all features or phases need be passed to Algorithm 1. Indeed, in the analysis presented in this study, cases where only phase ρ_0, phase ρ_3, phases $\{\rho_0, \rho_3\}$, and all phases ρ are analyzed.

4.2.5 Generating Challenge Instances C

For each micrograph, b_i was found from v_i using Algorithm 1 for each case discussed above. To test the robustness of this method, simulated damage was applied to each of the micrographs in the form of dark striations. Increasing damage severity was modeled as increasing amounts of striations. An example of such a "damaged" micrograph is provided in Figure 4.6. Pixels lying in these striations were assumed to be easily masked out of a micrograph and not considered when taking pixel intensity histograms for pixel classification as in Figure 4.3.

Damage profiles included 4, 8, 12, 16, and 20 striations applied to each of the raw micrographs at random locations. Thus, for each micrograph corresponding to an instance i, there were 5 challenge instances C that represented genuine but damaged instances $\widetilde{i}$. All other instances in the library of micrographs were taken as counterfeit instances j that did not correspond to i.

Data: Feature vector v_i, Phases $\rho \subseteq \{\rho_0, \rho_1, \rho_2, \rho_3\}$

Result: String b_i

initialization;

`/* Initialize` b_i `as an empty string */`

$b_i \leftarrow \{\}$;

for ρ_n in ρ **do**

`/*` $(F_{\rho_n})_r$ `is feature` F`'s value in` v_i`, for ROI` $r \in R$`, for phase` ρ_n `*/`

$\mathrm{ML}_{3,\rho_n} \leftarrow \mathrm{median}\Big(\{(L_{3,\rho_n})_r\}, \forall r \in R\Big)$;

$\mathrm{MV}_{\rho_n} \leftarrow \mathrm{median}\Big(\{(V_{\rho_n})_r\}, \forall r \in R\Big)$;

for r in R **do**

`/* compare each ROI feature value to median over all ROIs */`

if $(L_{3,\rho_n})_r \geq \mathrm{ML}_{3,\rho_n}$ **then**

$b_i \leftarrow b_i || \{1\}$;

else

$b_i \leftarrow b_i || \{0\}$;

end

if $(V_{\rho_n})_r \geq \mathrm{MV}_{\rho_n}$ **then**

$b_i \leftarrow b_i || \{1\}$;

else

$b_i \leftarrow b_i || \{0\}$;

end

end

end

Return b_i;

End

Algorithm 1: Construction of bit string b_i from feature vector v_i and micrograph phases ρ

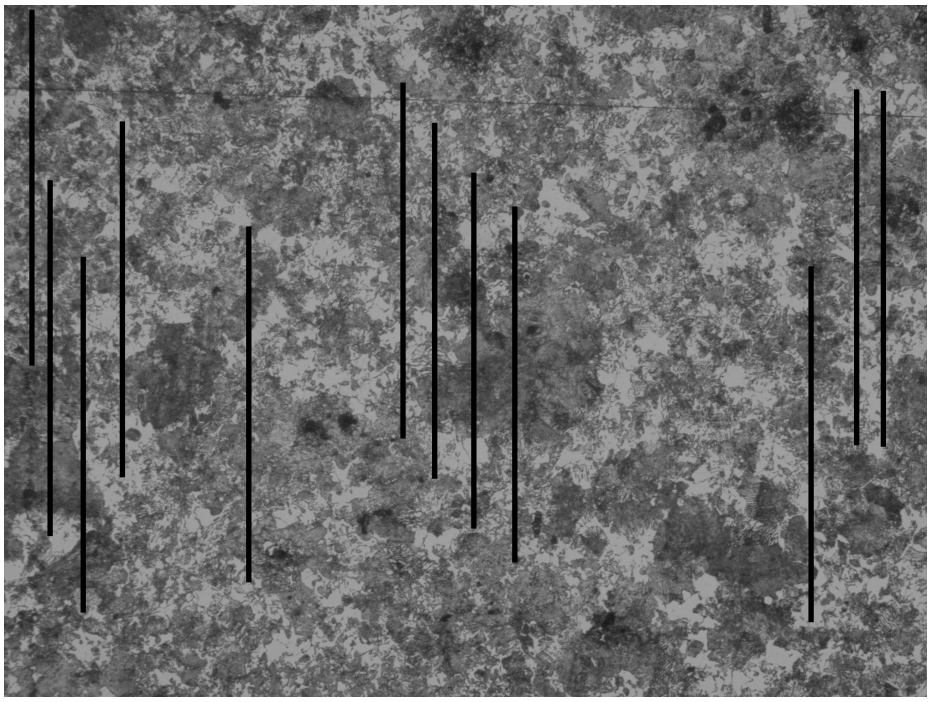

Fig. 4.6. Example of a processed micrograph after 12 striations were applied to the raw micrograph to simulate damage.

4.2.6 Defining $H\big(b_C, b_i\big)$: String Distances

The distance function H is defined between two strings b_i and b_C using a modified Hamming distance measure as in previous work [67] discussed in Chapter 3, but extended to account for an arbitrary feature vector extracted from the instance during the PUF protocol. Recall that for each enrolled instance i and challenge instance C, strings b_i and b_C are generated. For strings b_i and b_C of length L and entries $(b_i)_n$, $(b_C)_n$, $1 \leq n \leq L$, taken from images tiled with K ROI recursion levels [67], the proposed string distance function H is given as in Chapter 3 by

$$H(b_i, b_C, K) = \frac{\sum_{n=1}^{L} q\big((b_i)_n, (b_C)_n, k_n\big)}{\sum_{n=1}^{L} 4^{K-k_n}}, \tag{4.7}$$

where $q\big((b_i)_n, (b_C)_n, k_n\big)$ is the bit-wise comparison function on the bit string entries, k_n is the corresponding ROI recursion level of entry n, $0 \leq k_n \leq K$, and $\sum_{n=1}^{L} 4^{K-k_n}$ is a normalizing term ensuring $0 \leq H \leq 1$. $q((b_i)_n, (b_C)_n, k_n)$ is given by

$$q\big((b_i)_n, (b_C)_n, k_n\big) = \begin{cases} 4^{K-k_n}, & (b_i)_n \neq (b_C)_n \\ 0, & (b_i)_n = (b_C)_n \end{cases} . \tag{4.8}$$

Note that in this case, $K = 3$ total recursion levels. Also note that H is defined in such a way that dissimilar bits corresponding to features gathered from larger ROIs by area are penalized more heavily than those gathered from smaller ROIs. $H = 0$ implies perfect agreement between b_i and b_C, while $H \to 1$ implies increasing dissimilarity between b_i and b_C.

4.2.7 Defining $\Lambda(H)$: Challenge Instance Classification

Now for any enrolled instance i and challenge instance C, a distance H may be calculated between them. From these distances, a simple one-feature likelihood ratio function $\Lambda(H)$ may be applied to classify the challenge as genuine or counterfeit. That is, letting $C = \widetilde{i}$ indicate a genuine but possibly damaged challenge instance

and $C = j$ indicate a counterfeit challenge instance, and assuming both events are equally likely a priori,

$$\Lambda(H) = \begin{cases} \widetilde{i}, & p(H|C = \widetilde{i}) \geq p(H|C = j) \\ j, & p(H|C = \widetilde{i}) < p(H|C = j) \end{cases}, \tag{4.9}$$

where $p(H|C = \widetilde{i})$ and $p(H|C = j)$ are the probability density functions of the distance H given a genuine and counterfeit part instance, respectively.

To generate functional forms for these density functions, it is necessary to generate training data. Here, the experimentally gathered micrographs and their corresponding strings are used for this training. Call all distances $H(b_C||b_i, C = j)$ the *inter-distances* between different enrolled instances, and all distances $H(b_C||b_i, C = \widetilde{i})$ the *intra-distances* between an enrolled instance i and the same instance after simulated damage is applied. Functional forms of $p(H|C = j)$ and $p(H|C = \widetilde{i})$ were found by fitting the experimentally determined inter- and intra-distances to normal and log-normal distributions, respectively. Specifically, it is assumed that

$$p(H|C = j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{H - \mu}{\sigma}\right)^2 \right), \tag{4.10}$$

where μ and σ were found by fitting the experimental inter-distances, with no applied damage, to $p(H|C = j)$, and

$$p(H|C = \widetilde{i}) = \frac{1}{H\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\log\left(\frac{H}{s}\right)}{\sigma}\right)^2 \right), \tag{4.11}$$

where the shape parameter σ and scale parameter s were found by fitting the experimental intra-distances to $p(H|C = \widetilde{i})$ for each damage severity considered.

To estimate the probability of classification error, note that C is mis-classified if $\beta = \Lambda(H)$ does not match C's identity ($C = j, \beta = \widetilde{i}$, or $C = \widetilde{i}, \beta = j$). Also note that, since this classification is done via a likelihood ratio, for the given training data

there will be a threshold $H = \tau$ such that $H \leq \tau \Rightarrow \beta = \widetilde{i}$, and $H > \tau \Rightarrow \beta = j$. Thus the estimated probability of error, $P(\text{error})$, is given by

$$\begin{aligned} P(\text{error}) = P(C = j) \int_0^\tau p(H|C = j)dH \\ + P(C = \widetilde{i}) \int_\tau^1 p(H|C = \widetilde{i})dH, \end{aligned} \tag{4.12}$$

where as before, $P(C = j)$ and $P(C = \widetilde{i})$ are both assumed to be 0.5 a priori.

4.3 Implementation and Results

Inter- and intra-distance plots derived from strings constructed using individual pearlite and ferrite phase data, pearlite and ferrite phase data combined, and all phase data combined are given in Figure 4.7. Note that while the distance plots for the strings constructed utilizing data from multiple phases show greater separation, and therefore imply greater confidence in classifying a challenge part instance as genuine or counterfeit, these strings are longer than the strings derived from the data of individual phases.

The probability of error for strings constructed using various phase data, along with the string sizes and fitted parameters for the distributions of inter- and intra-distances, for each damage profile are summarized in Table 4.1. Note that the probability of error increases with increased damage (i.e., more striations applied) for all construction methods considered. This is an intuitive result, indicating that there is a higher probability of mis-classification with greater difference between i and $\widetilde{i}$'s micrograph.

The estimated probability of error for the case where all phases are considered when constructing b_i is the lowest across all damage profiles as expected, although this comes at the cost of more data per instance to store in the form of the enrolled string. Interestingly, the ferrite phase construction out-performs the ferrite-and-pearlite construction for the 4- and 20-striation damage cases with respect to error probability, indicating that including pearlite phase information may in fact

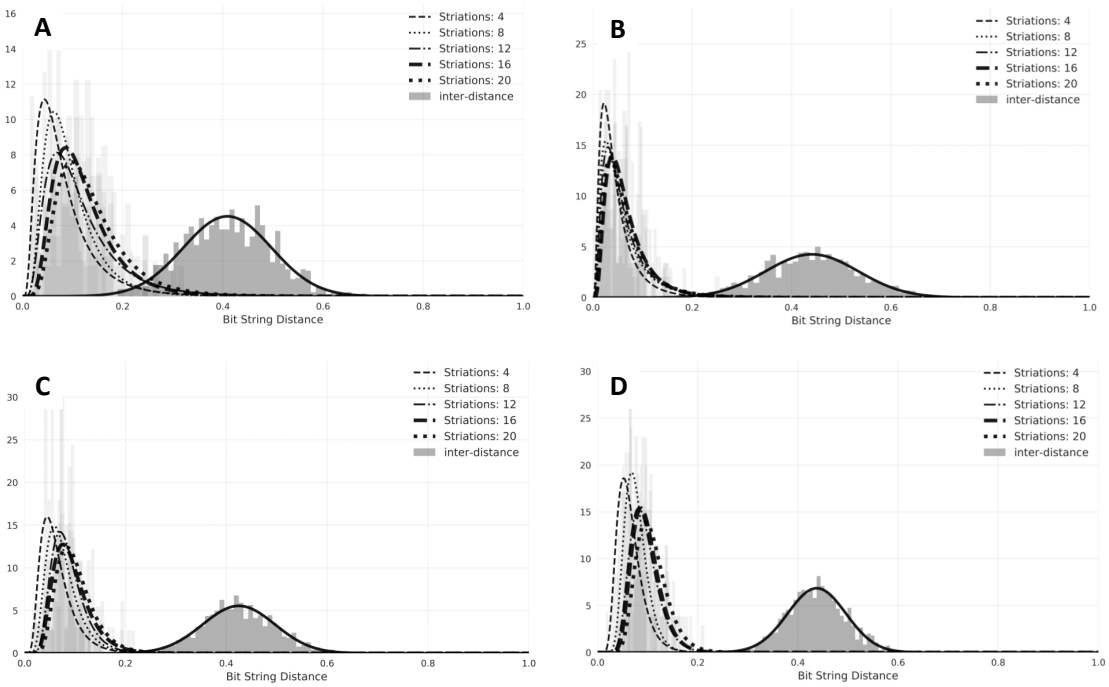

Fig. 4.7. Inter- and intra-distance histogram plots and the fitted distributions for strings constructed using (A) pearlite phase data, (B) ferrite phase data, (C) data from both pearlite and ferrite phases but not the intermediate phases, and (D) data from all four pixel phases. The intra-distance histograms for various damage severities (prominent at low bit string distances H) are fitted to log-Normal distributions, while the inter-distance histograms are fitted to normal distributions.

be introducing confusion into the string construction. This highlights the interdependence between features, expected damage, and string construction that must be studied before committing to a PUF formulation. This also emphasizes the need to study the discriminatory ability of each feature, with respect to damage and other legitimate instances, before committing to a PUF formulation. Of course, the expected damage here is simulated, so the results here may differ from results collected on samples damage by experiment or in practice.

A manufacturer should study the trade-off between string length, discriminatory ability, and robustness to a given damage profile carefully when choosing formulations for $\Gamma(i)$, $g(v_i)$, H, and $\Lambda(H)$. Analyses with respect to different extracted features, damage profiles, and string constructions that proceed similarly to the method presented in this Section would inform designers on the most appropriate PUF formulation for a given scenario.

Table 4.1.
String lengths, distribution fitting parameters, and estimated error probabilities for four string constructions taking features from different micrograph phases.

Phases Considered:		Pearlite	Ferrite	Pearlite and Ferrite	All Phases
String Length (bits):		170	170	340	680
Inter-distance,	μ	4.09 E-1	4.41 E-1	4.25 E-1	4.38 E-1
$H \sim N(\mu, \sigma)$	σ	8.82 E-2	9.37 E-2	7.19 E-2	5.82 E-2
Intra-distance,	σ	6.60 E-1	7.19 E-1	4.92 E-1	3.82 E-1
$H \sim \text{logN}(\sigma, s)$,	s	6.75 E-2	3.75 E-2	5.72 E-2	6.04 E-2
damage = 4 stri.	P(er)	8.09 E-3	1.65 E-3	1.90 E-3	7.87 E-4
Intra-distance,	σ	5.94 E-1	6.99 E-1	3.70 E-1	3.02 E-1
$H \sim \text{logN}(\sigma, s)$,	s	9.86 E-2	4.92 E-2	8.11 E-2	8.98 E-2
damage=12 stri.	P(er)	1.53 E-2	3.29 E-3	3.53 E-3	2.10 E-3
Intra-distance,	σ	4.78 E-1	6.68 E-1	3.59 E-1	2.86 E-1
$H \sim \text{logN}(\sigma, s)$,	s	1.23 E-1	5.44 E-2	9.54 E-2	1.05 E-1
damage=20 stri.	P(er)	1.76 E-2	4.02 E-3	5.01 E-3	2.97 E-3

4.4 Conclusion

In any anti-counterfeiting scenario, PUF design should be tailored to the situation at hand. Still, given a class of PUF designs, such as optical PUFs, and a class of implementation scenarios, commonalities should be leveraged. In this chapter, a general framework for designing PUF protocols for anti-counterfeiting in manufacturing was proposed, with an emphasis on developing optical PUFs that leverage surface microstructural information in multiphase materials like steel. This framework was applied to design an optical PUF for 4140-steel parts that leverages phase information to generate discriminating bit strings as identifiers that are robust to scratching damage the part may sustain. The applicability of the method is demonstrated by applying the PUF to a library of 50 4140-steel micrographs and applying simulated scratching damage to generate "challenges" occurring later in time. Better results, in terms of minimal expected classification error, are obtained by considering more features when constructing the identifying bit strings, at the cost of more required storage space per enrolled part instance. Alternatively, more challenge-response pairs can be used per instance, improving results (also at the expense of more storage per enrolled instance).

Limitations of the analysis presented here include (i) PUF application to only one material, 4140-steel, (ii) application of simulated damage profiles, instead of damage profiles gathered through experimentation, and (iii) analysis using samples with similar surface preparation, which may not always be applicable in manufacturing contexts. Still, this analysis represents a modest step towards a discriminating-yet-robust PUF design framework that may be applied to a wide range of metals and expected damage profiles.

In future work, this PUF design framework should be applied to new materials and classes of PUFs leveraging different surface characteristics of part instances. Also, effort should be made to expand the library of potential features that would be useful when enhancing discriminatory ability, robustness to particular damage types, or

ability to apply similar PUFs to different materials. Studies into the performance of various features under specific damage profiles would also help PUF designers when selecting features to include in a PUF formulation in practice. Each of these concerns are discussed in the following chapter.

5. DESIGN FOR TRACEABILITY SCHEMES USING LARGE LIBRARIES OF MICROGRAPH FEATURES

In the previous chapters, case studies have been introduced to illustrate two important points: (i) features extracted from micrograph data do indeed provide a feasible source of randomness for anti-counterfeiting in manufacturing, and (ii) the extraction-to-encoding pipeline may be applied to different material systems (e.g., polygonal grain structures and multi-phase structures). The discussion has been limited to the study of a relatively small pool of potential features based on accepted micrograph analysis methods and extracted from multiple regions of interest of the part, and the encoding schemes for generating the bitstrings encoding these features. In this chapter, I instead focus on methods for evaluating the *value* of features drawn from a large library, in terms of the desiderata of *discriminatory ability* and *robustness to expected damage or data transformation*, expanded from my previous study in this area [81,82]. Such analysis enables the automated ranking and selection of features to consider for a given problem context.

5.1 Leveraging Micrograph Data for Traceability

The goal of anti-counterfeiting schemes, PUF-based and otherwise, is to generate representations of part instances that are as unique to an instance as possible (discriminatory), but are insensitive to changes in the part's structure caused by expected or acceptable damage (robust). For this chapter, I consider such representations (strings) constructed from features that can be extracted from surface micrographs of a part. To meet the requirement of discriminatory ability, we desire features that are *representative of the unique structure of a micrograph instance*. To meet the robustness requirement, features should be *insensitive to acceptable after-*

enrollment perturbations in the structure of the micrograph instance. As we shall see, these concerns motivate the feature set investigated in this chapter.

Notably, the feature set discussed in this chapter is built on two-point autocorrelation responses of image windows taken within the micrograph, as well as global and local volume fraction information. I propose using principal component (PC) scores, specifically those corresponding to the autocorrelations of each phase of the input micrograph, to build a high-dimensional feature response for each input micrograph, given a set of training micrographs used to generate the PC transform. From these scores, as well as from features computed using volume fraction information, I propose an automated method for selecting "high-value" features for constructing the identifying string.

I pay particular attention to tailoring this process to produce discriminatory yet robust identifying strings, given expected damage types or imaging limitations when challenging a part's origin. The experimental evaluation of the proposed anti-counterfeiting approach leverages the open-source Materials Knowledge System (MKS) Python package [83] to generate micrograph libraries and the corresponding 2-point correlation and volume fraction data for the study presented in this Chapter, as well as software used for persistent homology computations [84] to extract high-value features from one-dimensional series data. In Section 5.3, I discuss refinements to this process that could enhance performance while controlling storage requirements in practical scenarios.

Why Features from Two-Point Statistics? Two-point auto-correlation and cross-correlation statistics are physically meaningful features that can be efficiently extracted from micrograph images where each pixel is assigned to one of N phases [70]. These statistics offer insight into the physically achievable structures of a given material system [85,86], and may be effectively compressed using dimensionality reduction techniques such as principal component analysis (PCA). These compressed representations have been used recently to establish highly accurate, computationally efficient structure-property linkages for materials characterization [87,88]. There is a demon-

strated utility in the materials science literature for using reduced-dimension 2-point responses (using PCA) to train meta-models used to estimate the homogenized mechanical behavior of a microstructure instance [89]. Recent advancements such as the materials knowledge system (MKS) presented by Kalidindi, Latypov, and coauthors [90–92] formalize much of this work.

5.2 Application

The method discussed in this study was implemented and evaluated using a synthetic micrograph data set, with the data sampled using the PyMKS software package [83] (and extended to experimental datasets in Section 5.4). Three-phase micrographs were generated to simulate a multi-phase system as observed in alloys like steels, with two distinct "island" phases and one intermediate phase. The parameters used to generate this data set were chosen such that the resulting micrographs resembled 4140-steel micrographs gathered in lab for use in a previous study [68] (see Chapter 4). A qualitative comparison between the steel data and the generated data is given in Figure 5.1.

The parameters used for data generation are summarized in Table 5.1, and representative example micrographs are shown in Figure 5.2 (A) and (B). As shown, the data took the form of two-dimensional arrays where each pixel was assigned one of three phases, $N \in \{0, 1, 2\}$; these arrays may be thought of as microstructure function instances as described by Fullwood and coauthors [70], characterized by the parameters in Table 5.1.

5.2.1 Damage profiles

Before discussing the implementation of the method, we first turn our attention to how the method will be evaluated in this study. For evaluation and robustness testing, several damage profiles were simulated. Each profile was applied to the library of micrographs used for training and testing separately, and results for each

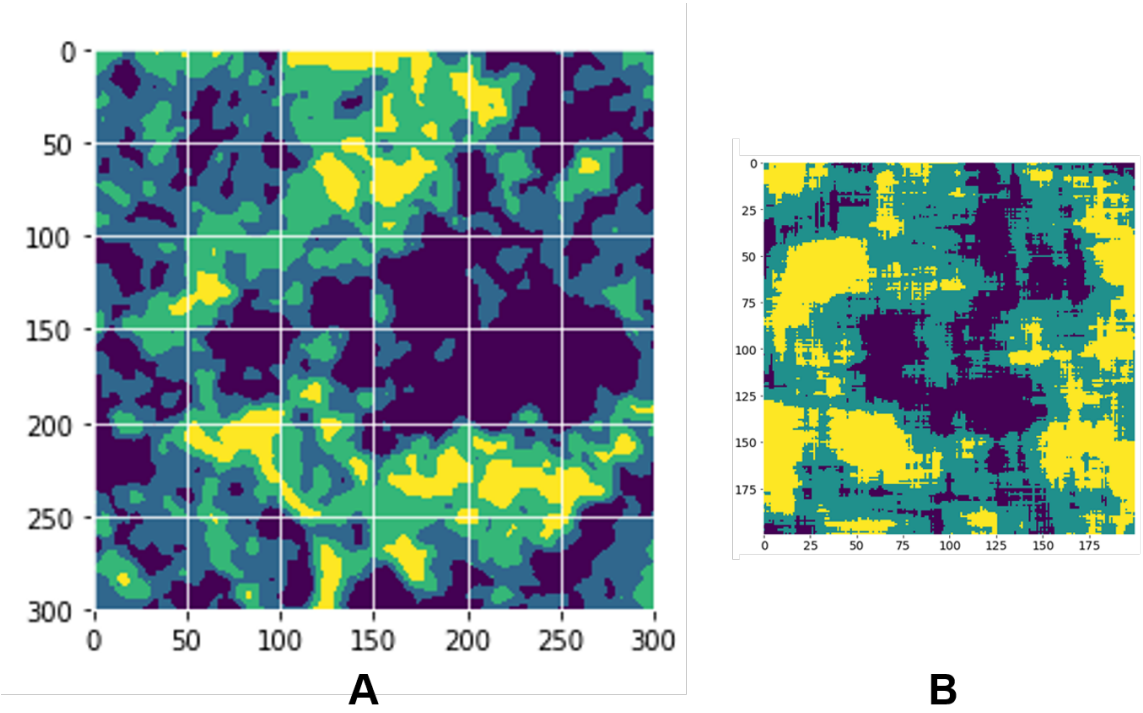

Fig. 5.1. (A) Pre-processed 4140-steel micrograph example used in [68], image approx 0.6mm x 0.6mm. (B) Example of a micrograph generated for use in this study.

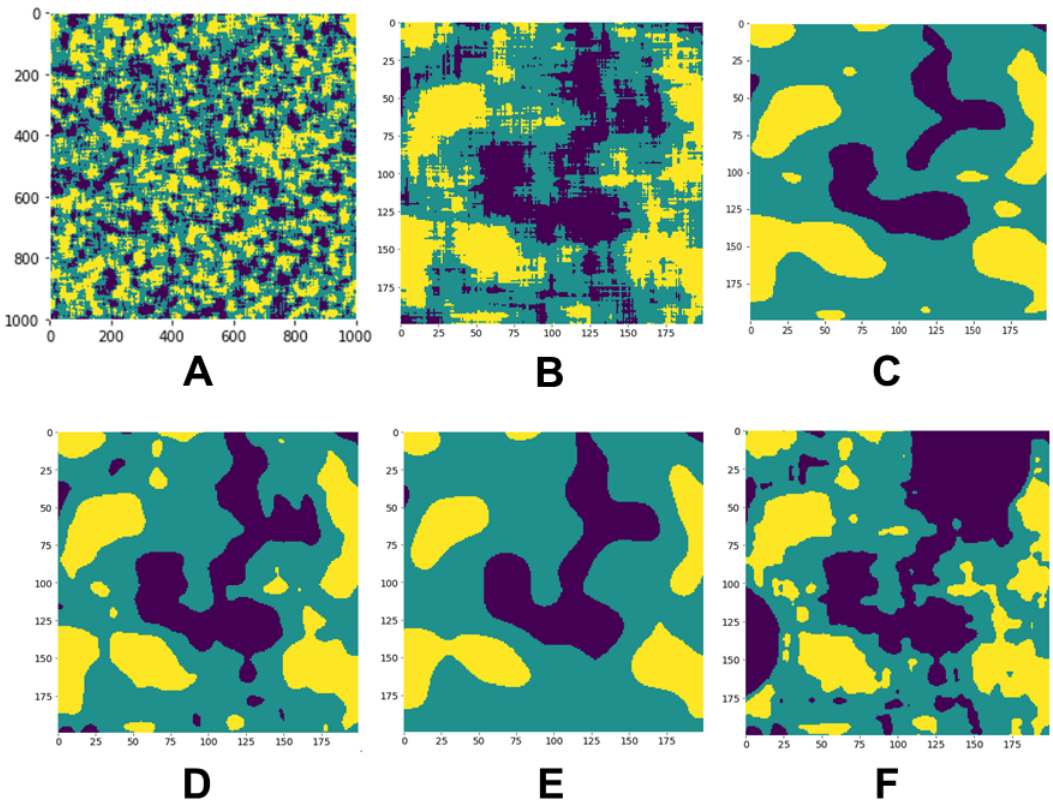

Fig. 5.2. (A) Example three-phase generated micrograph. (B) 200 x 200 pixel window taken within the micrograph. (C) Damage profile 1, (D) Damage profile 2, (E) Damage profile 3, (F) Damage profile 7.

Table 5.1.
Parameters used for data generation with PyMKS software package [83].

Parameter	IPCA Training Data	Input Data	Units
count	2000	300	micrographs
IPCA batch size	50	N/A	micrographs
micrograph dimensions	1000x1000	1000x1000	pixels
number of phases	3	3	
phase 0 volume fraction	0.2	0.2	
phase 1 volume fraction	0.5	0.5	
phase 2 volume fraction	0.3	0.3	
allowed variance in VF	5%	5%	
tiling window dimensions	200x200	200x200	pixels
phase 0 average grain size, x	30	30	pixels
phase 0 average grain size, y	30	30	pixels

of the profiles are compared in Section 5.3. Note that the profiles were selected either to (i) simulate poor challenge imaging quality or loss of high-frequency phase data over time through Gaussian blurring of varying intensity (profiles 1-3), (ii) to simulate mis-aligned imaging or translation transformations of the microstructure when comparing the original "enrolled" response to the challenge (profiles 4-5), or (iii) to simulate mechanical or chemical damage taking the form of pitting (profiles 6-7). The profiles are summarized in Table 5.2, and those profiles involving blurring and pitting are illustrated in Figure 5.2 (C) through (F). Note that when blurring was applied, each pixel was assigned the phase closest to the resulting pixel value to ensure all locations in the sample were assigned to exactly one phase.

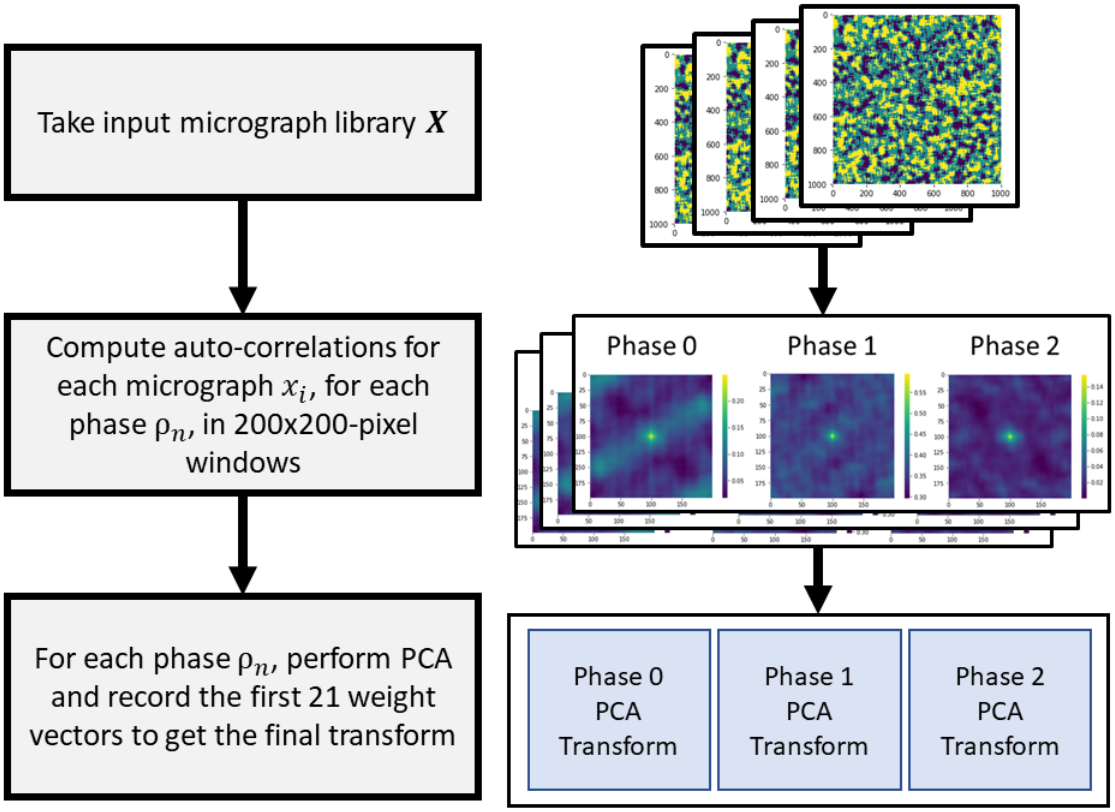

Fig. 5.3. PCA transformation extraction pipeline. Incremental PCA parameters are given in Table 5.1.

Table 5.2.
Damage profiles considered in this study. All units in pixels. For profiles 1 through 7, define $\sigma_G = (\sigma_x, \sigma_y)$.

damage profile	Gaussian Blurring		Translation		Pitting	
	σ_x	σ_y	in x	in y	Pit count	Pit radius
0 (original)	0	0	0	0	0	N/A
1	4	8	0	0	0	N/A
2	4	4	0	0	0	N/A
3	8	8	0	0	0	N/A
4	2	2	20	20	0	N/A
5	2	2	40	40	0	N/A
6	2	2	0	0	15	25
7	2	2	0	0	20	40

5.2.2 Defining $\Gamma(i)$: Feature extraction

Given the input micrograph libraries and damage profiles, we now look at how to extract the features and the subsequent strings from each instance in the micrograph libraries.

PCA training for feature extraction: Building on the previously-discussed motivation for using PCA responses in this study, 2,000 micrographs were initially generated to train the PCA models required. From each of these micrographs, a 200x200 pixel window was extracted. Then, the two-point autocorrelation statistic response, each of dimension 200x100 unique pixels, was calculated for each of these windows. These autocorrelations were calculated following the method of Fullwood and coauthors [70]. These responses were then used to train a PCA model for each of the three phases represented in the micrographs. As these involved large inputs, incremental PCA (IPCA) with a batch size of 50 micrograph windows was implemented using Python to train these models. An illustration of this process is presented in

Figure 5.3. The cumulative explained variances for each of the first 20 PCA scores for each phase are plotted in Figure 5.4; note that near-complete information on the original micrographs is recovered after about 4 PC scores.

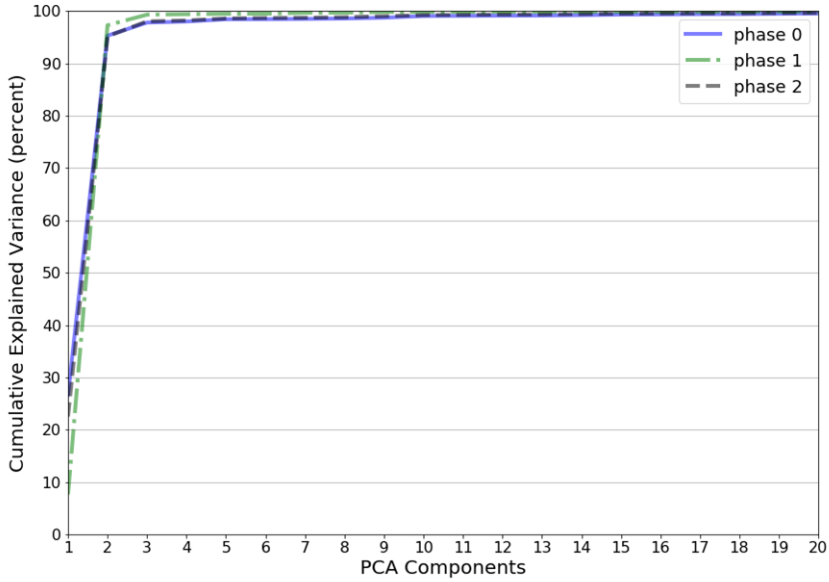

Fig. 5.4. Cumulative explained variance for each phase PCA transform for the first 20 PCA components.

Sliding window feature extraction pipeline: Now, we leverage this PC transform. From micrograph M, we extract features from J windows $m_{1\leq j\leq J}$, where each window m_j is related in space to windows m_{j-1} and m_{j+1}. In this study, windows are taken sequentially along a circle of radius 150 pixels. First, for each window, compute each phase's 200x100-pixel autocorrelation response, and then apply the previously-computed PC transform for each of the three autocorrelations. Thus, each tile provides a set of PC scores, indexed by k, for each of the three phases. If this is done for each tile along the path of the sliding window, then the result is a series-of-series of PC scores for each phase; these series may be used for further feature

extraction explained below. An illustration of this sliding-window-path approach is given in Figure 5.6 (A) to (B).

Let the PC score series corresponding to micrograph M, phase ρ, PC score k, be denoted $C^k_{M,\rho}$. It was observed during this study that common damage profiles, like scratching or pitting, can shift PC scores within the tiles affected an intolerable amount. But, the *difference* in PC responses between tiles was, for many features, more robust. Define the series $\Delta C^k_{M,\rho}$ such that

$$\left(\Delta C^k_{M,\rho}\right)_n = \left(C^k_{M,\rho}\right)_{n+1} - \left(C^k_{M,\rho}\right)_n, \tag{5.1}$$

then $\Delta C^k_{M,\rho}$ is the series of window-to-window differences in PC scores with a length one less than $C^k_{M,\rho}$. A robust feature extracted from a series of PC scores should be robust to these events if they occasionally occur in the data.

For this analysis, features were extracted from both the $\Delta C^k_{M,\rho}$ and $C^k_{M,\rho}$ series, and for each feature, the method that yielded the highest value during training, as defined later in Section 5.2.4, was used for the relevant extraction at test time. The relative rankings of features according to this value, for different damage profiles, are plotted as violin plots in Figure 5.5. In Figure 5.5, the y-axis of each plot denotes the rank of the feature according to feature value as computed in Section 5.2.4, and the x-axis denotes whether the "original" $C^k_{M,\rho}$ or "differenced" $\Delta C^k_{M,\rho}$ series was taken for that feature after training. Note that for some damage profiles, such as the offset profiles, "original"-series based features cluster at the preferred lower ranks, while for the blur and pitting profiles, the "difference"-series features are more prevalent at the lower rankings. This suggests there is value in investigating the preferred series to use for each feature during the training phase, and recording those choices for testing and application.

Now, we want to construct a set of features from the appropriately chosen list of series, built on the de-correlated PC scores of the autocorrelation responses. There are many feature extraction methods that could be used here; in the proposed approach, I take the five most dominant frequencies of the Fourier transform (FT) of each series after 0-padding, along with their relative persistence as discussed in literature on

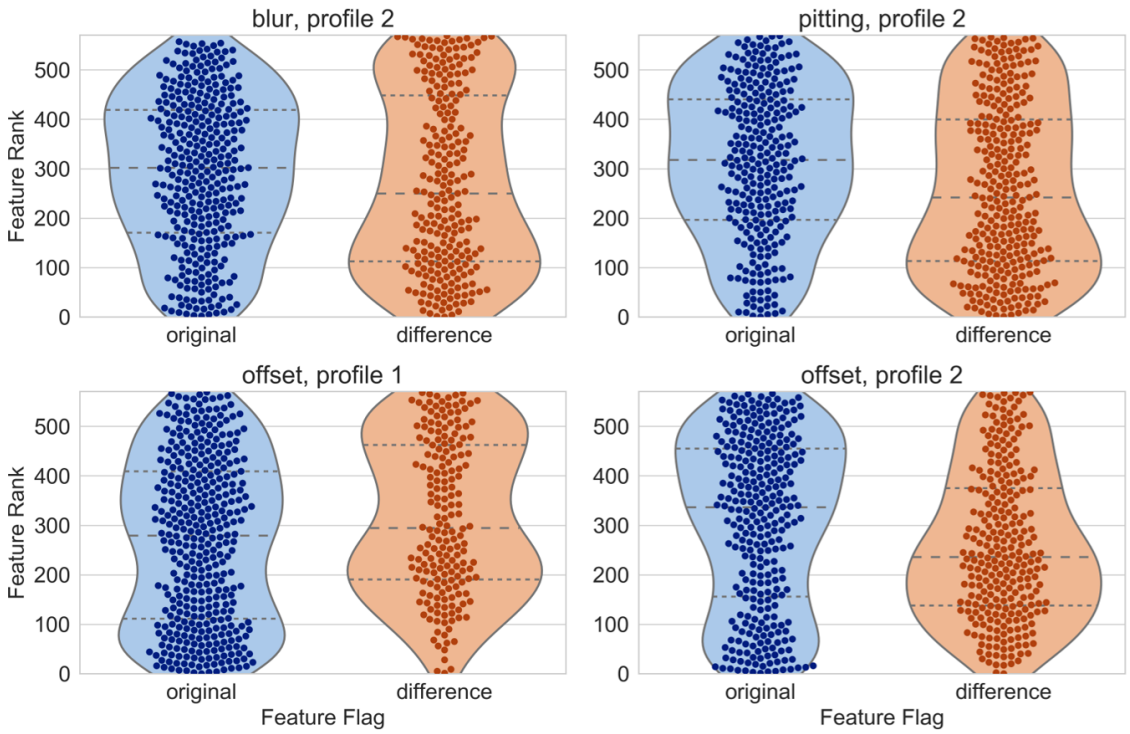

Fig. 5.5. Violin plots displaying ranks of features after selecting between features derived from original and differenced series. Lower rankings are preferred.

topological data anaylsis [84, 93–95]. If each series $\Delta C^k_{M,\rho}$ (or $C^k_{M,\rho}$) has N entries indexed by n, then the entries of the Fourier transform are

$$\left(F^k_{M,\rho}\right)_n = \sum_{v=0}^{V} \left(\Delta C^k_{M,\rho}\right)_v e^{-2\pi i n v/V}. \tag{5.2}$$

Since we only want to deal with real numbers, I consider the modulus of each entry, $|F_n|$, for this analysis. The transform was computed using the Fast Fourier Transform (FFT) algorithm implemented in Python's numpy library.

The peaks of $|F|$ are found using principles in the literature on topological data analysis (TDA) [94, 95], particularly persistent homology. For each FT, the "bar-code" of the signal is computed, which records the most significant peaks of the 1-D signal and a measure of their relative importance, called the persistence, in a noise-resilient way [93]. The locations and persistence scores of the first 5 peaks are recorded as features, such that each FT supplies 9 features to the total feature list (the first peak's relative persistence is always 1, and so is not useful). The same peak location/significance feature extraction method was used to extract additional features from series of local volume fractions for each phase in each tile, yielding another $3 * 9 = 27$ features per path. As the entries in these series are simply the volume fraction of the corresponding phase, no PC transform is needed to gather these features.

The peak and persistence results were calculated using the algorithm presented by Edelsbrunner [95] and implemented in the libstick software package [84]. A representative result is shown in Figure 5.6(D), with blue bars and their heights corresponding to the locations and order of significance of the persistent peaks, respectively. This method has the advantage of extracting noise-resilient features, at the expense of some correlation between the locations and persistences of the recorded peaks, as can be observed in Figure 5.7 (A) and (C). The effect of these correlations on scheme performance can be significant, and mitigation strategies for dealing with these correlation effects are discussed in greater detail below.

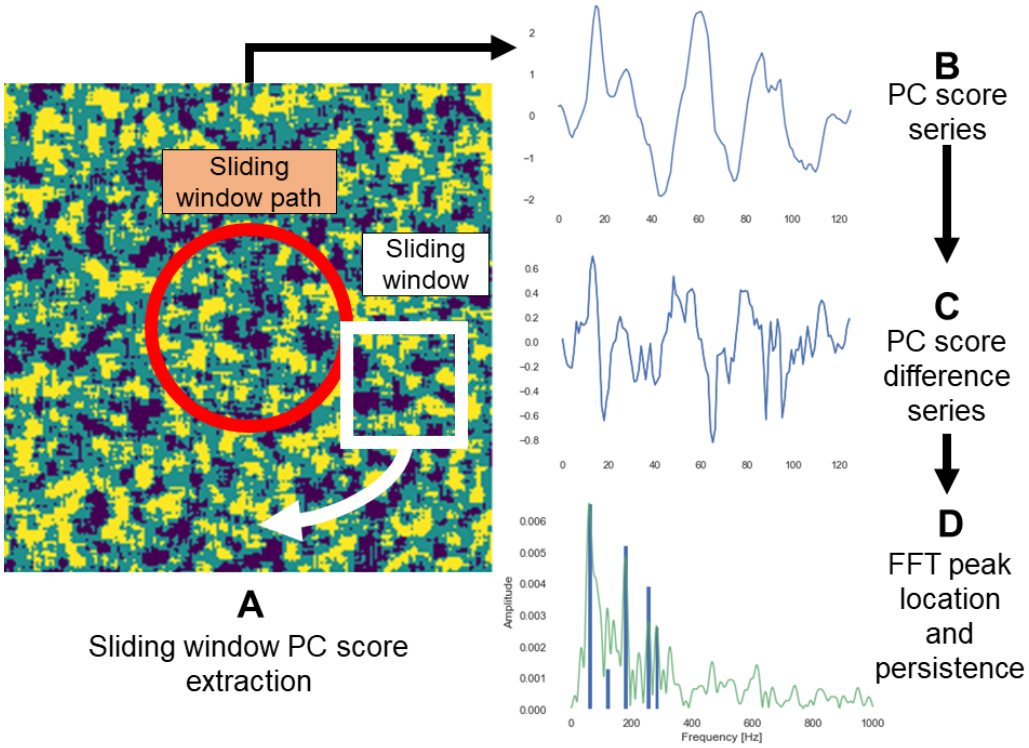

Fig. 5.6. Feature extraction pipeline with representative data. The use of PC score series or difference series for final feature extraction is determined for each feature during training. (A) Example of a micrograph window m taken along a circular path. For each window, compute each phase's autocorrelation. (B) The PC scores are computed for each window using the transform for each phase, and series of these scores are computed by concatenating each window response for each phase, for each PC score. (C) The differences in these scores forms another series, which is more robust to common damage types. (D) The FFT of (C) is computed and the most significant peaks and their persistence scores are recorded as features (order represented by the height of the blue bars for illustration).

5.2.3 Feature characterization

It is worth commenting on how these features may be characterized in the context of anti-counterfeiting. As discussed above, features are constructed from the FT response of the PC and volume fraction responses along a "series" of smaller micrograph windows. Features correspond to (i) a phase N, (ii) a corresponding PC score or volume fraction for that phase, (iii) the location of the corresponding significant peak of the FT response for that response, and (iv) the relative significance of that response, if the peak is not the first-most-significant. This yields $9 \times 3 \times 20 = 540$ correlation-related features. As I also extract features for the three global volume fractions and for the local volume fraction series for each input, I add an additional $3 + 9 \times 3 = 30$ features. So, in total, for this analysis 570 features are extracted from every input micrograph. Here, I discuss the desired properties of the extracted features and analyze a subset of these features for one damage profile for illustration.

Now, we know it is important to analyze the discriminatory ability and robustness of each feature, as this relates to the feature's usefulness when included in the full micrograph string for anti-counterfeiting. Intuitively, features with low correlation with other features in the set should have higher information carrying potential, enhancing discriminatory ability while reducing the number of other feature responses that need to be stored. Also, features with low expected *differences between response* before and after a damage profile is applied are desired, as this indicates that feature is robust to that damage profile. It is also desired that the differences between one feature response before and after damage have low correlation with the differences of other features, as this implies that if one feature is changed due to damage, other features are not any more likely to change with it.

As an illustration, feature correlation matrices containing data on 100 of the extracted features for damage profiles 2 and 5 are plotted in Figures 5.7 (A) and (C), along with the correlation matrices of the before-and-after feature differences in Figures 5.7 (B) and (D). Note that the features were ordered using the value metrics

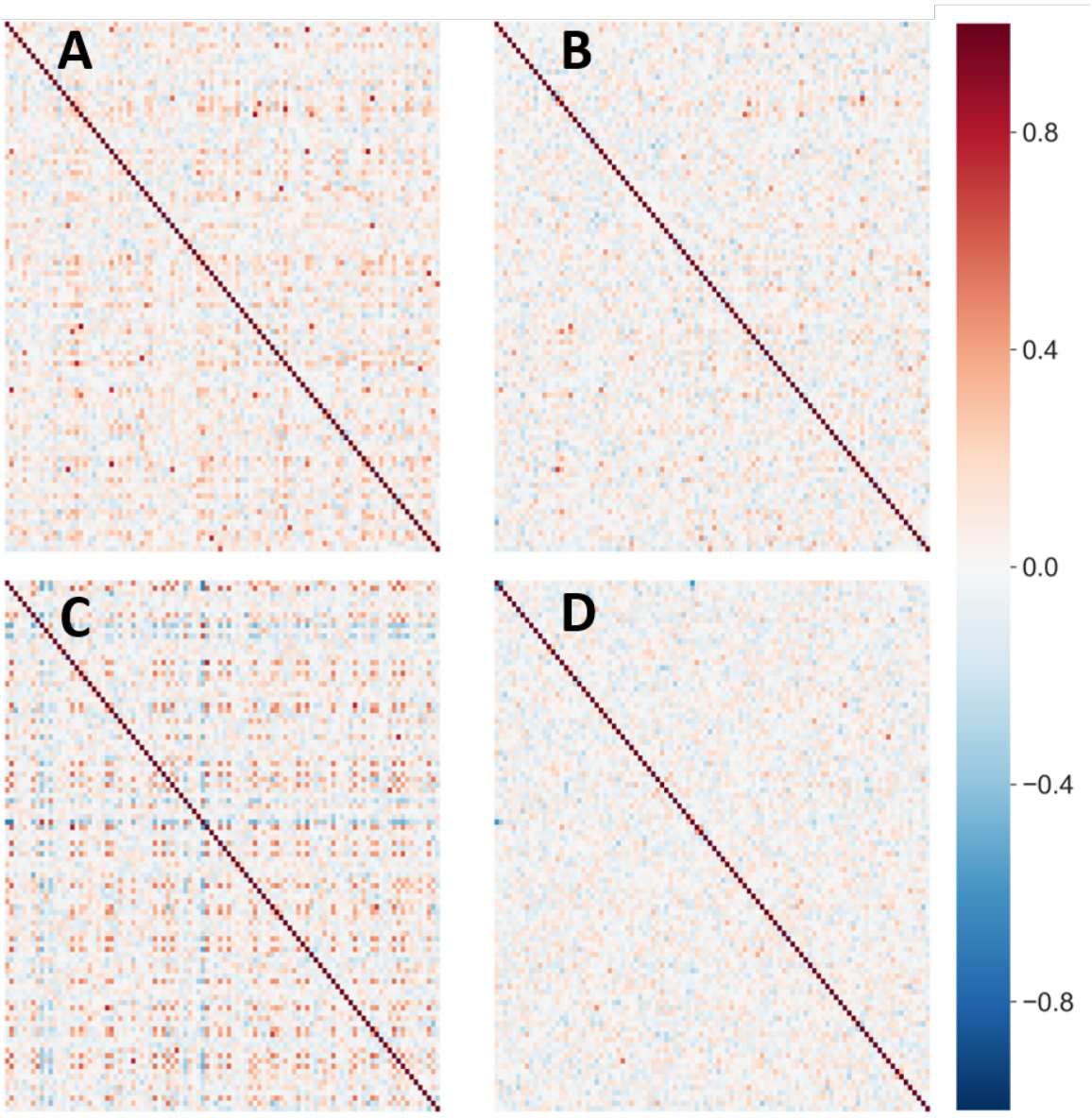

Fig. 5.7. Pearson correlation matrices of feature responses of the 100 highest-value features for the enrolled data under blur profile 2 (A) and offset profile 2 (C), and correlation matrices of the differences in feature responses before and after damage for blur profile 2 (B) and offset profile 2 (D).

discussed in the next section when considering exposure to damage profiles 2 and 5. For damage profile 2 in particular, most high-value features display low correlation with other features; this may not always be the case if feature correlation is not explicitly considered when ordering features.

For most features reported, before-and-after feature differences in Figure 5.7(B) and (D) have low correlation. This implies that feature differences due to damage should be fairly independent of each other, which is desired. However, note that correlation between features introduces redundant information in the features used to generate the final identifying string, needlessly extending the string length and hampering performance. This issue is addressed in this chapter by penalizing high pair-wise mutual information between features, estimated from training data. This will be discussed in greater detail in the following subsection. With these characterizations in mind, let us continue and construct the output strings from these features.

5.2.4 Feature selection and bit string construction

From this feature set, I construct a representative string for an input micrograph as follows. For each feature y_i, take the histogram of responses from the enrolled data as discussed in the previous subsection. For the number of bins to be recorded, segment the histogram according to the estimated cumulative distribution. Then, assign the input micrograph string an integer corresponding to the matching bin for that feature. This generates strings of length (number of bins-1)*(number of features) bits, assuming each feature has the same number of bins. Let B denote the number of bins minus 1, and let N denote the number of features considered in the string. Here, I will set $B = 7$ so each feature integer takes a value from 1 to 8. This scheme is illustrated in Algorithm 2.

Computing distances between strings: To evaluate this scheme, we must define a method to compute distances between the generated strings. This distance, then, captures our estimate of the dissimilarity between the input micrographs the

Data: Desired feature responses $y_i \in Y$, cumulative distribution functions for each feature $F_i(y_i)$, desired number of histogram bins for each feature $B + 1$.

Result: String s

`/* Initialize` s `as an empty string */`

$s \leftarrow \{\}$

`/* Assign each` y_i `an integer identifier using corresponding CDF */`

for y_i in Y **do**

  for k in $\{1, ..., B + 1\}$ **do**

    if $\frac{k-1}{B} \leq F_i(y_i) < \frac{k}{B}$ **then**

      `/*` b_i `is assigned` k`. */`

      $b_i \leftarrow k$

      break

    end

  end

  `/* Append` b_i `to string` s `*/`

  $s \leftarrow s||b_i$

end

Return s

End.

Algorithm 2: Construction of string s from features $y_i \in Y$. The final string is the concatenation of b_i for each desired feature for the input micrograph.

strings represent. In the proposed approach, I use a modified Manhattan distance metric that compares each feature represented in the strings being compared, and adds to the distance based on how dissimilar those responses are to each other. If s_1 and s_2 are the strings being compared, and $(b_i)_1$ and $(b_i)_2$ are the corresponding integers in each string for feature i, then call this distance $D_{MH}(s_1, s_2)$ with

$$D_{MH}(s_1, s_2) = \frac{1}{BN} \sum_{i=1}^{N} \big|(b_i)_1 - (b_i)_2\big|, \tag{5.3}$$

where $\frac{1}{BN}$ is a normalizing term. As D_{MH} compares strings feature-wise, $D_{MH}(s_1, s_2) = 0$ implies $s_1 = s_2$, and so the input micrographs are nearly or even exactly identical, while $D_{MH}(s_1, s_2)$ close to 1 implies the features represented in the strings, and therefore the corresponding micrographs, are highly different from each other. Note that for the remainder of this chapter I denote by $\bar{b_i}$ the vector of the feature's *integer* bin values as computed by Algorithm 2 for all training data, while I denote by $\bar{y_i}$ the corresponding vector of the feature's values as computed during training, which is assumed to be continuous.

The distribution of distances over input strings computed using different micrographs is the so-called inter-distance distribution (as discussed for case studies presented in Chapte 4), and is assumed to be the distribution the distance will be drawn from when the challenge part is a counterfeit. Similarly, the distribution of distances over inputs of the same micrograph before and after some damage is applied is the intra-distance distribution, and is assumed to represent the distribution of distances between strings generated by the same micrograph.

Estimating feature "value": The "value" of each feature may be thought of as its contribution both to the discriminatory ability and the robustness to expected damage or information loss of the output string. Consider each feature response as having a discriminatory score, $v_d(\bar{b_i})$, and a robustness score $v_r(\bar{b_i})$. One natural way to compute $v_d(\bar{b_i})$ would be to take the average normalized difference between string

responses for feature y_i across micrographs in the data set corresponding to different part instances. So, let

$$v_d(\bar{b_i}) = \frac{1}{B|N_{\text{diff}}|} \sum_{(a,b)\in N_{\text{diff}}} \big|(b_i)_a - (b_i)_b\big|, \tag{5.4}$$

where N_{diff} is the set of all pairs of strings taken from different micrographs and $|N_{\text{diff}}|$ is that set's length. This may be thought of as an estimate of the inter-distance of strings containing only this feature, which we want to maximize. Similarly, $v_r(\bar{b_i})$ may be computed by taking the average normalized difference when considering strings corresponding to the same part instances before and after damage. So, let

$$v_r(\bar{b_i}) = 1 - \frac{1}{B|N_{\text{same}}|} \sum_{(a,b)\in N_{\text{same}}} \big|(b_i)_a - (b_i)_b\big|, \tag{5.5}$$

where N_{same} is the set of all pairs of strings taken from the same micrograph before and after a specified damage profile. Note that here I subtract the distance average from 1: it is more valuable to minimize this distance average, as this average is an estimate of the intra-distance of the feature y_i and we want distances between feature strings before and after acceptable damage to be small. Since we want higher $v_r(\bar{b_i})$ to be desired, subtracting this summation from 1 provides a correct score.

Now, for an input library of micrographs, we can compute each feature's $v_d(\bar{b_i})$ and $v_r(\bar{b_i})$. A linear combination of these may be taken as a combined value score for each feature. For this chapter, I consider the average of the two scores,

$$v_c(\bar{b_i}) = \frac{1}{2}\big(v_d(\bar{b_i}) + v_r(\bar{b_i})\big). \tag{5.6}$$

Accounting for feature correlations: As discussed in the previous subsection, the above naive score does not consider the redundancy introduced by feature-wise correlations. To address this, I add one final penalty term to $v_c(\bar{b_i})$ which captures these effects. For this analysis, I proceed as follows:

I take as the first feature the one with the greatest v_c value as per Equation 6.6. I then take subsequent features one at a time, subtracting a term from each's v_c value that penalizes the estimated mutual information between the current feature's

y_i responses and all features added so far, denoted y_{-i}. Call this term $M(y_i|y_{-i})$. I then take the feature with the highest resulting score as the next feature to add, and continue in this way until the desired number of features have been assigned. This is in essence a modified version of the minimum redundancy maximum relevance (mRMR) feature selection method [96,97], where redundancy is captured by the term $M(y_i|y_{-i})$ and relevance by $v_c(\bar{b_i})$ rather than mutual information with a target variable.

The mutual information between two random variables X and Y, with joint distribution $\mu(x, y)$ and marginal distributions $\mu_x(x) = \int \mu(x, y)dy$ and $\mu_y(y) = \int \mu(x, y)dx$, is defined as (see for instance [98])

$$I(X, Y) = \int \int \mu(x, y) \log \frac{\mu(x, y)}{\mu_x(x)\mu_y(y)} dxdy. \tag{5.7}$$

However, this is generally intractable for continuous random variables, as we have in this case. To estimate the mutual information between two features given response vectors $\bar{y_i}$ and $\bar{y_j}$, I make use of the mutual information regression module of the scikit-learn python package [99]. Note that here, $\bar{y_i}$ and $\bar{y_j}$ denote the actual feature responses from the training data, and not the binned values as given by Algorithm 2. This module implements a k-means clustering-based algorithm for estimating mutual information between two continuous random variables, with the assumption that $(\bar{y_i}, \bar{y_j})$ are samples from the joint distribution of features $\mu(y_i, y_j)$. The algorithm implemented is that proposed by Kraskov and coauthors [98], section 2C. In the original paper, the authors propose using clustering with k=2 to 4 clusters; I use 3 for this analysis.

For two features y_i and y_j, denote the estimate of mutual information, computed from feature vectors $\bar{y_i}$ and $\bar{y_j}$ using the method proposed by [98] and implemented in [99], as $\mathrm{MI}(\bar{y_i}, \bar{y_j})$. Then, the penalty term may be taken as the average of this estimate over all features in the set y_{-i} already added to the string,

$$M(y_i|y_{-i}) = \frac{\sum_{j \in y_{-i}} \mathrm{MI}(\bar{y_i}, \bar{y_j})}{|y_{-i}| - 1}, \tag{5.8}$$

where $|y_{-i}|$ is the set's length.

Again, I choose to take the average of the naive combined value score and the penalty term $M(y_i|y_{-i})$, while noting that for specific implementations, the weights on each may be tuned to optimize performance. The final feature value score is then given by

$$v(\bar{b_i}) = \frac{1}{2}\big(v_c(\bar{b_i}) - M(y_i|y_{-i})\big). \tag{5.9}$$

The ordered feature value scores for each damage profile are plotted in Figure 5.8; the lines plotted for each profile are the results for five-fold cross validation. Responses are colored according to the corresponding damage profile. Scatter plots for feature values of v_d and v_r are provided in Figure 5.9, with points colored according to the phase that feature was computed from and markers denoting the type of feature (2-point statistic or volume fraction-based). Note that the phases contributing the most valuable features (those maximizing both v_d and v_r) change between damage profiles, indicating that engineers must carefully anticipate which damage profiles are expected before committing to a set of features.

5.3 String Construction and Results

Strings were constructed for each micrograph before and after applying each damage profile by taking only the highest-scoring features, up to a specified percent. By estimating the intra-distance and inter-distance distributions on one (testing) fold of the data, after using the other four folds to find value scores for each feature, a maximum-likelihood estimate classifier was constructed to label distances between before-and-after-damage pairs of micrograph strings (s_1, s_2) as genuine (drawn from the intra-distance distribution) or counterfeit (drawn from the inter-distance distribution). If on comparison, it is more probable that $D_{MH}(s_1, s_2)$ was drawn from the intra-distance distribution than the inter-distance distribution assuming equal a priori probability, then s_2 is considered the genuine part that created s_1. Else, s_2 is labeled a counterfeit.

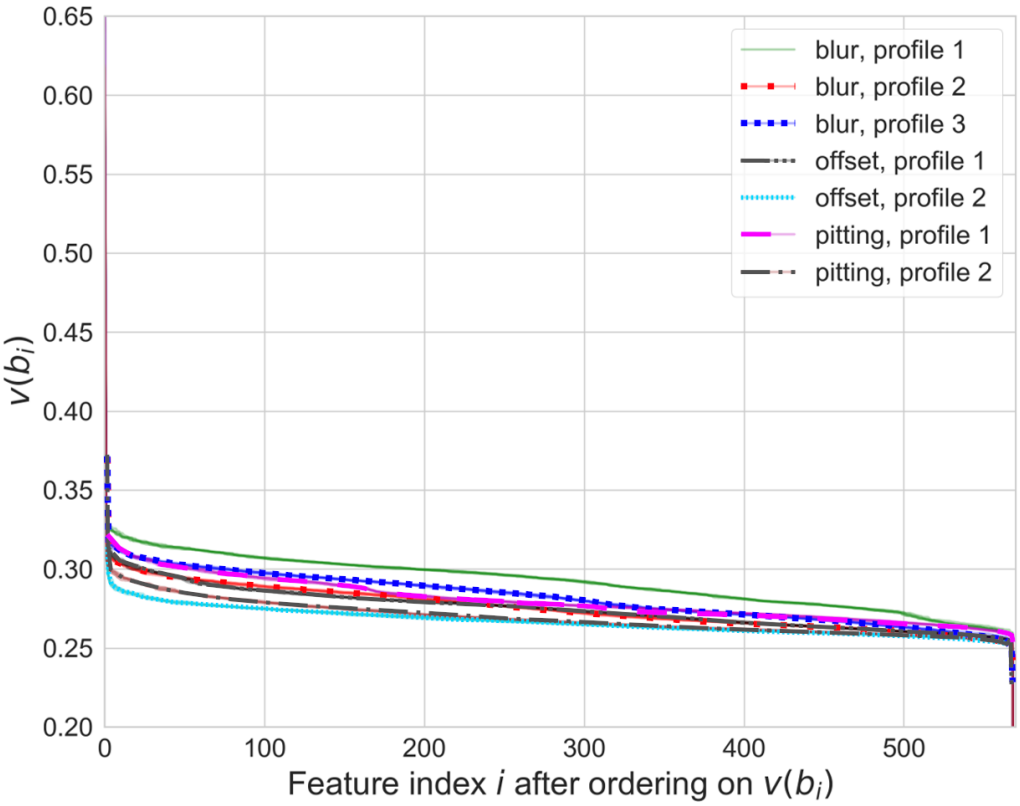

Fig. 5.8. Ordered feature contributions for each damage profile.

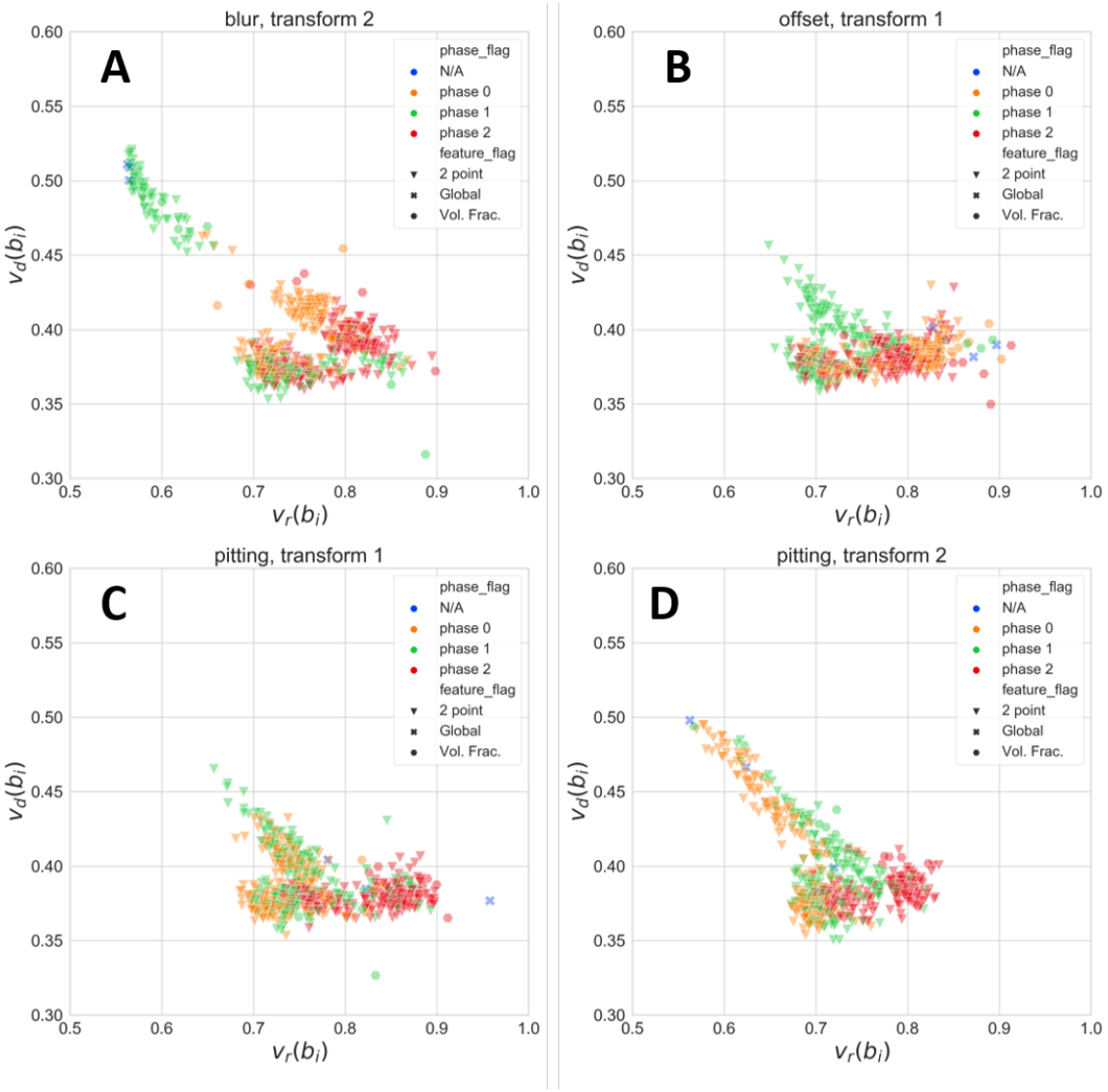

Fig. 5.9. Scatter plots for feature value scores v_d and v_r, for four damage profiles: (A) damage profile 3 (blurring), (B) damage profile 5 (translation), (C) damage profile 6 (moderate pitting), and (D) damage profile 7 (severe pitting). Colors correspond to the feature's corresponding phase, and marker types correspond to the type of feature: 2-point, local volume fraction, or global volume fraction-related.

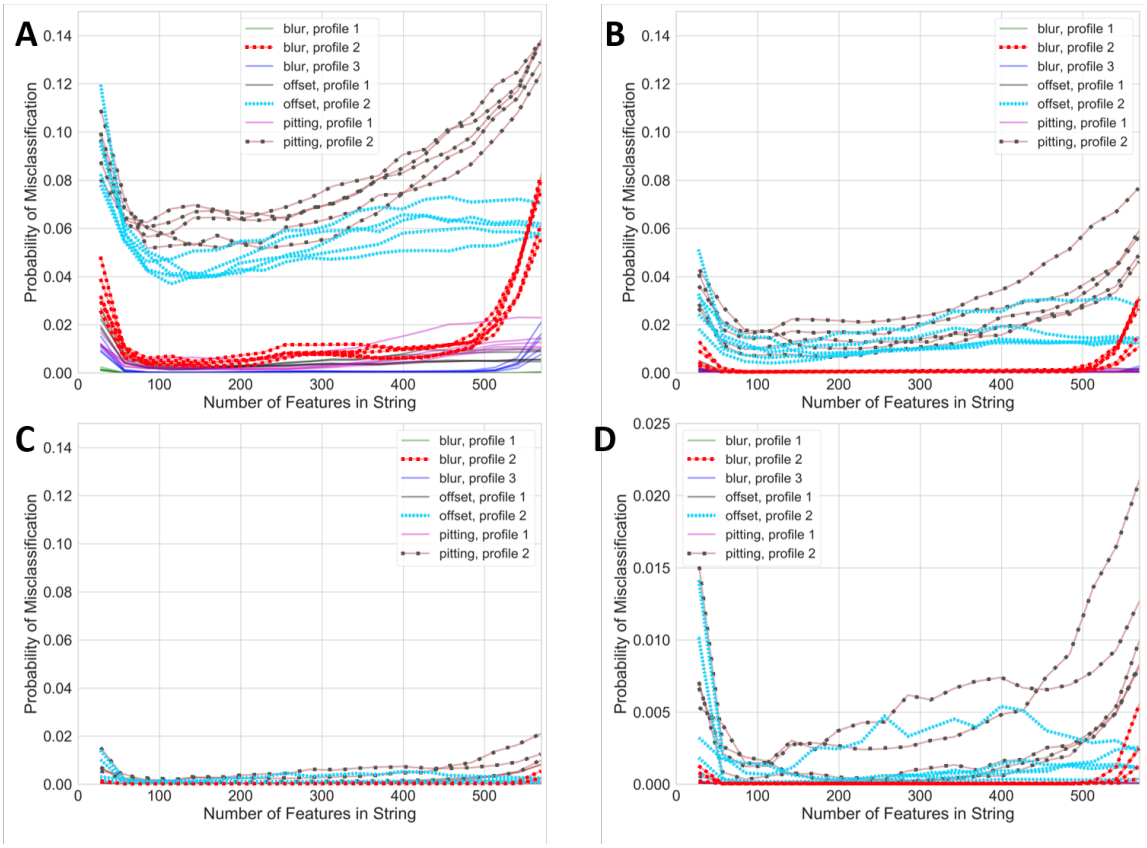

Fig. 5.10. Probabilities of mis-classification for five folds of the micrograph data set when (A) features are taken from one location in the micrograph, (B) two locations, and (C – D) four locations. The y-axis is scaled differently in (D) for clarity.

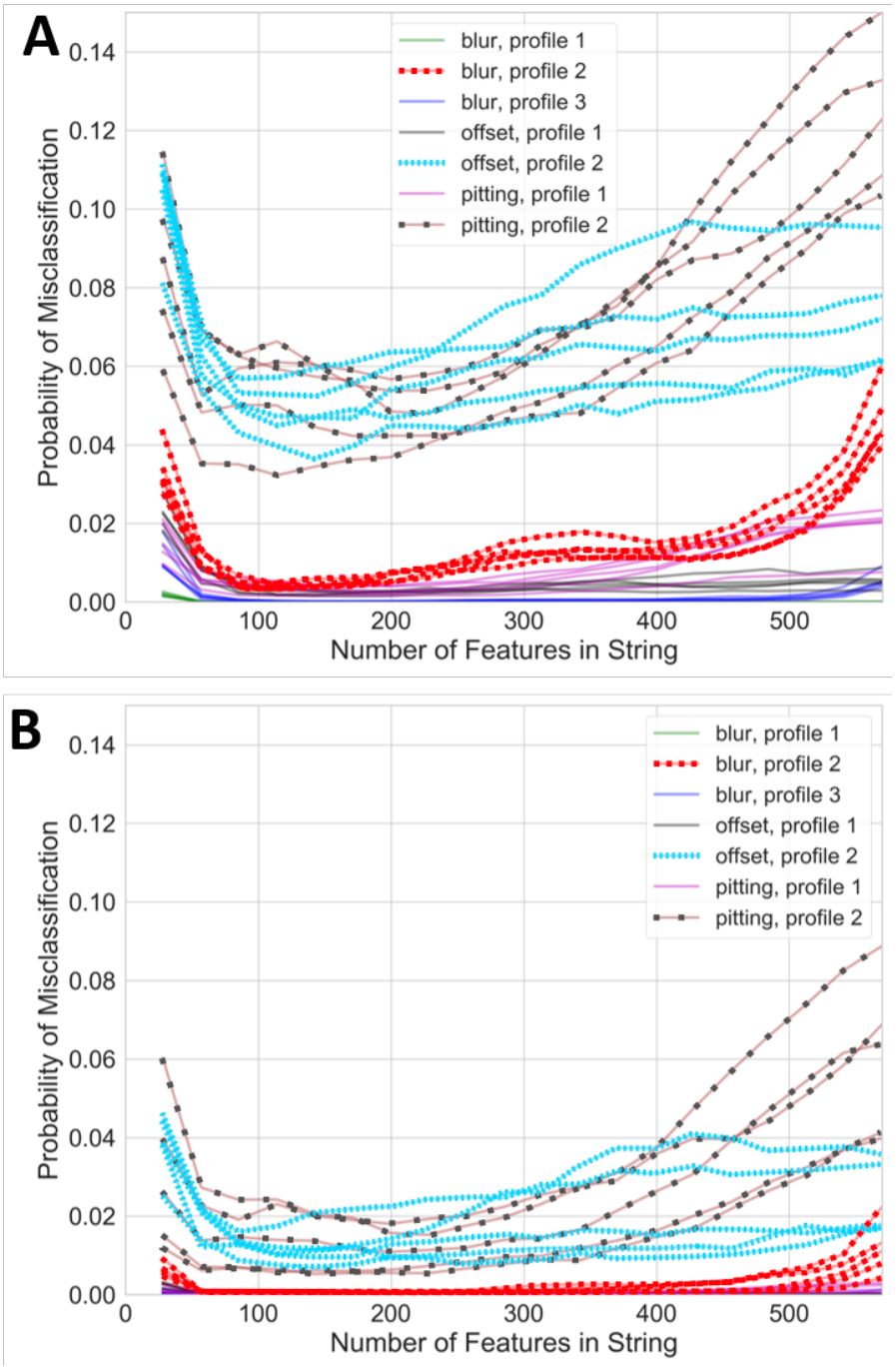

Fig. 5.11. Straight (horizontal) path performance. Probabilities of mis-classification for five folds of the micrograph data set when features are taken from (A) one location in the micrograph with horizontal path, and (B) two locations with horizontal path.

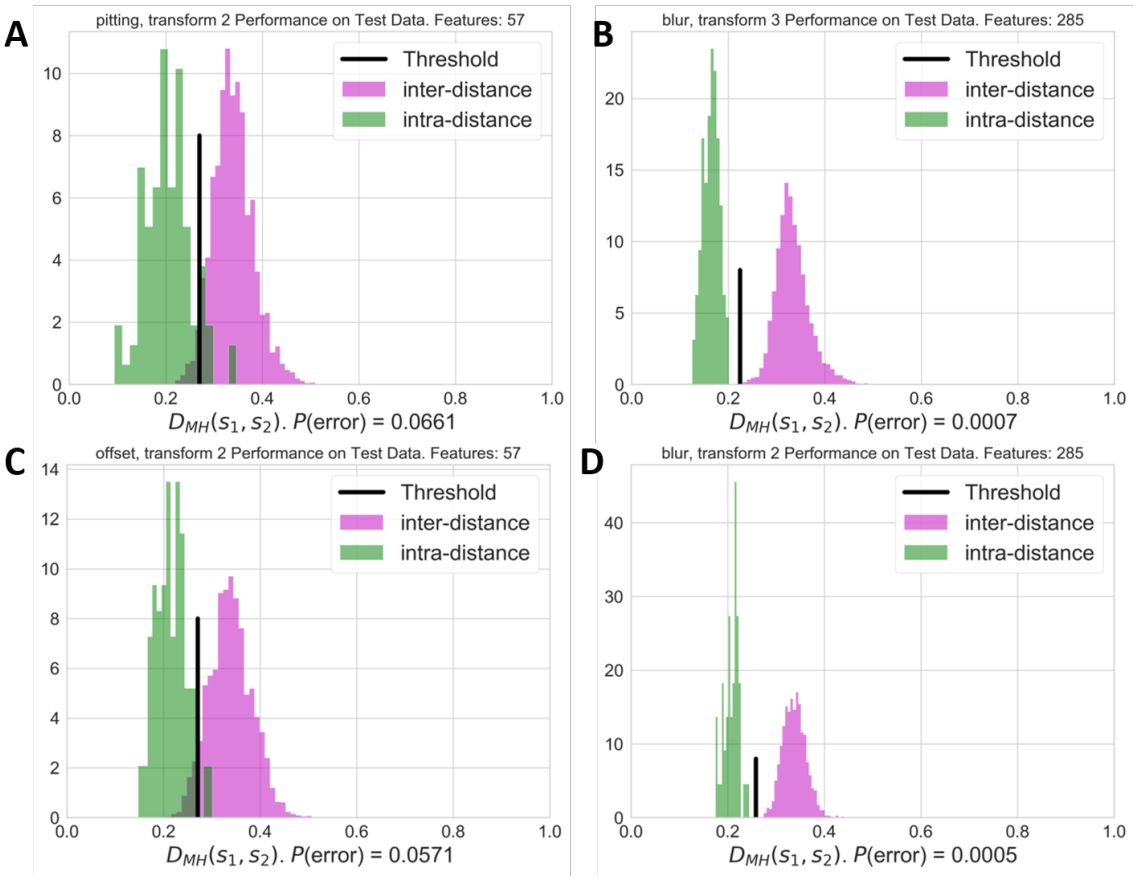

Fig. 5.12. Inter- and intra-distance histograms indicating poor (A and C) and good (B and D) classifier performance. That is, "peakyer" and further-separated inter- and intra-distance histograms are preferred. (A) pitting profile 2, considering 57 features from 1 micrograph location, (B) blur profile 3, 285 features, 1 location, (C) offset profile 2, 57 features, 1 location, (D) blur profile 2, 285 features, 2 locations. The black vertical bar indicates the equal-probability threshold τ for the classifier.

Estimated probabilities of mis-classification for strings constructed from x input features, with order given by the feature value scores for that damage profile, are plotted in Figures 5.10 (A) (where strings are taken from only one "path" location in the micrograph) and 5.10 (B-D) (where strings are taken from two or four paths in the micrograph and concatenated to improve performance). As discussed in Chapter 4, D_{MH} was computed for every (s_1, s_2) combination in the test fold of the data, with s_1 coming from the original data for each micrograph and s_2 from the damaged micrographs. The estimated mis-classification probability for each damage profile is calculated as

$$\begin{aligned} P(\text{error}) = P(\text{CF}) \int_0^{\tau} p(D_{MH}|\text{CF}) dD_{MH} \\ + P(\text{Genuine}) \int_{\tau}^{1} p(D_{MH}|\text{Genuine}) dD_{MH}, \end{aligned} \tag{5.10}$$

where $P(\text{CF}) = P(\text{Genuine}) = 0.5$ are the a priori probabilities that the part is a counterfeit or genuine respectively, τ is the distance value D_{MH} such that $p(\tau|\text{CF}) = p(\tau|\text{Genuine})$ for the training data, and $p(D_{MH}|\text{CF})$ and $p(D_{MH}|\text{Genuine})$ are the normally-distributed probability density functions fitted to the damage profile's inter- and intra-distance D_{MH} histograms, respectively.

It can be seen that performance does not improve after around 100 included features, provided those features are ordered according to expected performance for the given damage profile. For the more severe pitting profile 2, performance drops off drastically once more than 200 to 300 features have been included in the string. In practice, one wants to select the number of features to consider that minimizes mis-classification probability, given the most likely damage profiles. Damage profiles corresponding to more severe translation and pitting are consistently poor performers over varying string length. This may be due to the fact that translation and more severe pitting alters all pixel responses in the input tile window and the distribution of phases across the tile path, respectively, which affects the autocorrelation and PC score responses much more drastically than the more localized blurring damage. Similar trends may be seen in Figures 5.11(A) and (B), which correspond to the

same experiment run using a straight (horizontal) path across the input micrographs, rather than a circular path. This implies that these different tile paths, for this study, do not drastically change performance and may instead be chosen by the user based on other factors, such as ease of implementation.

For more severe damage types with high mis-classification probability over all string lengths, reliability may be increased by performing feature extraction and string construction at two or more locations in the part's micrograph. This increases string storage requirements given a desired number of features to extract per location, but results in better identifier performance as seen in Figure 5.10 (B-D) and 5.11 (B). Results for strings considering one location are given in Figure 5.10 (A), while results for strings considering two and four locations in each micrograph are given in Figure 5.10 (B) and (C-D) respectively. Results for strings considering two or more locations were generated by creating groups of original instances in the original dataset, and taking each group as one instance in the new dataset.

Note that the less-severe damage profiles achieve good performance (less than 0.01 probability of misclassification) with only one string location, but more severe profiles require at least two locations to be considered for practically acceptable performance. In practice, this may be done by enforcing two or more responses per micrograph challenge input. For visualization, and to give a qualitative sense of what these misclassification probabilities encode about the feature responses, several inter-intra-distance plots are given in Figure 5.12, with the corresponding mis-classification probabilities. Note the significant inter-distance-intra-distance histogram overlap for more severe damage profiles considering only one location for feature extraction; this highlights the need to evaluate this probability before committing to an anti-counterfeiting scheme design.

An engineer using this method may proceed to design an anti-counterfeiting scheme from these results as follows:

1. The engineer provides training data for damage profiles for the parts under consideration.

2. Knowing which data profiles are most likely to occur, perhaps through experience or experiment, the engineer identifies the most "valuable" features for those profiles.

3. Based on these results, the engineer estimates the per-part-location performance given a set of ranked features using a figure similar to Figure 5.10, generated for the given use case.

4. The engineer selects the number and identity of features to extract, and the number of locations from which to extract features, that minimizes the expected probability of mis-classification subject to string length constraints.

5.4 Comparison to Performance for Experimental Data

To get a better sense of how this approach generalizes for different systems of micrographs, the same methodology has been applied to the brass and steel data previously analyzed in Chapters 3 and 4. Results using this experimental data from Chapter 3 (2-phase images from processed brass micrographs) and 4 (3-phase images from processed steel micrographs) were generated for comparison. The 50 raw images from each library were segmented into 850x850-pixel tiles, resulting in a total of 285 micrographs available for analysis. Relevant data counts and IPCA training information is listed in Table 5.3.

Preprocessing: Steel data was pre-processed as described in Chapter 4. Brass data was pre-processed by first applying a Gaussian blurring filter of size 7 pixels and sigma 2, and then binarizing the blurred images using adaptive Gaussian thresholding with a neighborhood of 201 pixels. These pre-processing steps were implemented using the python OpenCV library [100].

PCA Training: 45 images from each of the experimental datasets, chosen at random, were held out to estimate the PCA transformation, using the same procedure discussed for the synthetic dataset. These images were then removed from the remaining analysis. From each of these images, 16 200x200-pixel tiles were extracted

Table 5.3.
Parameters used for analysis of experimental data analyzed in Chapters 3 and 4. Units in pixels for consistency.

Parameter	IPCA Data	Input Data	Units
count	45 (720 total tiles)	240	micrographs
IPCA batch size	40	N/A	micrographs
micrograph dimensions	850x850	850x850	pixels
tiling window dimensions	200x200	200x200	pixels
number of phases	2 (brass), 3 (steel)	2 (brass), 3 (steel)	N/A

for IPCA training, resulting in 720 tile images for training, with a batch size of 40. This IPCA training was carried out otherwise identically to the synthetic data analysis presented earlier in this chapter. These transforms were then used to generate the PCA traces for each dataset as discussed above.

Analysis: For each experimental dataset, 4 folds of 60 images each were generated from the images not used for PCA training, and were used to generate results for each of the damage profiles discussed for the synthetic data. Note that for the brass data, which only has 2 phases (labeled 0 and 1), pitting damage was assumed to mark pixels as belonging to phase "0." Probability of misclassification results are presented in Figure 5.13, assuming an ordering of features as determined by the method discussed for the synthetic data. Violin plots for the relative rankings of features according to this ordering are given in Figure 5.14 for different damage profiles, and Pearson correlation matrices for the resulting 100 best features are presented in Figures 5.15 and 5.16.

Misclassification Probability. Similar to the results for the synthetic dataset, it can be observed in Figure 5.13 that the poorest performance occurs with the most severe offset and pitting damage profiles. This is consistent across both experimental datasets, although overall performance is significantly worse for the 2-phase brass

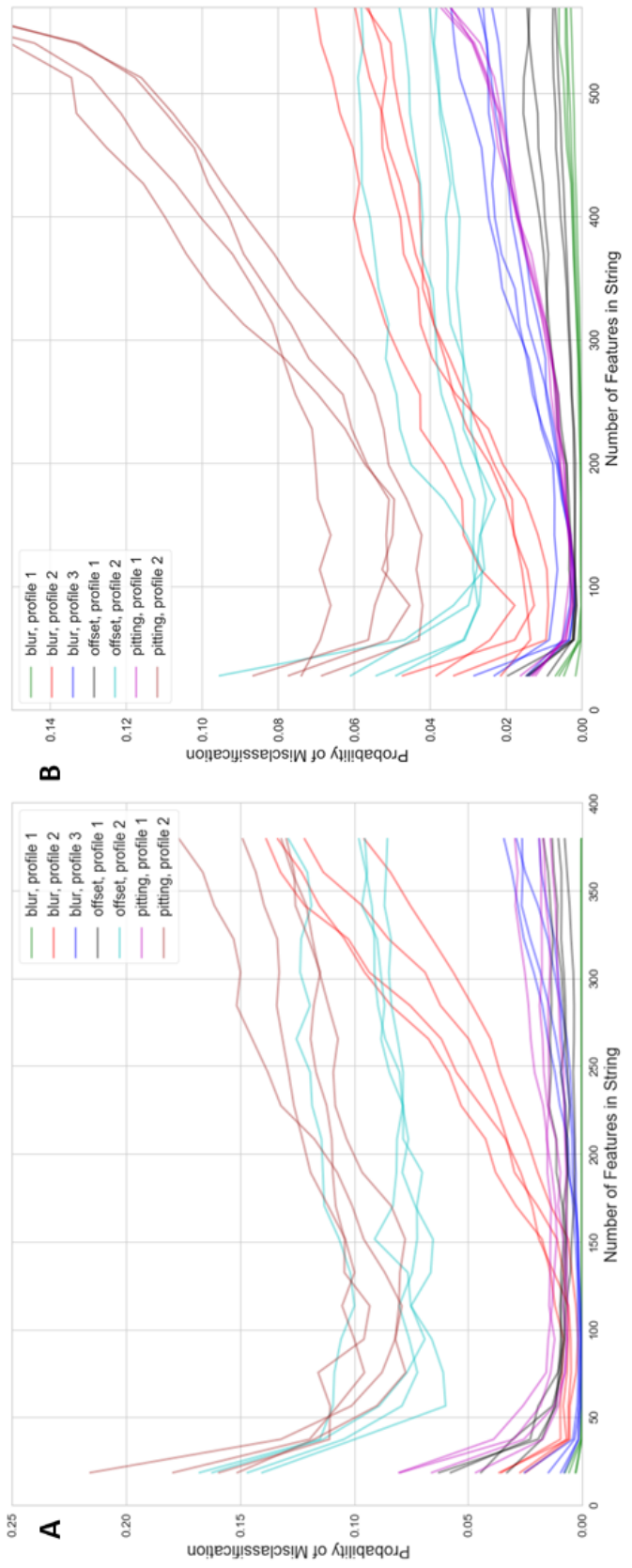

Fig. 5.13. Probabilities of mis-classification for four folds of the micrograph data set when (A) features are taken from one location in the micrograph for brass data, (B) features are taken from one location in the micrograph for steel data. Note that for clarity, the y-axes of panels (A) and (B) do not align.

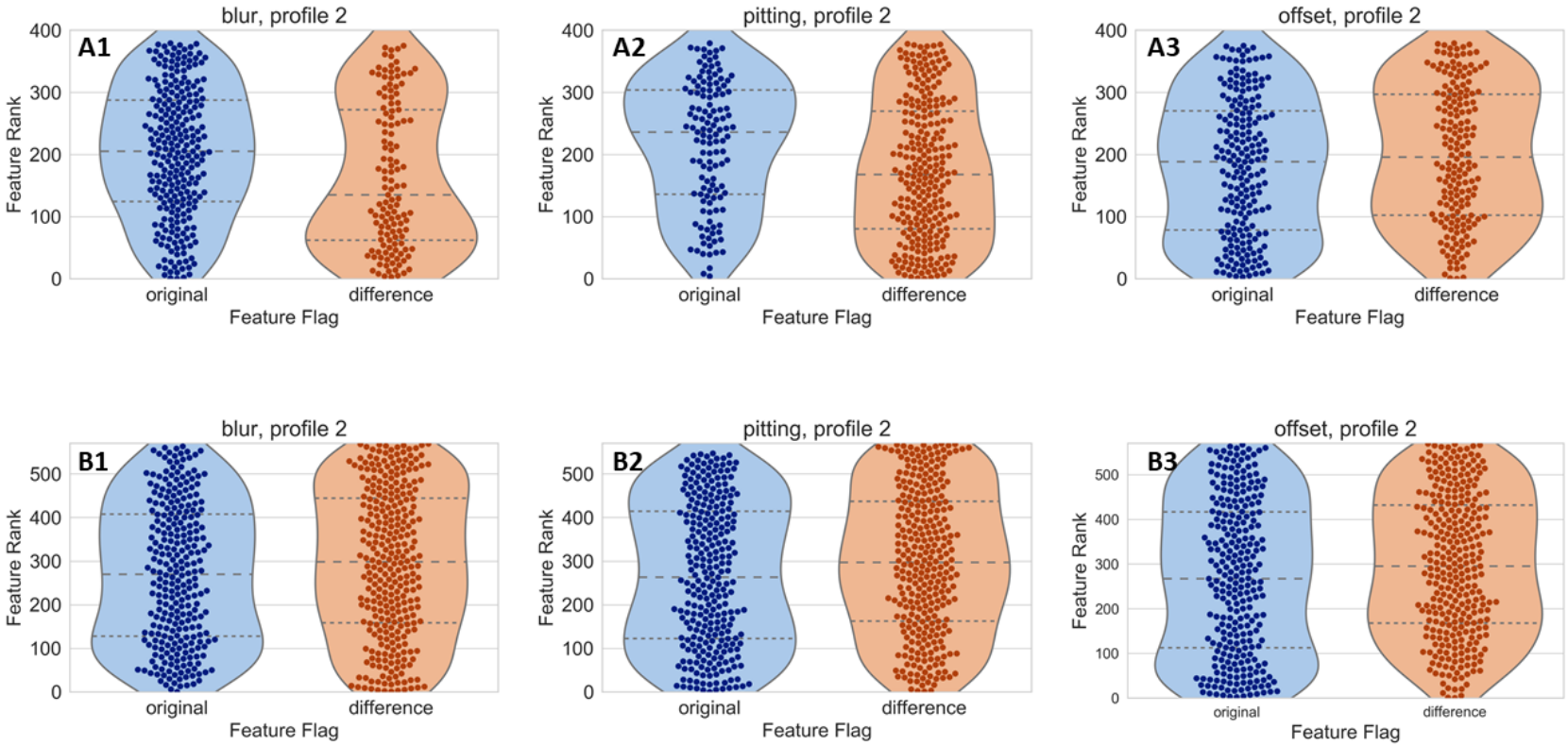

Fig. 5.14. Violin plots displaying ranks of features after selecting between features derived from original and differenced series. Lower rankings are preferred. (A1-A3) brass feature data, (B1-B3) steel feature data.

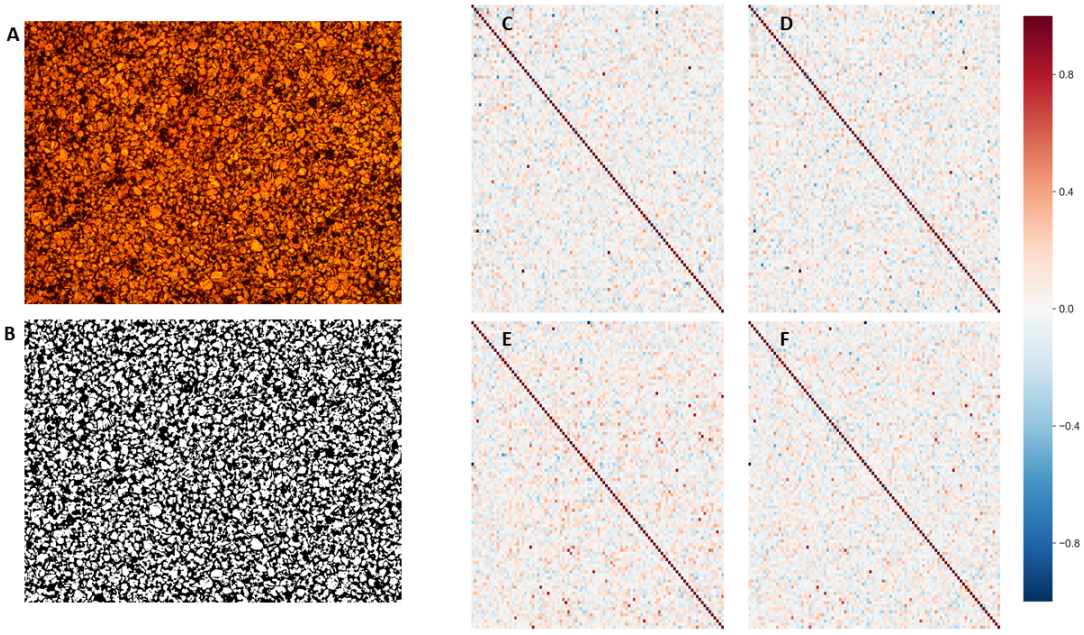

Fig. 5.15. Example brass input micrograph (A) and thresholding result (B). For the brass data, Pearson correlation matrices of feature responses of the 100 highest-value features for the enrolled data under blur profile 2 (C) and offset profile 2 (E), and correlation matrices of the differences in feature responses before and after damage for blur profile 2 (D) and offset profile 2 (F).

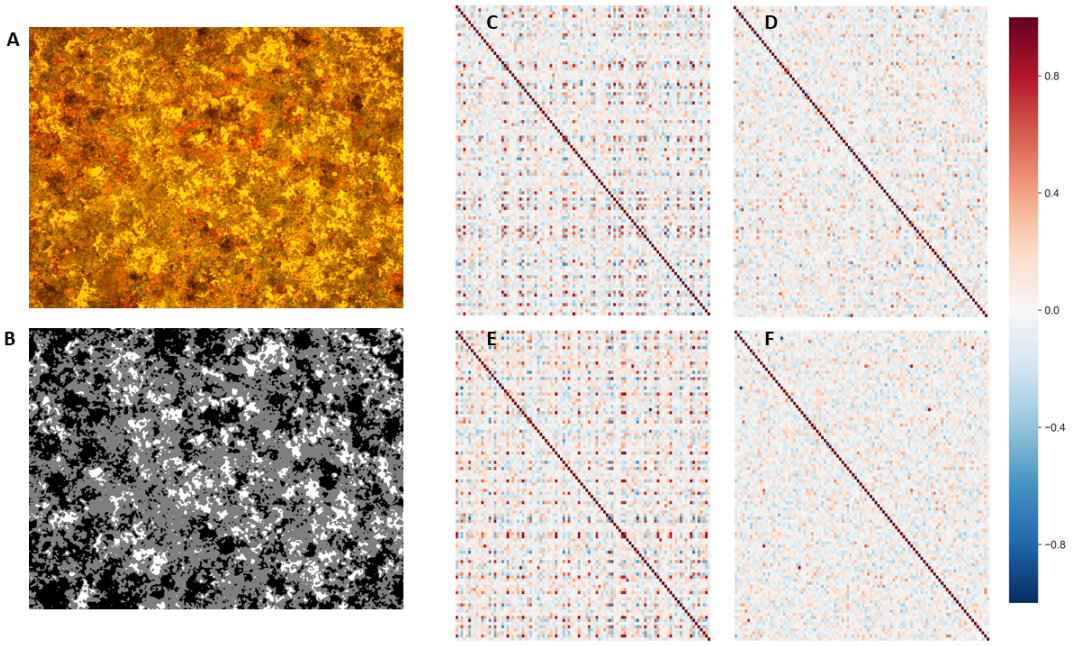

Fig. 5.16. Example steel input micrograph (A) and thresholding result (B). For the steel data, Pearson correlation matrices of feature responses of the 100 highest-value features for the enrolled data under blur profile 2 (C) and offset profile 2 (E), and correlation matrices of the differences in feature responses before and after damage for blur profile 2 (D) and offset profile 2 (F).

data compared to the 3-phase steel data, even when considering an equal number of features. This may suggest lower information content in the 2-phase data.

Feature Preference. Interestingly, it can be seen from the violin plots in Figure 5.14 that preference for original-series features or difference-series features are not consistent across damage profiles for the same material system, nor are they consistent for the same damage profiles for different material system. This underscores the need to tailor feature selection both to the material system *and* the damage profile(s) under consideration. Future work should focus on characterizing feature libraries across both of these dimensions for a variety of manufacturing processes.

Feature Correlation. Observing Figures 5.15 and 5.16, relatively low correlation between top features are observed for both material systems. This is especially true for the brass data. For the steel data, it can be seen that features are occasionally selected that have high correlation with several higher-ranked features, but such features are added sparingly and not in sequence. For both material systems, the differences between feature responses also display low correlation, as desired.

5.5 Conclusion

To summarize, the last three chapters have attempted to do the following:

- establish the feasibility of microstructural descriptors and features extracted from micrograph data in anti-counterfeiting schemes,
- formulate of encoding schemes for translating feature responses into bitstrings that serve as a part identifier and provide an intuitive way to compute a distance metric to the expected part response,
- propose a method for evaluating the value of a feature for a given problem characterized by its material system, available sensors and computing equipment, and expected damage profiles, and
- evaluate that method using synthetic and experimental micrograph data.

The last three chapters represent my investigation of RQ1. In investigation of RQ2, the next two chapters will deal instead with *embedding* information during manufacturing, rather than simply reading information for traceability purposes. This discussion will be guided by the understanding that there is a wealth of inherent information at our disposal, whether from micrograph descriptors, surface characterization, or other sources of randomness gifted to us by physics. Indeed, as I shall discuss combining this intrinsic information with methods for embedding extrinsic messages will provide interesting extensions to existing traceability schemes.

6. INFORMATION EMBEDDING IN ADDITIVE MANUFACTURING

The previous three chapters discuss investigations into RQ1, leveraging *intrinsic* information available in manufactured goods for traceability. This is done by finding features of these goods that are well-suited for PUF-like formulations by answering questions like: Do the features display high variance (uniqueness, unpredictability)? Are they hard to control (one-way)? Are they relatively easy to read (evaluatable)? Are they robust to whatever damage or evolution you anticipate (reproducible)? Methods that use such features may indeed be suitable for designing traceability schemes, but as discussed previously, the "message" encoded using these schemes is essentially "I'm me!" This is great in an anti-counterfeiting scenario, but how could these features instead be used to communicate additional information? How can we use the manufactured good itself as a communication channel throughout the product life cycle? This chapter and Chapter 7 focus on this question specifically. In this chapter, RQ2.1 and 2.2 receive attention, and preliminary schemes for *dynamically encoding information during manufacturing* are discussed, from the perspective of additive manufacturing.

This chapter proceeds as follows. First, a brief review of the literature in information embedding and security in additive manufacturing is presented, along with modest proposals for extending this work. Then, the motivation and format of information embedding schemes in additive manufacturing is discussed abstractly, and formalized in the concept of the *malleable PUF*. This sets the stage for the following discussion of possible schemes for additive manufacturing that make use of user-specified secrets (keys), and optionally intrinsic information of manufactured goods that is observable at manufacturing time. Preliminary simulation results are then

presented and discussed. Chapter 7 then presents a discussion on how these schemes may be implemented in practice.

6.1 Existing Literature

This chapter and Chapter 7 are built on the following premises: 1) *additively manufactured parts display a high degree of heterogeneity at the microstructural level, and characterizing this heterogeneity is a significant research focus in the additive manufacturing (AM) community,* and 2) *heterogeneity in AM parts, introduced either intrinsically by uncontrollable processes or extrinsically through local perturbations in manufacturing parameters, can be leveraged as information carriers for traceability and security schemes.*

Additive manufacturing's attractive qualities (customized parts, fine control over geometry and material properties, minimal tooling requirements relative to subtractive manufacturing processes, and so on) have inspired significant recent work in characterizing the relationships between process parameters and part properties. For instance, NIST's Additive Manufacturing Metrology Testbed (AMMT) [101] has been designed in part to characterize part quality through in-situ measurements of important process parameters. This characterization is important for ensuring part quality, say by dynamically controlling laser power in laser powder bed fusion (LPBF) to minimize microstructural defects [102], or attempting to map real-time, in-situ melt pool monitoring signals to ex-situ measured melt pool geometry features [103]. Still, more work is needed to better understand the relationships between stock material (like metal powders), AM machine characteristics, build parameters, and final part properties [104]. As discussed by Sutton and coauthors [105], the complex relationship between powder morphology, chemistry, and microstructure, and the laser fusion parameters contribute to existing unintended variation in AM metal parts.

6.1.1 Security Concerns

On top of these manufacturing-related concerns, AM also exhibits unique security-related issues. As might be expected, the AM process generally requires a high volume of data exchanges (from design, which may be collaborative, to geometry specification in STL or other format, to tool path generation and execution), with this data constituting the "digital thread" of the part [106, 107]. These exchanges, if not carefully tracked, can introduce opportunities for adversarial attacks that may negatively impact the quality of the final part. Padmanabhan and coauthors [108] detail potential attacks that may occur between the design/ideation stage of AM manufacuturing and final quality checking of the finished part, highlighting the large potential attack surface. Yampolskiy and coauthors [72] provide a detailed overview of existing security concerns in additive manufacturing, and provide detailed taxonomies for important potential attacks including theft of technical data and process sabotage. Importantly, the authors highlight literature detailing potential attack vectors and mitigation strategies at multiple stages of product design, manufacture, and distribution. Even if no attacks occur during manufacturing, users will still need reliable, low-cost access to manufacturing information that will be relevant downstream, or at least a method to determine which attack detection and mitigation strategies were used during manufacturing.

Of course, just as in other systems with security components, the cybersecurity-system performance trade-off is alive and well [109]. NIST's testbed for industrial control systems (ICS) is explicitly designed to study how cybersecurity instrumentation impacts performance in industrial control systems like those that control additive manufacturing processes [110, 111]. However, AM presents a unique opportunity to "play both sides" of this tradeoff, leveraging the inherent heterogeneity and unprecedented level of control over part geometry and composition to develop inherent traceability schemes with little to no impact on the cost and performance of the manufacturing process itself.

6.1.2 Information Embedding

Embedding information using single- and multi-material printing has seen recent attention in literature, with information carried using a range of sources, from optical or geometric [112, 113], to temperature [114] and chemical signatures [114, 115]. Embedded information can be used to create unique signatures or carry additional information through, say, QR codes. Wei and coauthors [114] propose a method for embedding information in SLM through embedding a "tagging" material dissimilar to the bulk stainless steel print stock. The tag's signal is then carried by the differing chemical composition of the tag and stock materials, and is read using X-ray fluorescence or thermal imaging. Maia and coauthors [112] propose a method that modifies the color or geometry of layers of a 3D-printed part to generate a bar code that can be de-coded from an image of the part. Kennedy and coauthors [115] link tagging technologies to traceability through other recent technologies: distributed ledgers (blockchain) and digital twinning. QR codes are embedded in a 3d-printed part by blending PLA material with UV-fluorescent nanoparticles. These codes are then linked to an external blockchain ledger containing the part's digital twin information (for reference, see Tao and coauthors [116]), and potentially a ledger of ownership transfers.

Delmotte and coauthors [113] provide both a good overview of existing watermarking and anti-counterfeiting schemes in polymer additive manufacturing, as well as propose a blind watermarking scheme that encodes information by varying printing layer thickness in designated "patches" of the part. These patches can then be read with inexpensive scanners, and the data may then be processed to recover the watermark. Importantly, the authors note challenges in error correction for codes printed on additively manufactured parts. In the fused deposition modeling (FDM) process considered by the authors, "random" errors, where bits are more or less equally likely to be incorrect, occur more frequently than the "burst" errors for which common codes, like Reed-Solomon codes, are appropriate. Any applied error-correcting codes

may need to be modified or completely re-written to account for this. Further, the authors' notion of "blind watermarking" is to check parity bits in each line of the embedded watermark matrix, and use those checks to determine if the watermark is correct. Our goal, rather, is to encode messages containing more than just identity. The authors' method could be extended to encoding a message, but that message would repeat without variation over the surface, and an attacker would have access to the watermark and watermarks on other parts to help them decode it. The methods proposed in this chapter would produce unique signals at every patch, even using the same key and plaintext message from the manufacturer.

Traceability of parts through a combination of keyed information and digital representations of the part has also been studied. Aliaga and coauthors [117] propose a keyed genuinity testing scheme for 3D objects that makes use of strategically added manufacturing noise to generate a footprint that is relatively easy to query but not reproducible by an adversary with knowledge of the part. Testing requires access to the original CAD model of the part, however, which may be undesirable for IP reasons, and is limited to encoding identifying information as in Chapters 3 to 5.

6.1.3 Fuzzy Extractors, PUFs, and a Proposal for Malleable PUFs

Finally, it is worth revisiting the concepts of PUFs and fuzzy extractors, as the embedding schemes discussed in this Chapter build on this work. As a reminder[1], *fuzzy extractors*, as defined by Dodis and coauthors [41], are cryptographic primitives that extract nearly-uniform random information from non-uniform biometric data. Here, "non-uniform" implies the information source does not provide the (uniformly) random information assumed in most cryptographic applications, and this certainly describes the information sources already discussed in this dissertation[2]. "Biometric"

[1]See Chapter 2 for additional discussion of fuzzy extractors and PUFs.

[2]Although, the desire for random-like data motivates in part the mRMR feature selection and percentile-based feature binning methodology presented in Chapter 5. This desire should motivate similar decisions for other schemes generated using the intrinsic traceability scheme design framework presented in that Chapter.

generally refers to any data gathered from some human source, such as a fingerprint or iris scan, but in the context of fuzzy extractors can be abstracted to any source w that, when queried with an appropriate randomness extractor, produces some noisy and near-uniformly random output R. Fuzzy extractors allow for recovery of a constant output R as long as the queried response R' is sufficiently close to R, where "sufficiently close" is quantified by some distance measure like the Hamming distance or (if one must consider bit additions and deletions) the edit distance.

Thus, fuzzy extractors are well-suited to problems where source *intra-distance*, or differences between responses of the same source instance at different times, is the main concern. However, the PUF-like schemes discussed in this chapter (and in the earlier chapters) must be designed in a way that considers both intra-instance and *inter-distance*, or differences between separate source instances, given the manufacturing context at hand. Further, fuzzy extractors, and for that matter PUFs more generally, do not consider writing information beyond the yes/no identification signal. For additive manufacturing information embedding, integration of PUF-like methods can open the way for physically unclonable message embedding beyond simple identification.

The Malleable PUF. I will refer to schemes discussed in this chapter and Chapter 7 as "malleable PUFs." These PUFs combine notions of designing for robustness (as with fuzzy extractors) and for secure identification (as with fuzzy extractors and PUFs), while also requiring an ability to design for information carrying capacity. For the purposes of this dissertation, a malleable PUF is defined as a *process for modifying a product during manufacturing and a corresponding procedure for querying that product* that satisfies the following requirements:

1. The query procedure has the required properties of PUFs [43], that is, it should be evaluatable, unique, reproducible, unclonable, unpredictable, one-way, and tamper-evident.[3]

[3]See Section 3.4 for a more detailed discussion on the properties of PUFs, as presented by Maes and coauthors [43].

2. The modification process allows for the writing of some manufacturer-specified information that is readable through the query procedure. Ideally, this reading should require little to no data saved externally from the product.

It should be noted that this concept is a straightforward extension of the goals of the methods presented in Chapters 3 through 5. Now, I just impose the need to be able to write information to the part during manufacture. However, for the unclonability requirement to hold, it will be necessary for the query procedure to make use of some *intrinsic* information of the part that is unmodifiable by the manufacturer, like the microstructural information discussed in Chapters 3 through 5. Then, for one-wayness and unpredictability to hold, the query procedure must also have some mechanism for obfuscating the extrinsic written message using this intrinsic signal. This introduces interesting technical challenges that are not considered in the read-only cases presented earlier in this dissertation.

6.1.4 Research Opportunities in PUF-like Information Embedding

So, how do we incorporate intrinsic information, as done in PUF and fuzzy extractor literature, to create malleable PUFs that encode extrinsic information during the additive manufacturing process? The literature in this area leaves room for several expansions in traceability:

- Although much work is underway in characterizing AM processes, there is a lack of manufacturing feedback-based encoding schemes which would introduce randomness intrinsic to the part into the information being embedded. "Feedback" here refers to using information about the part as it is being manufactured (solidification, phase properties, surface characteristics like roughness, etc.) and using this information in combination with the data to be encoded to produce an intrinsically "signed" signal.

- Feedback-based encoding has its own unique set of issues. From an engineering and materials perspective, any manipulation done on the part to encode some

signal must at the same time preserve the mechanical properties of the part. From an encoding perspective, this implies a potentially complex model with severe constraints on transitioning between successive locations on the encoding surface, impacting a resulting method's channel capacity.

- Many of the solutions discussed in the literature suffer from the fact that they can be easily interrogated by an adversary, and they may be easily re-created using standard additive manufacturing techniques. Thus they may be decoded and/or reproduced by a determined counterfeiter. It remains to propose methods to embed information that (i) are reliably difficult for an attacker to decode, even when given many examples, and (ii) in the vein of PUF literature, are reliably difficult to reproduce even when given the part's "response" message.

6.2 Why Additive Manufacturing?

Manufacturers often wish to transmit some information with their goods, for a variety of reasons discussed in Chapters 1 and 2: batch IDs and specifications may be needed for quality control and tracking, meta-data may be needed to track parts across multiple stages of a project, design or maintenance data may be needed for safely using and maintaining goods, and so on. This information may also be used for anti-counterfeiting purposes discussed previously in this dissertation: a manufacturer may simply want to encode a part's ID and its source for traceability.

This chapter presents schemes that leverage *intrinsic* information of the part being manufactured to embed these *extrinsic* signals. By introducing an embedding mechanism that uses intrinsic information of the part measured earlier in the manufacturing process, a location dependency is introduced that is self-evident (so it does not have to be stored and recalled) and, if distorted at one location on the part, may be safely ignored if a readable message is recovered elsewhere. This is of particular interest in additive manufacturing, where local features of the part can be modified *during manufacture*, and *where the properties of the part at each location may already*

be monitored for quality assurance purposes. Other aspects of additive manufacturing are particularly useful for information embedding:

- Many AM processes build parts layer-by-layer, which implies a natural ordering to physical locations on the part that could be used to order symbols of an encoded message.
- Existing literature in characterizing local surface features of parts for different process parameters implies the existence of information channels for embedding information, if these processes can be dynamically controlled. Progress in monitoring changes to the part over time and studying the effect on the encoded message, and in building high-dimensional encoding alphabets using sufficiently complex features, may lead to higher capacity, more robust schemes as the field develops.
- Outside of AM (if a manufacturer wishes to extend encoding methods to "subtractively" manufactured goods), many surface finishing processes (milling, peening, etc.) proceed in an ordered and controllable way over the part surface.

The remainder of this chapter is organized as follows. First, a conceptual framework for designing and implementing these embedding schemes is introduced. Then, a general keyed embedding scheme that leverages assumed intrinsic information that can be reduced to a binary alphabet is presented. This scheme is then expanded to account for localized damage that would manifest as burst errors, followed by a discussion on error correction approaches necessary for this scheme. Finally, simulated results are discussed as a motivation for future experimental studies.

6.3 The Information Embedding Process and Terminology

A *feature* in this context is a random variable characterizing some response that can be *read* from the manufactured good and, importantly, can be *controlled* during

manufacturing with some degree of confidence. These features can then be mapped to an *alphabet* of characters to be encoded in the part during manufacturing.

Uncertainty inherent in feature responses and the feature reading process at challenge time, combined with potential damage or other changes to the part, imply that the message encoded is transmitted through a *noisy channel* whose behavior is characterized by the encoding, embedding, and evolution of the part. This necessitates careful consideration of *error correction methods* in the face of *limits on the rate of the channel* and *the behavior of the noise*, which may be random-like in the case of unreliable encoding during manufacturing, burst-like in the case of localized damage to the part, or some combination.

So, we have that the information embedding process entails a *design stage* where noisy features are mapped to an alphabet of encoded symbols, which are then used to transmit a message over a channel with (possibly highly complex) noise. This is followed by implementation, where the control necessary to implement the encoding scheme is provided to the manufacturing apparatus. An overview of this process is given in Figure 6.1. Note that gray boxes in this figure represent important design choices or constraints that must be characterized for each use case.

6.3.1 Watermarking and Steganography

For this discussion, embedding schemes can follow one of two philosophies: that of *watermarking* and that of *steganography* [118]. The distinction is important to establish, so that there is no confusion regarding the goals of the schemes discussed in this chapter. Consider, in the vein of cryptographic literature, the following "embedded message game" played by secret senders, secret (legitimate) readers, and (illicit) eavesdroppers. The sender has, through some method, embedded the the message "The Lannisters Send Their Regards" in one of two otherwise indistinguishable artefacts, both of which are then sent to the reader. In transit, these artefacts are intercepted

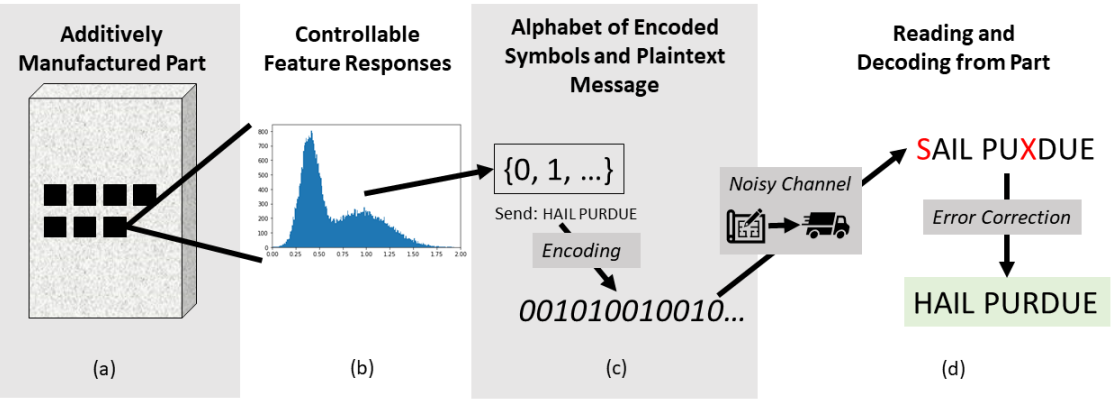

Fig. 6.1. Information encoding with intrinsic, controllable features. (a) Locations for encoding symbols are defined on the part, and ordered according to the AM process (eg., in raster order). (b) Features at each location are characterized for robustness, information content, and controllability. (c) High-value features are used to define an alphabet of symbols to be encoded at each location. This alphabet is used to encode the possibly encrypted message on the part. (d) Given the encoding scheme and error rates due to encoding and transportation damage, decode the message. The "noisy channel" in this framework includes the encoding process during manufacture, potential damage/part changes over time or during transport, and noise in reading the message from the part at challenge time.

by the adversary eavesdropper, and the adversary is not told which has the embedded information and which does not.

In Game 1, the eavesdropper wins if they can correctly *determine which artefact has an embedded message*. In Game 2, the eavesdropper wins if they can *determine which artefact has an embedded message, and reproduce the message in another artefact they create.* If the probability of the eavesdropper winning Game 1 is not 1/2, that is, if they have some better-than-random chance of determining which artefact carries the message, then the embedding scheme is not steganographic as the mere *presence* of the message is not secret. If the eavesdropper has a better-than-random chance at winning Game 2 and reproducing the message in their counterfeit artefact, then the embedding scheme fails as a watermark for the purposes of this discussion.

While steganography is a fascinating area in computer science, it is perhaps too strong of a requirement for the schemes investigated in this dissertation. Generally, in steganography literature it is assumed i) that the sender can manipulate any aspect of the object being used for transmission (such as an image, video, or text document), and ii) that the object is unchanged over time (as these files, saved on disk, usually are). Indeed, formally demonstrating that any scheme is steganographic is a tall order, especially in relatively new areas like additive manufacturing. Here and for the remainder of this dissertation, schemes will not be held to this standard. However, I mention it here to call out steganographic embedding schemes as an important topic for future work.

6.4 Potential Embedding Schemes

This section presents a scheme for embedding secret messages in two-dimensional arrays that can, depending on the implementation, meet the Game 2 *watermarking* requirements for embedding information. It is deliberately designed to be general enough to extend to cases with highly different available alphabets. The scheme assumes an ordering to the entries in the array, and uses information of the previous

L entries and a key to embed messages as the array is written one entry at a time. In particular, the method is appropriate for embedding messages in the surfaces of physical components that are built additively, such as 3D-printed parts. Given at least one controllable feature whose (assumed binary) value can be embedded locally during manufacturing with a high degree of confidence, and a feedback mechanism capable of reading previous feature values, the proposed scheme is able to embed messages in a keyed fashion that is (as of now, qualitatively) indistinguishable from noise (proof will be provided once the scheme is specified below). Note that in this chapter, it is assumed that the "alphabet" of characters that can be directly encoded is binary, that is $\{0, 1\}$.

Consider a process by which a surface is built sequentially, in that every pixel p_n (or voxel) on the surface is created at time n, and the set of pixels on the surface can be ordered by their time indices. Further, assume that at time n, the process has access to information on the values of all previous pixels $p_{n-1}, p_{n-2}, ..., p_0$. In the proposed scheme, we make use of this previous information, as well as a user-provided key K, to robustly encode a secret message in the surface as it is built.

For now, I will restrict the discussion to binary pixel values, $p_n \in [0, 1]$, and make the assumption that the pixel values are equally likely to occur without dynamic input. This assumption can be relaxed, resulting in a more complex model of performance. Further, I assume that we have access to an encoding process ENCODE(b) we can use to dynamically write a bit value b to the current pixel location, possibly with a chance of failure. In practice, ENCODE(b) will most likely involve manipulating some process parameters at time n to create the desired characteristics at the location. Given ENCODE(b) and the ability to read previous pixel values, we can construct the proposed scheme.

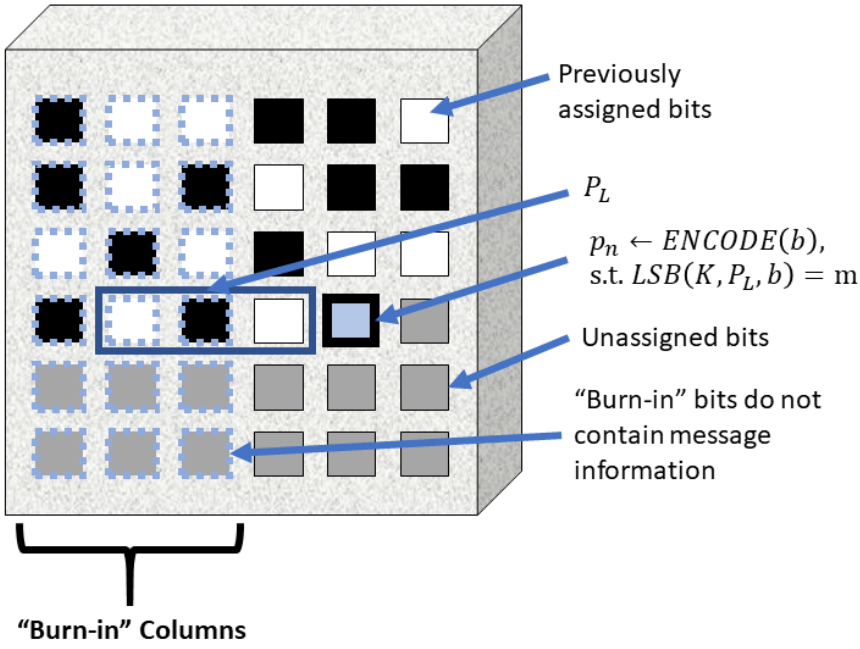

Fig. 6.2. Illustration of the proposed 1-channel open-loop encoding scheme. The schematic assumes bits are written in raster order, proceeding from the top left to the bottom right. The schematic shows the embedding in progress, currently at the bit location marked in blue with bold outline.

6.4.1 Sequential Embedding with One Information Channel

The user embedding the secret message must supply both the message m and a secret key K, where in this analysis both are assumed to be strings of ASCII characters (this assumption is relaxed later in this chapter, allowing for better message compression). I adopt a method for message embedding that leverages the least significant bit (LSB) of $\mathrm{PRF}(K, P_L, p_n)$, where PRF is a suitable pseudorandom function [119] (in this analysis, the Sha-256 hash function is used), P_L is a string containing the previous L pixel values and, optionally, some digest of intrinsic features observed from the part, and p_n is the value of the current pixel. Note that p_n here refers to the *response* (0 or 1 in the binary) elicited from measuring the result of the embedding

process for this spatial location (for instance, varying local layer thickness or phase information).

Similar LSB-based approaches are common in digital steganography [120], and here I propose an extension to physical objects. The scheme proceeds by marching along the surface through time, and at every pixel p_n attempting to embed the next bit b_n contained the message string. So if the message is of length M, we should have the full message embedded after M such steps.

I now describe the bit embedding procedure. A schematic illustrating the process is given in Figure 6.2. At time n, read P_L from the existing surface of the part, and compute r_0 and r_1, where

$$r_b = \mathrm{LSB}\Big(\mathrm{PRF}\big(K, P_L, b\big)\Big) \tag{6.1}$$

for $b \in [0, 1]$. Our goal is to embed the current message bit b_n by choosing b such that $r_b = b_n$. There are three possibilities for each step:

1. **bliss:** both r_0 and r_1 are equal to b_n, so the correct bit will be encoded no matter our choice for b.

2. **bane:** neither r_0 nor r_1 are equal to b_n, so the correct bit will not be encoded no matter our choice for b.

3. **choice:** either r_0 or r_1 are equal to b_n, but not both.

In the bliss and bane cases, we have no control over the output $\mathrm{LSB}\Big(\mathrm{PRF}\big(K, P_L, b\big)\Big)$, so we can choose b at random. In the choice case, we choose to encode the value of b that gives the desired r_b result using ENCODE. Assuming pixel values 0 and 1 each occur with equal frequency and that ENCODE can encode b with no error, this means that for 75% of pixel cases, we can embed the desired result. Thus, some error correction is needed. The simplest linear code that could be used, adopted here for illustration, is to use bit repetition for error correction [121]: for each message bit, embed that bit d times, and take the mode of each group of d bits as the error-corrected bit value. Then the number of incorrect bits, X_{inc}, can be modeled as a

binomial random variable with success probability $p_S = 0.25$ and number of trials d, $X_{\text{inc}} \sim \text{Binomial}(0.25, d)$. The probability of correctly computing the embedded bit given each group can then be easily computed from the cumulative distribution function; for example the probability of getting the correct bit is 0.8438, 0.8965, and 0.9294 for $d = 3, 5$, and 7 respectively.

Still, even for relatively high values of d, there may be other errors inherent in the embedding scheme or possible damage to the surface that could pollute the signal. To address this, consider a character-level voting scheme similar to the bit voting scheme in the previous paragraph. If the message is much shorter than the space available on the surface, simply repeat the message over the surface, and when reading back the message allow each instance of the message to "vote" for each character in the final output. This corrects for any lingering bit-level encoding errors and provides robustness against local damage to the surface between embedding and read time. Further, if repeated over the surface of a part on a sufficient scale, this provides some defense against an adversarial attack to "erase" the message locally, and would also provide a way to recover the encoded message from a part fragment, if necessary.

At this point, it is important to point out that this approach implies a crucial signal processing assumption: errors occur as bit-flips, but not bit-insertions or bit-deletions. That is, the message's length must be known to the reader, and the knowledge of bit locations must also be preserved, even if the specific bit value is not. This is reasonable in the current case, where we assume the physical location of each encoded bit is known.

In summary, the encoding procedure is as follows:

1. Set the secret message m of length $|m|$, key K, bit string length L, and bit repetition count d.

2. Allow random choice of pixels p_n until L pixels have been created. These are the "burn-in" bits in Figure 6.2 which do not carry message information. Note,

if using intrinsic information from some other location on the part, there may be no need for such burn-in bits. See Chapter 7 for a discussion of such schemes.

3. For each new pixel, compute $r_b = \text{LSB}\Big(\text{PRF}\big(K, P_L, b\big)\Big)$ for $b \in [0, 1]$.

4. If r_b is controllable (the "choice" case), set the next pixel value as $p_n = \text{ENCODE}(b)$ such that $r_b = b_n$, with b_n the desired message bit to encode for this pixel.

The reading procedure is then:

1. Using $|m|$ and d to determine where each bit falls in the read message, for each pixel p_n, compute $b_n = \text{LSB}\Big(\text{PRF}\big(K, P_L, p_n\big)\Big)$. This yields a repeating string containing each encoded version of the secret message.

2. Correct for errors by allowing bit-wise voting for every group of d subsequent pixels in the resulting string, and then allow character-wise voting for every character in each instance of the encoded message. Character-wise voting is dependent on how characters were encoded in generating the binary string. In results presented later in this chapter, the most common character across all message instances was reported, with ties broken randomly.

Proof that the embedded signal is indistinguishable from noise: It is sufficient to show that the probability of each bit is 0.5 regardless of the message being encoded. Note the fact that each bit is set as the least significant bit of a pseudo-random function (hash) of a changing bit string with probability 0.75, or is set to the bit value that will yield the message's next bit as the LSB given the rest of the bit string with probability 0.25. In the first case, if the function is truly a PRF, the probability of each bit being 0 or 1 is 0.5. In the second case, by the same argument, the requirement for the next bit being 0 or 1 to produce a hash with LSB 0 or 1 is also 0.5. Therefore each bit value is an iid random variable, $\sim$ Bernoulli(0.5).

As discussed above, this repetition error correcting code approach, though simple, is necessary to handle the high random bit errors introduced by the bliss/bane/choice

encoding scheme. Other approaches to error correction and message compression are discussed later in this chapter.

6.4.2 Extension to Two Information Channels

To improve the robustness of this encoding scheme to localized damage, I present a modest modification to the one presented above that reads the bits in L along *two* spatial directions, where each direction entails an independent encoding channel. In this formulation, the message is embedded using the method described above, however rather than just reading the previous bits horizontally, two encodings are performed with the bits for each encoding interleaved. Every other bit takes as P_L the previous L bits encoded in the horizontal direction, where the remaining bits takes the previous L encoded in the vertical direction. In this way, if an erasure happens in one direction (as a "burst error" in channel coding terminology), say through some scratching damage, then the data in the orthogonal channel should be less affected by that localized damage. See Figure 6.3 for an illustration of this encoding method.

Unless otherwise noted, results in this chapter will always make use of this two-channel scheme.

6.4.3 The Limiting Rate of the Channel

Consider now what limits might exist on the data rate using this channel. Without loss of generality, the channel capacity for this analysis is assumed to be specified in bits per unit area, say bits per cm^2. If the (random) bit error probability is p_{er}, and the (noisy) channel capacity is assumed to be a function of the bit error probability denoted $C(p_{er})$, then the theoretical limit on the rate of the channel $R(C, p_{er})$ is given by the Shannon Noisy Channel Coding Theorem [122] as $R(C, p_{er}) \leq C(p_{er})$, where

$$C(p_{er}) = \sup_{p(x)} I(X; Y|p_{er}) \tag{6.2}$$

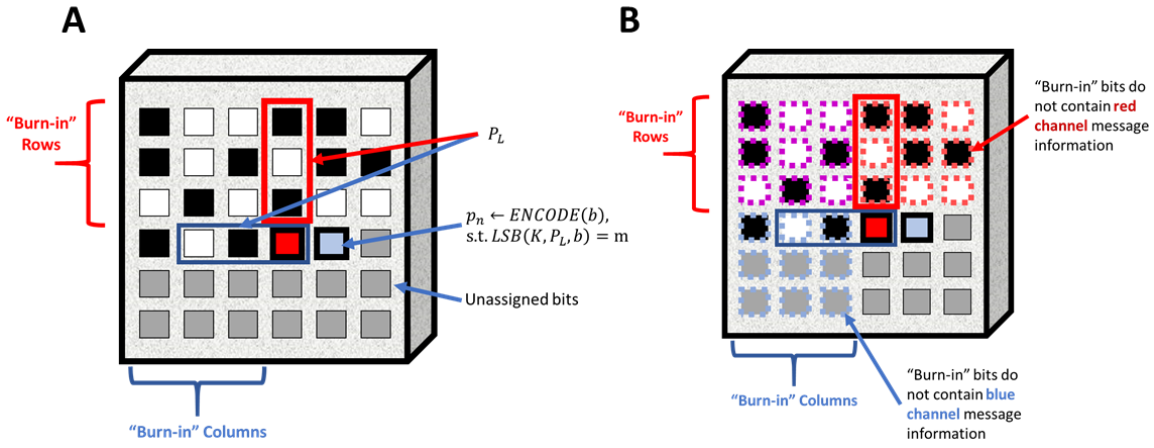

Fig. 6.3. (A) Illustration of the proposed 2-channel open-loop encoding scheme. The schematic assumes bits are written in raster order, proceeding from the top left to the bottom right. The schematic shows the embedding in progress, currently at the bit location marked in blue with bold outline. (B) Illustration of the burn-in bits for each of the two channels. Bits marked in purple contain no message information for either channel.

with X and Y denoting the encoding and decoding symbols, respectively (here, both are random variables over the alphabet $\{0, 1\}$), $p(x)$ is (with some abuse of notation) the probability mass function for X, and $I(\cdot;\cdot|p_e r)$ denotes the mutual information as defined in Chapter 5, conditional on the bit error rate (or more concretely, on the probability mass function $P(Y = i|X = i, p_{er})$, which is assumed to be parametrized by p_{er}).

As the channel transmits one bit per spatial location, the channel can be modeled as a noisy discrete memoryless binary symmetric channel, implying $P(Y = i|X = i, p_{er}) \sim \text{Bernoulli}(1 - p_{er})$, $\forall i \in 0, 1$. In this case it can be shown that the capacity is

$$C = 1 - H_2(p_{er}), \tag{6.3}$$

where $H_2(p_{er})$ is the binary entropy of the bit error rate

$$H_2(p_{er}) = -p_{er}\log_2 p_{er} - (1 - p_{er})\log_2(1 - p_{er}). \tag{6.4}$$

Theoretical limiting message rates as a function of bit error rate are plotted in Figure 6.4, with representative rates for this scheme called out.

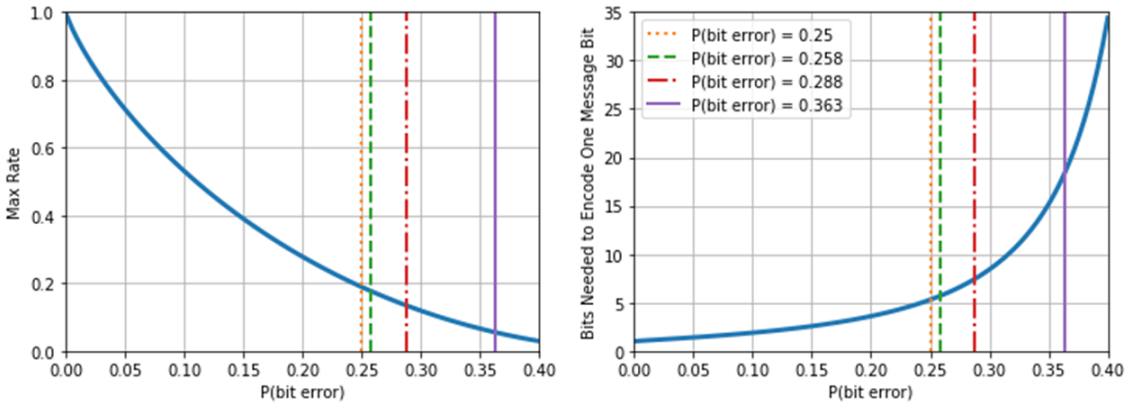

Fig. 6.4. Maximum rate and (equivalently) minimum number of bits required to transmit one message bit with optimal encoding. Points corresponding to probabilities of random encoding errors of 0 (P(bit error) = 0.25), 0.01 (P(bit error) = 0.258), 0.05 (P(bit error) = 0.288), and 0.15 (P(bit error) = 0.363) are called out as vertical lines.

If a larger alphabet can be encoded, say by allowing for three or more symbols at each spatial location, then the capacity as a function of space may in fact increase. Crucially, the new limiting rate would be a function of the joint distribution of the new encoding and decoding symbols, as observed through experimentation with the encoding process during manufacturing, and would of course would require a modification of the bliss/bane/choice encoding scheme to handle alphabets with more than two symbols. Such an analysis should be considered when evaluating possible encoding mechanisms in AM.

Finally, it must be noted that this channel model restricts its assumptions only to errors that would occur *during manufacturing*, namely bit writing errors and errors inherent to the coding scheme. Of course, other errors due to damage or reading after manufacturing will also impact error rates, and the presence of these would need to be accounted for to better characterize limiting bit rates. This analysis,

however, would be highly dependent on the part's transportation and intended use. Such characterization of the "complete" channel will be necessary in future work considering realistic use cases.

In the following section, methods for handling error correction in this environment are considered.

6.4.4 Error Correction and Methods to Improve Data Rates

As discussed above, there are three key sources of error that must be considered in this context:

1. The bliss/bane/choice procedure for assigning bits at each spatial location implies a 25% chance of an incorrect assignment at *each bit*.

2. The encoding method itself, which will be dictated by the manufacturing method and what can be controlled during the manufacturing procedure, may be noisy, implying some probability of an incorrect bit assignment.

3. Decoding may occur after some damage to a part, which will erase or otherwise damage parts of the encoded message.

The first two of these error sources are examples of *random bit errors*, while the third source may manifest as either random bit errors or *burst bit errors*. Thus, error correction for this method must be able to correct for errors of both kinds. Here, this is tackled through *repetition in depth*, where symbols themselves are repeated throughout the embedded message and bits within symbols are also repeated. These approaches may be paired with other methods common in signal processing literature, for instance by concatenating with existing error correcting algorithms capable of handling burst and random errors. The specific techniques considered are discussed below; modification of the key parameters in these techniques and the resulting experimental design and results are discussed in the next section. This is followed by recommendations for implementing such schemes.

Bit Repetition Codes and Symbol Repetition. As already discussed, the first level of defense against the inherently high bit error rate in this encoding scheme is bit-level and symbol-level repetition. Encoded bits are repeated a set number of times d, and the message being sent is itself repeated over the embedding surface until the available space is exhausted. Importantly, note that it is the *plaintext* bit $m_i \in m$ and message that is being repeated. The "cyphertext" pattern of symbols on the surface itself should remain random-looking, as each symbol is a function of $\text{LSB}(\text{PRF}\big(K, P_L, b)\big)$. Then, the message itself is repeated over the available encoding space as room allows, effectively contributing repeated instances of each symbol.

It should also be noted that modest relaxations regarding what can be encoded can imply other ways to address the encoding errors these bit repetition schemes address. For instance, if there is some third symbol, say -1, that can be encoded corresponding to "error," then this could be used to denote a "bane" location and remove that bit from consideration during reading. This approach would open other issues, however: if there is any possible ambiguity in whether a bit location is marked as a "bane" error, say through damage or the reading method, then this introduces *bit addition/deletion* errors that do not arise in the original scheme. This may require changes to additional error correction approaches, and these are not discussed in this chapter.

Symbol Compression using Gray Code Encoding. In this extension to the scheme, the alphabet of symbols that can be used to create the message are reduced and encoded as six-bit codewords, rather than the eight-bit codewords used in ASCII encoding discussed above. This is done by reducing the set of permissible symbols from the full ASCII set to the *ordered* set {ABCDEFGHIJKLMNOPQRSTUVWXYZ .,!? 123456789*}, of size 41. As $2^5 < 41 < 2^6$, six bits are sufficient to encode these characters. Here, Gray code encoding is used to generate a codebook over all members of this ordered set. Gray codes [123], or "reflected binary codes," are generated over an ordered alphabet of symbols such that each symbol's encoding differs from its neighbors in exactly one bit. For example, in the present encoding scheme, the

symbol ‘X’ is encoded as ‘010100’, while the symbols ‘W’ and ‘Y’ are encoded as ‘011100’ and ‘010101’, respectively.

This has two immediate benefits. First, the bits required to represent symbols in the message alphabet are reduced from eight (original ASCII) to six, implying a 25 percent savings on needed space per symbol. Second, the main motivation for using Gray Coding is that random-like bit errors that are not corrected for through repetition will now more likely manifest as “nearby” characters as defined through the ordering in the new, reduced alphabet. That is, errors should be more likely to appear “readable” to the human if they arise. It should be noted that since there are 64 possible six-bit code words, but only 41 codewords assigned to symbols, the algorithm returns an error symbol ‘*’ if an unassigned codeword is read.

6.4.5 Comments on Other Possible Extensions

Burst error correction with Reed Solomon encoding. Reed-Solomon (RS) codes [124] are a class of block error correction codes that are well-suited to correcting burst errors, where it is assumed that nearby bits are corrupted together. This makes them popular in use cases like CD reading, where “scratches” on the disc fit this error model well. Due to large random-like bit error rates inherent in this encoding scheme, RS codes alone are not appropriate for handling the needed error correction. However, schemes incorporating RS and codes more appropriate for random-like errors (like convolutional codes), may be more appropriate.

Random bit error correction with convolutional codes. Convolutional error correcting codes are a class separate from block codes like Reed-Solomon codes [125], and allow for high user control over the data rate through so-called “puncturing” methods [126]. The maximum-likelihood sequence that would generate such a code can then be recovered through application of the Viterbi algorithm [127]. However, convolutional codes are not ideal for correcting burst-type errors, and so perform poorly for striation-like damage. The effect of burst-type errors on the readability of

the recovered message may be alleviated in part through interleaving of the transmitted data [128, 129], at the cost of added complexity (which may not be tolerable during real-time manufacturing processes) and the need to retain information regarding the interleaving process for decoding purposes. Further, for schemes with large P_L lengths, messages recovered after applying convolutional error correction would still be sensitive to random-like damage, as this contributes to incorrect values passed to the PRF through P_L. A combination of convolutional codes with interleaving and block codes like Reed-Solomon codes may be appropriate to account for both types of error, improving performance over the repetition schemes presented here.

Error correction with combinatorial group testing. Combinatorial group testing allows for provable error correction performance as long as the number of "defective" bits in the recovered message are known [130,131]. However, such methods require storing data for each message necessary to perform the group testing on the corrupted received message, and also require knowledge on the expected number of incorrect bits in the message. This may be prohibitive in scenarios where this knowledge, or the necessary storage, in not available.

6.5 Embedding Scheme Simulation and Analysis

Finally, let's apply this encoding method and bit-repetition error correction to study the procedure's performance for synthetic test data under random-like errors and burst errors present in the face of, say, striation damage like that studied in Chapters 3 and 4.

6.5.1 Methodology

Experimental Design. The study was carried out by manipulating the channel rate (through varying the message length, encoding area size, and P_L length) and damage type and severity (simulating random-like errors though flipping bits at random and burst-like errors through applying simulated striation damage). Variables

and other parameters for this analysis are summarized in Table 6.1. For illustration purposes, example encoding surfaces before and after random error application (bit flipping) and burst error application (striation damage, at an angle of 45 degrees) are given in Figures 6.5 and 6.6, respectively.

Table 6.1.
Open-loop encoding experimental parameters.

Encoding Parameters		
Parameter	*Values*	*Units*
Surface Dimension	64x64, 100x100	Pixels
Number of previous pixels used in hash, $\|P_L\|$	8, 16	Pixels
Message	PURDUE, PURDUE PETE, PURDUE PETE RULES	Gray code encoded string
Key, K	ThisIsMyPassword	ASCII encoded string
Bit Repetition	3, 5, 7	Bits
During-Manufacture Damage Parameter		
Damage Type	*Parameter*	*Values*
Incorrect Encoding Value	Probability of error p_{er}	0.01, 0.05, 0.15
Post-Manufacture Damage Parameters		
Damage Type	*Parameter*	*Values*
Bit Flip (Random Errors)	Probability of flip p_{flip}	0.02, 0.05, 0.10
Striation (Burst Errors)	Number of striations n_{stri}	2, 4, 6

Simulated Damage. *During-manufacture* damage was considered by allowing for bit values to be incorrectly set during the writing process with a probability $p_{er} \in [0.01, 0.05, 0.15]$. Experimentally, this means that with probability p_{er}, a pixel value is encoded with the wrong value. Crucially, the encoded value may still be

read when computing P_L for future pixel value computations. The result is that the single pixel response will be incorrect when read, but nearby pixel readings should be unaffected if no post-manufacture damage was applied.

Two types of *post-manufacture* damage were simulated. First, random-like errors were considered using "bit-flipping" damage, where each bit in the encoded surface was "flipped" to the opposite value with a probability $p_{\text{flip}} \in [0.02, 0.05, 0.1]$. This is intended to model possible distributed damage to a surface that may occur during processing or transportation, or poor reading during a challenge. Second, burst-like errors were considered using striation damage, where set regions of pixels were set to the same value of "0." These pixels corresponded to regions contained within n_{stri} designated striation locations, depending on the severity of the damage being considered. For this analysis, $n_{\text{stri}} \in [2, 4, 6]$.

6.5.2 Results and Discussion

Results for these experiments are summarized as Hamming distance box plots in Figures 6.7 through 6.10, quantifying performance as the Hamming distance between the original encoded message and the message read after damage was applied (as in previous chapters, lower Hamming distance implies better performance, and a distance of 0 implies the original message was perfectly recovered).

The Figures each present an array of boxplots for a set of experiments run for different combinations of encoding surface size and P_L length. For these Figure series, the left 3x3 plot arrays (colored green) correspond to random-like bit flip error simulation, while the right 3x3 arrays (colored blue) correspond to burst-like striation damage. The top-most row of each array corresponds to during-manufacture encoding error probability $p_{er} = 0.01$, with the middle rows corresponding to the higher values 0.05 and 0.15 respectively. The left-most column of each array corresponds to bit repetition parameter $d = 3$, with the middle and right-most columns corresponding to $d = 5$ and $d = 7$ respectively.

Fig. 6.5. Original surface with binary encoding alphabet, encoding the string "PURDUE PETE" with 2-channel scheme, and resulting surface after random-like bit flip damage. Encoding errors occurred with a probability of 0.01 at each bit location. Bits were flipped after embedding with a probability of 0.05. 3-bit repetition codes were used, and all characters in the final string were represented using 6-bit grey code encoding.

Tradeoffs in Choice of $|P_L|$. Comparing results between Figure 6.7 and 6.8, it can be seen that performance is universally better for smaller $|P_L|$ (8 rather than 16 bits). This stems from two contributing factors. First, a smaller $|P_L|$ implies fewer "burn-in" bits must be included the embedding surface, leading to slightly lower effective data rate and therefore slightly more damage tolerance. Second, smaller $|P_L|$ implies fewer bits must be passed to the PRF to read each message bit. Since $\mathrm{PRF}(K, P_L, p_n)$ requires P_L to be identical to that provided during embedding to return the same value, a shorter P_L is more likely to lead to a correct evaluation of $\mathrm{PRF}(K, P_L, p_n)$, and therefore a correct read bit $\mathrm{LSB}\big(\mathrm{PRF}(K, P_L, p_n)\big)$. This *introduced fragility* may in fact be desirable in some use cases, where a manufacturer may

Fig. 6.6. Original surface with binary encoding alphabet, encoding the string "PURDUE PETE" with 2-channel scheme, and resulting surface after striation damage. Encoding errors occurred with a probability of 0.01 at each bit location. 3-bit repetition codes were used, and all characters in the final string were represented using 6-bit grey code encoding.

want to detect when the encoding surface has been overly damaged. Importantly, this is *tunable* by the manufacturer's goals. This "deliberate fragility" concept is discussed further in Chapter 7 in the context of closed-loop schemes.

Effect of Data Rate. Lower data rates occur when i) shorter messages are ii) transmitted using greater bandwidth (larger encoding surfaces). In general, lower-rate experiments result in better performance (lower Hamming distances). This is to be expected, as lower-rate transmission should have better performance given identical damage and error correction behavior (see Figure 6.4). Performance for very high data rates (see especially Figures 6.7 and 6.9) is quite poor for moderate to high random- and burst-like errors, suggesting a steep drop-off in scheme quality as these

errors become more prominent. Manufacturers will have to carefully study these damage types and enforce rates consistent with good performance at desired damage profiles. Rates vary between 0.004 ($\frac{1}{250}$) and 0.04 ($\frac{1}{25}$), significantly lower than the most conservative bound specified in Figure 6.4 (about $\frac{1}{18}$ for the highest probability of encoding error), although i) the Figure results do not consider additional bit errors due to damage and ii) given unexplored potential extensions in the error correction, the encoding may still benefit from other error correction implementations. Indeed, these results suggest there is still much future work to be done in optimizing the rates of these encoding schemes.

Effect of Bit Repetition for Error Correction. Across damage types and data rates, it appears varying the bit repetition has relatively little effect, compared to varying $|P_L|$ and the data rate itself. This suggests the impact of the inherent per-bit error implicit in the bliss/bane/choice encoding scheme is corrected well-enough at relatively low bit repetition values (3, rather than 5 or 7). Note varying bit repetition has little impact on the effective data rate as greater bit repetition reduces the number of repeated message instances in the encoding surface. Given that there is then little to no data rate tradeoff, a manufacturer may still want to check multiple settings to determine the optimal value. This investigation may be more important in scenarios where encoding errors for the manufacturing method are poorly understood, or when imaging is unreliable; in such scenarios correcting for these errors at the bit level may yield better performance.

Effect of During-Manufacture Errors vs. Post-Manufacture Errors. Across damage types and encoding surface sizes, during-manufacture errors, quantified through p_{er}, appear to have a low to moderate impact on performance. This is in contrast to the fairly severe impact that post-manufacture errors have on the Hamming distance. Consider for instance the impact of increasing the probability of bit flips from 0.05 to 0.1 for the 64x64-pixel surface, $|P_L| = 8$ case (see Figure 6.10). For all nine cases considered with this damage profile, the median Hamming distance jumps by about 0.1 to 0.15 between these two damage settings. This behavior holds,

albeit not quite as pronounced, for striation damage, jumping between the conditions of 4 and 6 applied striations. For lower-rate scenarios (see for instance Figure 6.8), this behavior is observable but again not quite as pronounced.

Closing Thoughts. The scheme studied in this Section is of course an abstraction of more realistic scenarios that would need to be implemented. The goals of this abstraction were i) to establish a framework for representing uncertainties in the encoding process and part use as encoding and damage models, and ii) to demonstrate how a scheme could be evaluated for performance given those representations, in terms of efficiency (data rate) and error quantified through the Hamming distance. This abstraction is general enough to be applied to many classes of embedding schemes, including closed-loop schemes. Importantly, in the spirit of the anti-counterfeiting scheme discussed in Chapter 5, this embedding procedure is *tunable*: a manufacturer can specify key components, such as $|P_L|$, the alphabet of embeddable symbols, and error correction methods, to best suit the use case at hand. This analysis is a modest illustration of the *malleable PUF* design, implementation, and evaluation process.

Of course, such a procedure must be further validated in a practical environment, and further extensions are still possible. Extensions to these schemes for practice are discussed in the following chapter.

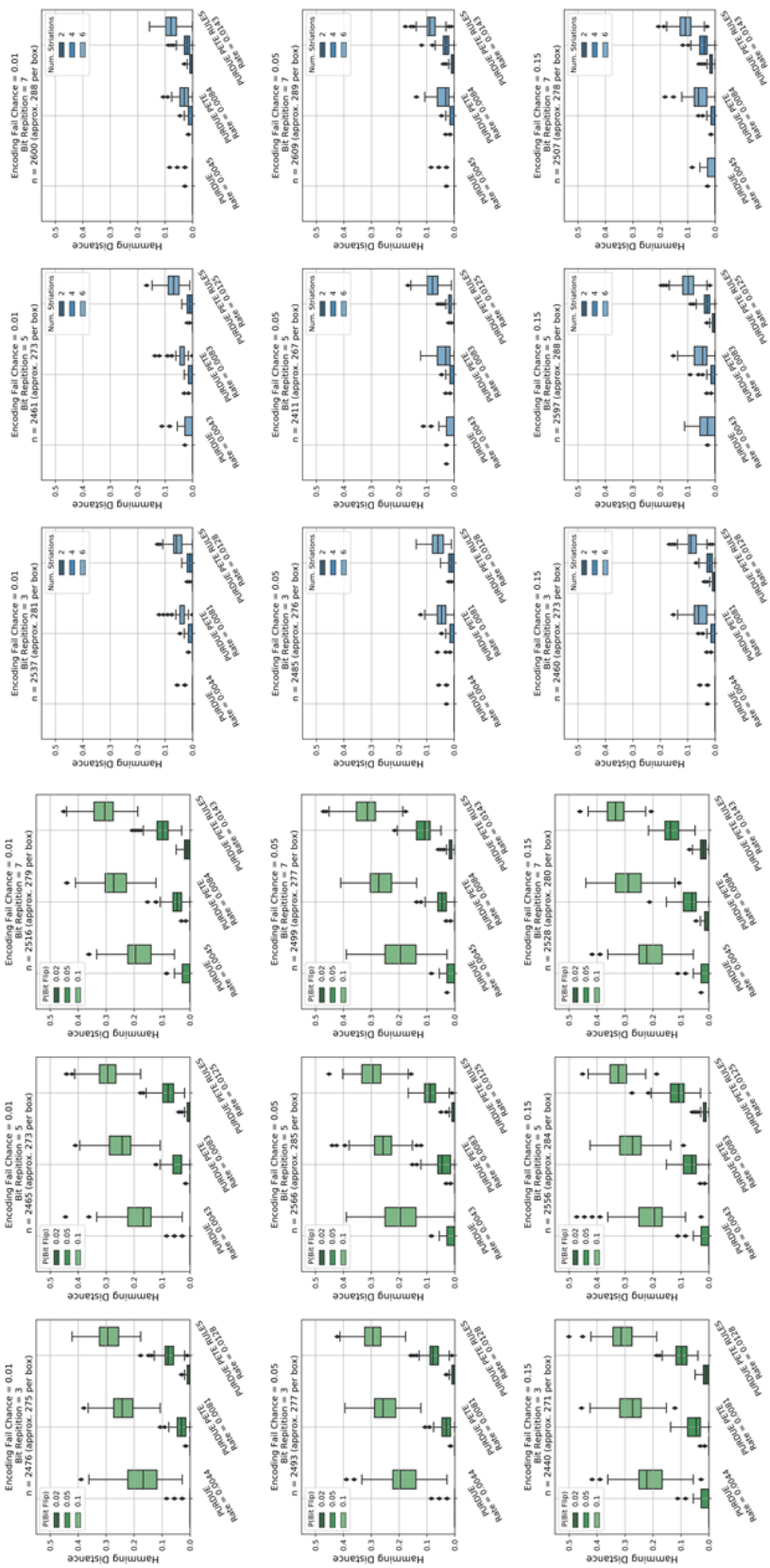

Fig. 6.7. 100x100-pixel encoding performance, with $|P_L| = 16$ pixels. Results for random-like bit flipping damage (left) and striation damage (right).

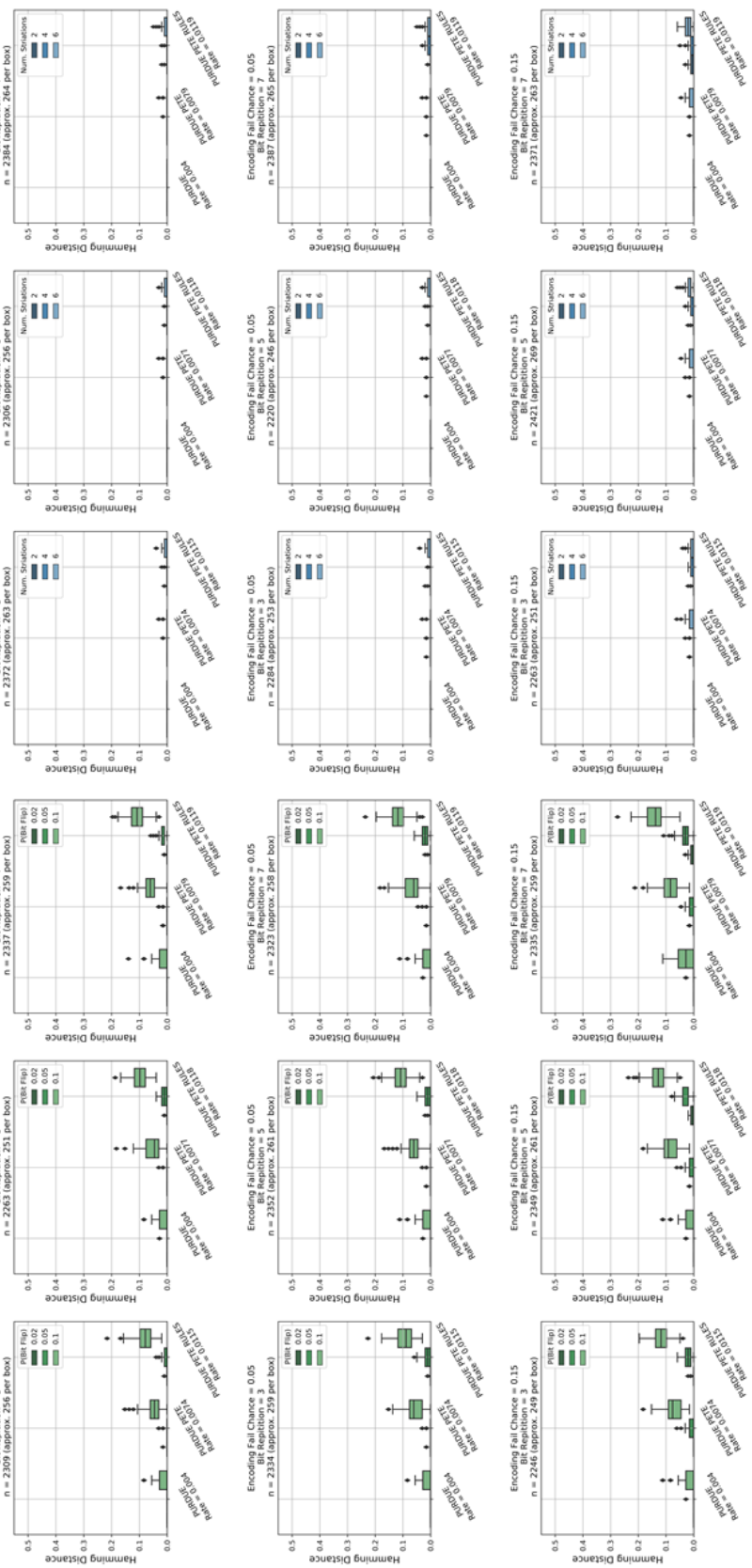

Fig. 6.8. 100x100-pixel encoding performance, with $|P_L| = 8$ pixels. Results for random-like bit flipping damage (left) and striation damage (right).

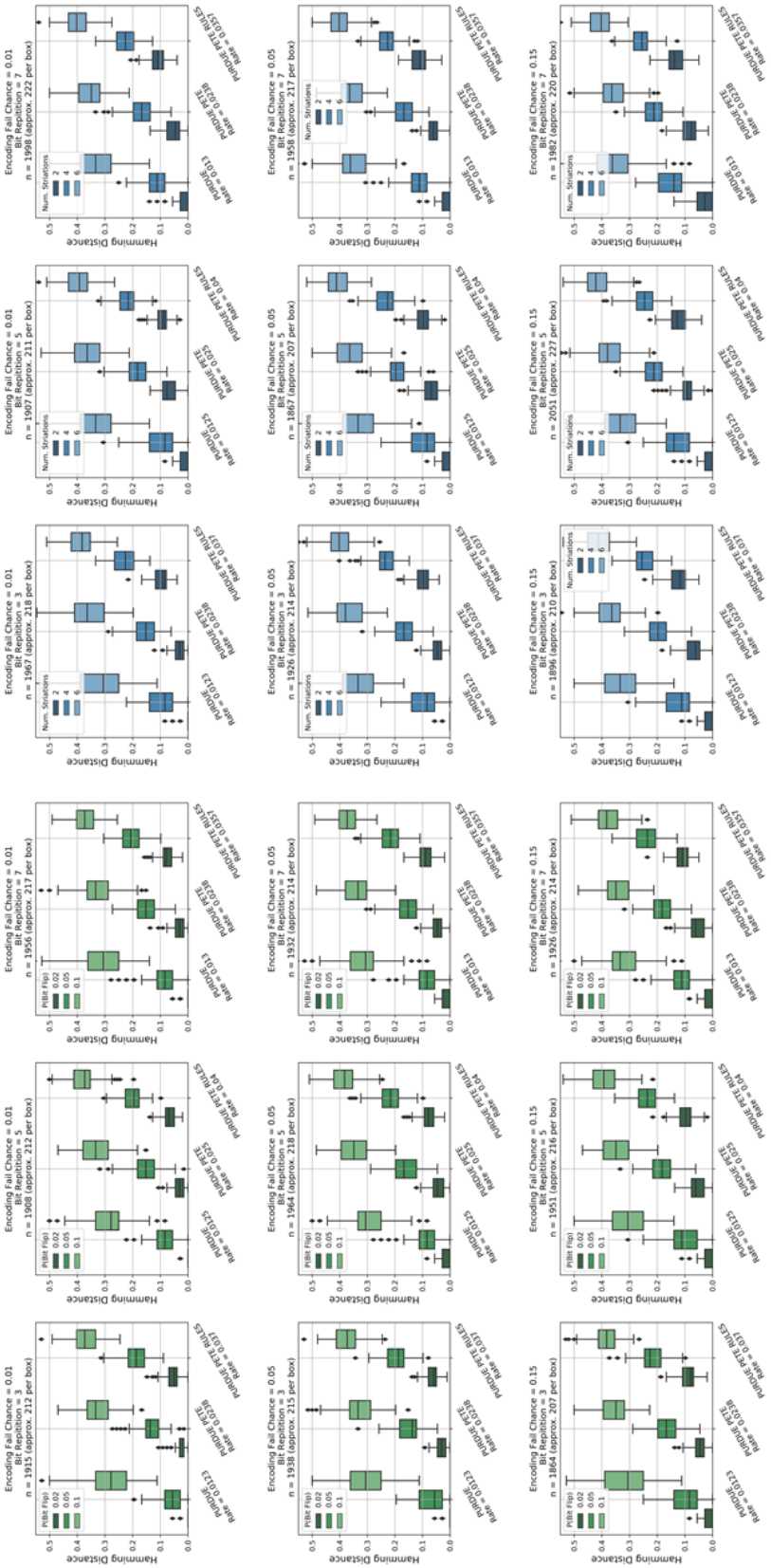

Fig. 6.9. 64x64-pixel encoding performance, with $|P_L| = 16$ pixels. Results for random-like bit flipping damage (left) and striation damage (right).

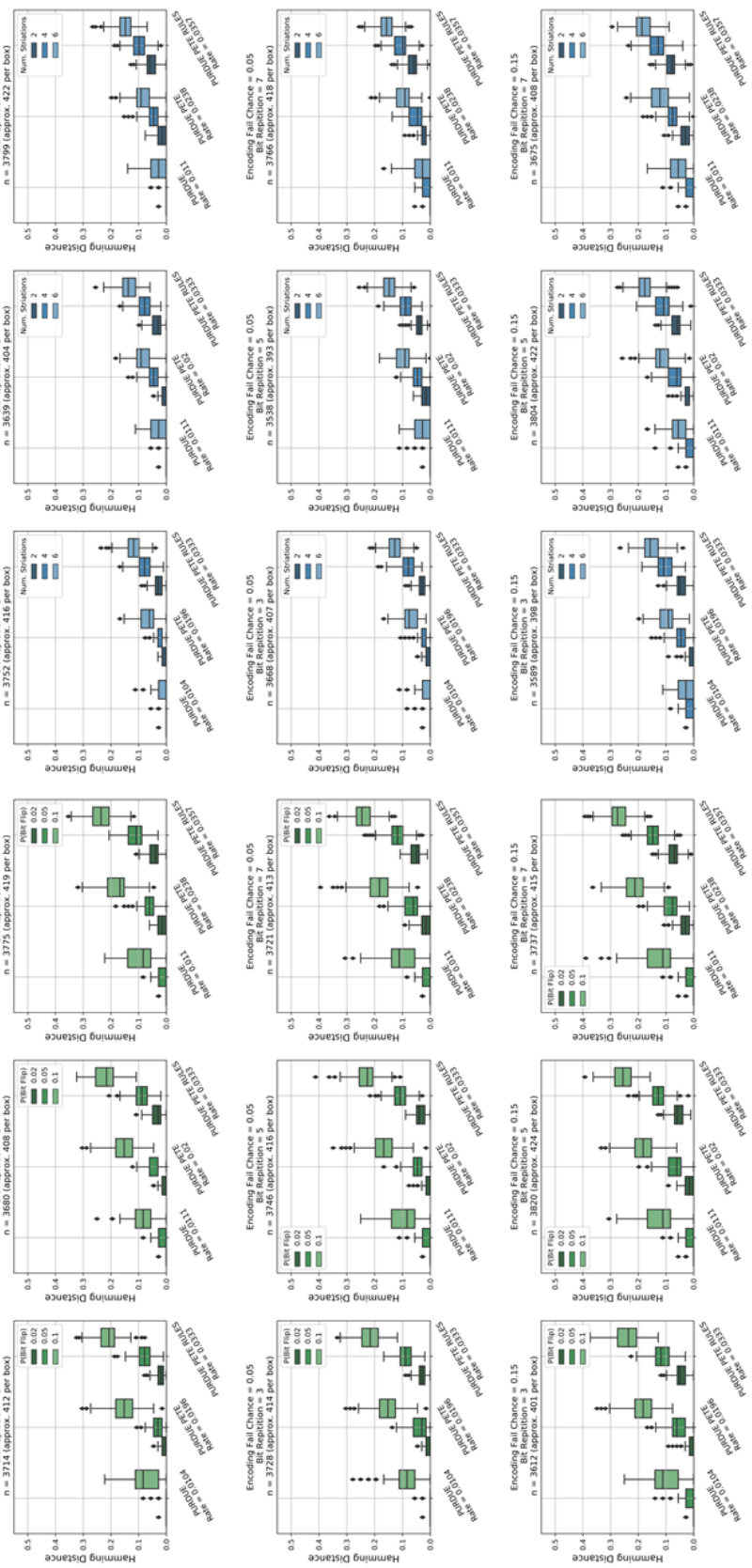

Fig. 6.10. 64x64-pixel encoding performance, with $|P_L| = 8$ pixels. Results for random-like bit flipping damage (left) and striation damage (right).

7. DESIGNING SECURE EMBEDDING SCHEMES FOR ADDITIVE MANUFACTURING

In previous chapters, evidence for the existence and value of inherently random features for traceability and anti-counterfeiting in manufacturing has been presented (Chapters 3 through 5), and possible schemes for leveraging this inherent information for *embedding information* within manufactured goods have been discussed (Chapter 6). In this penultimate chapter, I consider methods for *implementing* these schemes in practice. These methods are inspired by the literature in PUFs, fuzzy extractors, and AM characterization discussed in previous chapters. In the vein of the traceability scheme discussed in Chapter 5, the focus of this chapter is developing a *general framework* that allows malleable PUFs to be easily and consistently evaluated for *tailorability to specific use cases*.

Fusion Deposition Modeling (FDM) 3D printing is presented as a running example use case to guide discussion, although the general approach could be extended to other additive manufacturing processes given a properly characterized *alphabet* and expected damage or adversarial attacks. With this example, the chapter then discusses the concept of alphabet characterization in the context of malleable PUF design, and proposes several extensions to the open-loop embedding scheme discussed in the last chapter.

7.1 Hardware-Intrinsic Embedding

As presented in Chapter 6, recent work in information embedding has not addressed the goals of *malleable PUFs*, namely writing PUF-queried messages to parts with high information content. I argue that for hardware-intrinsic traceability, such tailored schemes will be necessary. Further, the tailorability of these schemes is crucial

to account for the wide variety of design and manufacturing processes (and the relevant attack vectors these processes open up), part treatments, and use cases that are present in industry today. This catalog will grow as additive manufacturing matures and becomes more widespread. As new use cases are considered, it will be necessary to establish the proper cyber and physical implementations to enable malleable PUF embedding, as illustrated in Figure 7.1.

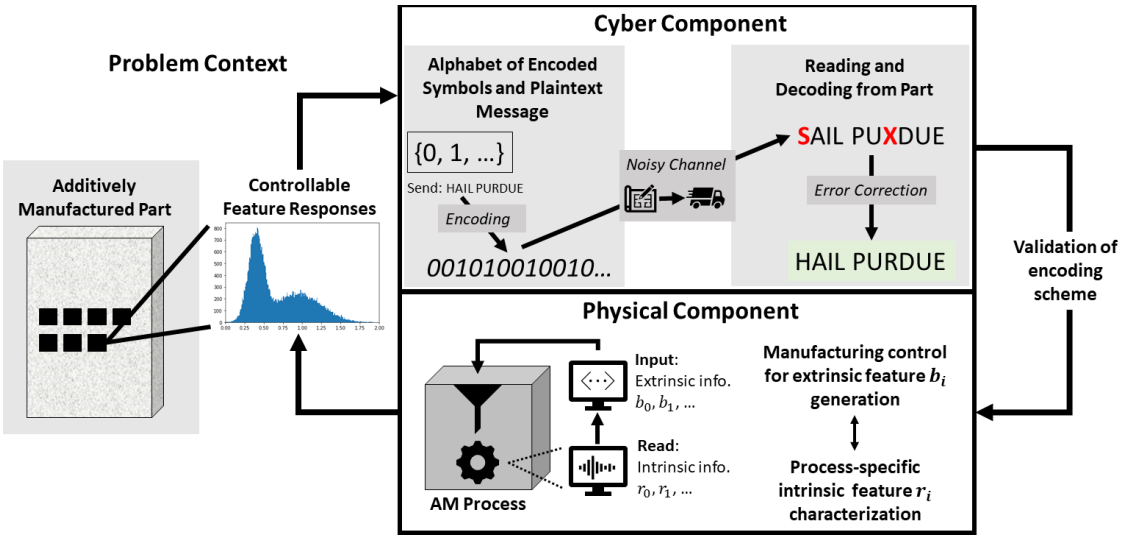

Fig. 7.1. Hardware intrinsic security schemes in AM will require characterizing features that can be used to encode context-specific symbols (physical domain), and embed these reliably and securely (cyber domain).

In additive manufacturing, real-time control over topology and material properties offer plenty of room for investigation of novel embedding schemes. However, progress in this area will need to track the actual needs and constraints of manufacturers that would actually implement these schemes. The goal of this chapter is to outline modest steps towards implementing and evaluating these schemes for practical contexts.

7.1.1 A Running Example: Fusion Deposition Modeling for Information Embedding

For illustration purposes, consider *open-loop* encoding schemes like that outlined in Chapter 6. In the simplest case, encoded symbols are not a function of random-like intrinsic information of the part, but rather specified before manufacturing begins. However, note that encoded symbols may indeed be a function of other symbols in the message.

Such a scheme could be implemented in FDM to embed a message by using external actuators or printing process parameters to generate symbols at different spatial locations during printing. For example, for a given 2D binary array encoding a message, set each row in the array to correspond to a layer of the printed part. Then, set symbol "1" in that layer to correspond to the region of that layer where some actuator (say, a vibration motor mounted to the print head or bed) is active, and "0" to regions where it is inactive or set to a different level. Assuming there exists some feature(s) that can be easily evaluated that differentiate between these regions, this could be used as a simple encoding method. Alternatively, symbols could be encoded by varying process parameters during printing and dynamically changing some observable local features, such as layer thickness, as proposed by Delmotte [113]. Subsequent investigation would allow characterization of potential symbol alphabets for encoding extrinsic information in both open and closed-loop schemes, and these symbols could be evaluated for their usefullness following evaluation methods like those presented in Chapters 4 and 5.

7.1.2 Why FDM as an Example?

As is evidenced by the literature in FDM watermarking and information embedding, FDM is a common, well-understood additive manufacturing process. Aside from familiarity, FDM has several properties that make it useful for information embedding experiments. First, FDM is relatively cheap, with consumer desktop printers

available for only several hundred USD.[1] Thus modification is inexpensive and, from resource and safety perspectives, relatively low-stakes. Second, FDM is a relatively simple layer-wise AM process with a well-established workflow for creating prints: a geometry file (like an .STL) is validated and processed with some "slicing" program that outputs a file containing machine commands (usually a .gcode file) that capture both the intended geometry of the part and the user's specified print parameters (such as layer thickness, precision, and so on), which is then passed to the printer firmware to actually print the product. This passing is often done through some host software that allows the user to monitor and modify the build as it executes. With the rise of open-source slicer software like Slic3r [132], interface software like Printrun[2], and printer firmware, implementing schemes that modify the geometry or printing process itself is relatively straightforward. A user can modify G-code over time and manipulate machine commands, including commands to external actuators if desired, based on an extrinsic encoding scheme, feedback from the printing process, or both.

7.2 Using Intrinsic and Extrinsic Features for Malleable PUF Design

Much of this dissertation has been concerned with characterizing *intrinsic* microstructural features for their usefulness in traceabilty schemes. This analysis considered i) robustness to damage and ii) the discriminatory value of the information derived from the features, both of which are sensitive to the use case, material properties, and processing of the product. To more clearly extend this analysis to extrinsically introduced features, consider the following notation. Note this discussion follows the notation used in Figure 7.1.

Let r_i denote a bit of information derived from some intrinsic part feature, whose index i indicates the location on the part corresponding to the bit and whose value is not controllable by the manufacturer, or at least is exceedingly difficult to control.

[1] At time of publication, the Creatly3D Ender-3 desktop printer, with PLA and ABS filament support, heated bed, 8.6x8.6x9.8-inch build footprint, and 0.1mm precision is selling for $369.

[2] Available: https://github.com/kliment/Printrun

Without loss of generality, this concept can be extended to incorporate symbols of an alphabet defined over intrinsic feature responses, where a symbol would consist of one or more r_i; an example would be the bit values corresponding to the 2-point statistic feature responses studied in Chapter 5.

Then, let b_i denote a bit of information derived from some extrinsically introduced feature, again indexed by the location on the part corresponding to the bit. The key point is that the value of b_i is deliberately written to the part in some controllable way. Considering our running FDM example, this may be through deliberately altering the layer thickness at location i to encode a symbol of "1."

The interplay between the intrinsically observed information r_i and the extrinsically written information b_i is the essence of the schemes discussed in this chapter. This is also true for the schemes discussed in Chapters 3 through 5, only those cases represent those "embedding" schemes where b_i is empty. In *open-loop malleable PUF schemes*, the embedded b_i is independent of the intrinsic r_i, whereas in *closed-loop malleable PUF schemes*, each b_i (or a string $[b_i, b_{i+1}, \ldots b_{i+m}]$) is embedded after r_i is observed, so that the overall bit sequence becomes $\sigma = [r_1 b_1 r_2 b_2 \ldots r_m b_m]$. Here, it is assumed that b_i is computed *as a function* of the previously observed $r_1, \ldots, r_i$ and the already-embedded $b_1, \ldots, b_{i-1}$; it cannot depend on any of $r_{i+1}, \ldots, r_m$ because these are not known (or knowable) at the time the manufacturing process embeds b_i. The following subsection aims to formalize these concepts.

7.2.1 The Generic Closed-Loop Malleable PUF

Let X denote the physical location used for information embedding. This is assumed to take the form of an ordered list of N spatial locations $[X_0, X_1, \ldots, X_i, \ldots, X_{N-1}]$ to which individual bits (or other symbols) are written/read. (Running example: this could be a rectangular region on a printed part designated for embedding, segmented into N pixels in raster order). $\hat{X}$ denotes the surface after embedding and damage transformations have been performed; $\hat{X}$ is the surface a challenger will

have to query. The "hat" will be used to denote other responses read after damage has been applied.

Let Ω_i denote the intrinsic information relevant to spatial location i (running example: this could be some uncontrollable variation in surface roughness or other microstructural measurement like phase volume fraction). It is then assumed that Ω_i is inherent to X (but not necessarily just X_i; for instance it may include information about the last few locations as well), and that an r_i value is a function of Ω_i (running example: let $r_i = 0$ if the measured volume fraction for phase 1 is less than the historical median, and let $r_i = 1$ else).

Let θ_i denote additional information specified by the manufacturer that allows an intended user to read the plaintext message during a challenge. Importantly, this information captures the uniqueness of whichever scheme is being studied. θ_i must include a cryptographic key K known to the manufacturer and downstream authorized users, and additionally may include a password, the current portion of the cyphertext being written m_i. Call this information *specified* before manufacture θ_i^s. θ_i may also include information about previously encountered r_i and b_i values that must be computed during the challenge process; call this information *read* from the existing surface of the part θ_i^r (running example: this could simply be a vector of the previous n values for r_i and b_i). Then, $\theta_i = \theta_i^s || \theta_i^r$.

Let τ_i denote the extrinsic response of spatial location i (running example: this could be a controllable layer thickness measurement, varied by actuation of a vibration motor or dynamically altering the nozzle feed rate). τ_i, combined with other information θ_i, should yield b_i when queried by an authorized user, assuming the damage to X has not been too great.

Then, given a location X, a message m, a manufacturer's specified θ_i^s, and the necessary θ_i^r, the *malleable PUF embedding and challenge processes for closed loop control* are given by:

1. **Writing:**

(a) A **character writing function** f that computes the desired extrinsic information to be written to location X_i:

$$b_i \leftarrow f(\Omega_i, \theta_i, m_i),$$

where m_i is the message information relevant to location i.

(b) An **embedding process** Γ that writes the desired extrinsic information to the manufactured part at location X_i:

$$\tau_i \leftarrow \Gamma(b_i, \theta_i).$$

2. **Damage Model**:

(a) A **damage process**, or set of processes, Υ that applies acceptable damage to the embedding location and outputs a corresponding modified representation of X,

$$\hat{X} \leftarrow \Upsilon(X).$$

In the previous chapter, such models included random bit-flipping errors and striation damage. This model could also include the imaging, translation, and pitting damage profiles considered in Chapter 5, or any combination of the above.

3. **Reading:**

(a) A **recovery process** Ψ that, at challenge time, recovers necessary information for each spatial location X_i from the manufactured part:

$$(\hat{\Omega}_i, \hat{\tau}_i, \hat{\theta}_i^r) \leftarrow \Psi(\hat{X}, \theta_i^s).$$

$\hat{\Omega}_i$, $\hat{\tau}_i$, $\hat{X}$, and $\hat{\theta}_i^r$ denote the (possibly changed) intrinsic feature responses read, the (possibly changed) extrinsic feature responses read, the (possibly changed) information embedding location, and the (possibly changed) ancillary read information recovered at challenge time, respectively. Call the (possibly changed) additional information vector $\hat{\theta}_i = \theta_i^s || \hat{\theta}_i^r$.

(b) A **character reading function** g that computes the relevant message information for each location i,

$$\hat{m}_i \leftarrow g(\hat{\Omega}_i, \hat{\tau}_i, \hat{\theta}_i),$$

where $\hat{m}_i$ denotes the (possibly changed) message data.

The term *process* in the above definitions refers to some physical process (manufacturing, measuring, etc.) that must be followed to physically embed the needed information at spatial location i, alter the part, or read information needed to recover a message. *Function* refers to some computation performed on the relevant data.

Thus a full malleable PUF scheme is specified with $(X, f, \Gamma, \Psi, g, \theta_i)$[3]; each of these components must be carefully considered when designing a scheme. If the scheme is "good," then the embedded message m should be identical or very similar to the recovered message $\hat{m}$ in the presence of acceptable damage processes Υ applied to the part between manufacture and challenge time. The measure of similarity depends on the allowable damage and corresponding errors; if bit values are expected to change but message length will remain constant, then Hamming distance makes sense. If the message length may change due to bit insertion or deletion, an edit distance could be used.

7.3 Potential Malleable PUF Embedding Schemes

Armed with this generic formulation of the malleable PUF, and with the FDM process as a running example, I now consider potential additive manufacturing embedding schemes. Note that it's assumed intrinsic and extrinsic feature characterization has been carried out; that is, I'll assume the alphabet of intrinsic and extrinsic features have been sufficiently studied as to ensure the resulting r_i and b_i are reliable

[3]In order, (surface, character writing function, embedding process, recovery process, character reading function, supplementary information)

enough to be effective. I want to stress again that this characterization is critical in practice, and will be the topic of future work.

Consider now a scheme that embeds only genuinity information; that is, the message is simply "this part is genuine," in the vein of schemes discussed in Chapters 3 through 5. As we shall see, an extension to encoding a more complex message will follow naturally.

Denote $H_K(\cdot)$ as a cryptographic one-way hash function that is keyed using a cryptographic key K. Let S denote some identifying signal imposed by the manufacturer, such as the serial number of a manufactured part. During the writing process, the bit b_i that is embedded in the part immediately after reading random bit r_i from some intrinsic information Ω_i is computed as $b_i = \text{LSB}(H_K(S||i||r_i))$, where "||" denotes concatenation. During the reading process to test for genuinity at challenge time, the following is done for every r_i, b_i pair:

- First r_i and b_i are measured, then r_i is used to compute $\text{LSB}(H_K(S||i||r_i))$, which is compared to the measured b_i: If they are equal, then "i voted that the part is genuine." Else, say that "i voted that the part is counterfeit".

In expectation, a counterfeit part will have about as many favorable votes as unfavorable votes (not far from $m/2$). For a genuine part, the vote will be close to m (with some variability caused by the r_i or b_i that were accidentally modified through damage Υ to the part).

Thus the full malleable PUF is specified with:

- X: The surface of the part used for embedding the b_i's.
- f: $\text{LSB}(H_K(S||i||r_i))$.
  - Ω_i: Microstructural information at location X_i, such as phase information, used to generate r_i, determined by manufacturer.
- Γ: The embedding method used to encode $LSB(H_K(S||i||r_i))$, determined by manufacturer.

- Ψ: The measuring method for recovering $\hat{i}$, $\hat{r_i}$, and $\hat{b_i}$, determined by manufacturer.
- g: $LSB(H_K(S||\hat{i}||\hat{r_i}))$.
- θ_i^s: The serial number of the part, S, and the key K.
- θ_i^r: Current location index i and current intrinsic response r_i.

The manufacturer would then create an experimental design to determine the fitness of a proposed X, $f(\cdot)$ and corresponding Ω_i, Γ, Ψ, and $g(\cdot)$ in the face of a specified Γ, like those proposed in Chapters 3 through 6.

Encoding a more complex message. This scheme could be extended to encode a more complex message by modifying f to select b_i to correspond to the i^{th} bit of a (possibly encrypted) message string. This approach is similar in spirit to the b_i determination discussed in the open-loop scheme presented in Chapter 6, but now by design each embedded b_i is a function of extrinsic information (S, K, m) and intrinsic information r_i.

Deliberate Fragility. A manufacturer can introduce an intended and tunable level of fragility to damage by investigating more complex r_i, similar to how P_L was formulated in the open-loop scheme of Chapter 6. This fragility may be desired if a manufacturer wants the reading to fail if excessive damage or sabotage occurs. Let the r_i that is read prior to embedding b_i be a bit string rather than a single bit or character. If the damage probabilities to different bits of r_i are independent, then the fragilization would be substantial as the probability of damage to r_i would substantially increase with the number of bits in r_i (and be much higher than if r_i had been a single bit). This is observed in simulation in Chapter 6, where random-like damage produces quite bad performance, especially when the number of bits considered in the string is longer. For burst-like errors, this impact may still be present, but not as pronounced. If a manufacturer anticipates damage that may lead to random- or burst-like errors for the chosen Ω_i and r_i, this information could be used to inform an experimental design investigating the impact of differing r_i lengths.

Keyed Voting Schemes. At the cost of slightly more information to store, this voting scheme may be extended to incorporate subset selection of locations in X designated for encoding, based on some "location" key K_{loc}. With n possible locations to embed, such a scheme gives $n!$ "next locations" for the next reading location, creating a very large space for the would-be attacker to search. This method could also be a way to select a relatively small subset of regions to embed. The manufacturer may then set the remaining locations to encode random, noise-like bits for further obfuscation.

8. CONCLUSION AND FUTURE WORK

Traceability is a large and fascinating concept. The challenges posed by potential adversarial attacks in additive manufacturing (AM), and globalized manufacturing in general, require creative solutions that must incorporate information from multiple stages of the product lifecycle: design, manufacturing, and use. In this dissertation, I have presented approaches to traceability that leverage *intrinsic* information generated by the physics of the manufacturing process, and the ability to embed *extrinsic* information (messages) through the dynamic manipulation of manufacturing parameters or external actuators. These approaches are presented according to a framework that allows the manufacturer to *tailor* traceability schemes to their specific scenario, considering unique intrinsic features, expected damage profiles, and constraints on message length and bandwidth.

The work presented in this dissertation is a modest step along the path to widespread, validated manufacturing traceability schemes that admit tailored implementations. In investigation of **Research Question 1**, optical data has been investigated as a source of identifying information with several experimental and synthetic case studies (RQ1.1), and features derived from this data, including areal, lineal, and 2-point statistics features, have evaluated for both robustness to damage models (RQ1.2) and information carrying capacity (RQ1.3). In investigation of **Research Question 2**, I have presented frameworks that can leverage interplay between intrinsic structural information and extrinsic message information, and simulation results for the resulting performance of those schemes (RQ2.1). I have also presented extensions to these frameworks that exploit open-loop or closed-loop control during additive manufacturing for information embedding (RQ2.2).

In this final Chapter, I provide a summary of the contributions of this dissertation, and my thoughts on potentially fruitful directions for future research.

8.1 Tailored Traceability Schemes for Metallic Goods

The contributions of this dissertation in the area of tailored anti-counterfeiting and traceability schemes for metallic goods include:

- **RQ's 1.1-1.3:** A literature survey on existing work in anti-counterfeiting, including discussions on counterfeiting models and existing legal, organizational, and technical approaches for mitigating the problem (Chapter 2).
- **RQ's 1.1-1.3:** Proof-of-concept case studies for leveraging intrinsic information for tailored traceability schemes, for different material systems (Chapters 3, 4, and 5). These case studies investigate the usefulness of lineal and areal micrograph measures, as well as 2-point statistics for generating intrinsic signals of genuinity.
- **RQ's 1.2-1.3:** Formulation of anti-counterfeiting feature selection and string generation schemes as a design problem, and analysis for a large library of synthetic data (Chapter 5). This design problem can generalize to various use cases and manufacturing processes.
- **RQ's 1.2-1.3:** Validation of the above design problem using large feature libraries derived from 2-point statistics and other phase information for synthetic and experimental micrograph datasets (Chapter 5).

8.2 Embedded Traceability Schemes for Additive Manufacturing

The contributions of this dissertation in the area of tailored embedding schemes for additive manufacturing (and hopefully other manufacturing processes!) include:

- **RQ's 2.1-2.2:** A literature survey on existing work in additive manufacturing security, fuzzy extractors, and physically unclonable functions (PUFs) (Chapters 2 and 6).

- **RQ 2.1:** Analysis of possible information channels for transmitting data via information embedding (Chapter 6), for manufacturing scenarios like additive manufacturing. Such schemes could be extended to other manufacturing processes where dynamic control is possible, like some surface finishing, turning, or milling processes.

- **RQ 2.1:** Formulation of schemes for open-loop embedding and preliminary results demonstrating its utility with simulated embedding surfaces and manufacturing and use-case introduced errors (Chapter 6).

- **RQ 2.2:** A methodology for implementation of tailorable open-loop embedding in fusion deposition modeling (FDM) additive manufacturing (Chapter 6).

- **RQ 2.2:** A framework for designing closed-loop malleable PUFs for additive manufacturing (Chapters 6 and 7). This framework can generalize to AM processes where parts are built according to some time-ordering of voxels.

8.3 Future Work

I have presented this dissertation in essentially two parts: anti-counterfeiting with intrinsic information, and traceability in additive manufacturing through malleable PUFs. It is only fair that I divide my thoughts on avenues for future work along similar lines. In traceability and anti-counterfeiting, some future avenues of research include:

- *Improving schemes for traceability by searching for additional high-value features, improving damage models, and testing proposed schemes in more realistic manufacturing environments.* This dissertation has certainly not exhausted the potential information sources for tailored traceability schemes. Indeed, as I hope I have emphasized, one main outcome of this work are frameworks that will make integrating new feature sources as streamlined as possible. New fea-

tures may leverage additional material properties beyond the optical properties considered in this dissertation, such as magnetic or thermal responses.

- *Expanding the evaluation of proposed schemes to more thoroughly consider the economic feasibility of implementation throughout the product lifecycle.* This work will need to be carried out with the collaboration of industrial sponsors, who will be the decision-makers determining practical adoption of these traceability schemes. Interviews and in-situ case studies may help to identify the relevant traceability needs, wants, and constraints for these stakeholders

- *Implementing testbeds for evaluating traceability schemes in realistic environments.* Such testbeds must include proper metrology for reading the necessary intrinsic information, and methods for simulating realistic part damage. Again, long-term industrial partnerships will help here, both to ensure realistic operating conditions and to get rapid feedback regarding the feasibility of any proposed metrology or embedding mechanisms.

And in information embedding, future research topics include:

- *Expanding each component of the malleable PUF framework proposed in Chapters 6 and 7.* This should include specifying additional intrinsic signals, embedding methodologies, reading procedures, and damage profiles for different use cases, as well as validation through practical use cases and testing. These extensions may follow naturally from the additive manufacturing and cyber-physical systems literature, and from new manufacturing, quality testing, and logistics trends in relevant industries (automotive, aerospace, and so on) as they develop.

- *Integration of closed-loop malleable PUFs in additive manufacturing processes.* As discussed in Chapter 7, these malleable PUF designs require processes that lend themselves well to ordered embedding locations and spatially-varying material properties that can be used to carry information. Good process candidates

for test implementations include fused deposition modeling (FDM), selective laser sintering (SLS), and selective laser melting (SLM). Non-additive processes that treat the surface of goods in some ordered fashion may also be good candidates.

- *Investigation of deliberate fragility as a method for selectively tolerating certain kinds of damage and/or certain amounts of damage.* By tailoring malleable PUF schemes to specific transformations of the embedding surface, a manufacturer may be able to design for certain kinds of failures, while deliberately breaking the encoding for other, less tolerable failures. This requires practical investigation in realistic scenarios to validate, but is a promising method for enhancing the security of embedding schemes.

8.4 Traceability Throughout the Product Life Cycle

It's unfair to claim that manufacturing is entering a new frontier without offering significant caveats. Decades of work have built to the current environment. Decades of work in additive manufacturing, materials science, logistics, cybersecurity, and other academic fields have created and continue to improve the state of manufacturing and supply chain management.

In the final analysis, the goal of this dissertation is to add a few drops to this frankly dizzying bucket of progress. I believe that framing traceability as, first and foremost, a design task that incorporates knowledge from many different domains is a critical step along the path to truly secure supply chains. And as we continue making progress in these domains, traceability approaches that come from this perspective will continue to improve. As we discover better ways to characterize and control the material we use to make goods, as we learn more about the form those goods should take to meet the needs of diverse users, as we better understand those users and their needs over time, we can combine this information to ensure traceability, security, and safety over the product lifecycle. I hope this dissertation serves to highlight a

few ways the lessons learned in each of these areas can be used to formulate useful frameworks for traceability.

This approach will require the cooperation of stakeholders from each of these fields, each considering how their expertise fits into this puzzle. While the democratization of science in many ways appears to mirror that of manufacturing, that a topic for a different dissertation. For now, I'm content to see where the future takes these topics, and contribute where I can. It's sure to be a fun ride.

REFERENCES

REFERENCES

[1] A. W. Pense, "The decline and fall of the roman denarius," *Materials characterization*, vol. 29, no. 2, pp. 213–222, 1992.

[2] M. Brettel, N. Friederichsen, M. Keller, and M. Rosenberg, "How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective," *International Journal of Mechanical, Industrial Science and Engineering*, vol. 8, no. 1, pp. 37–44, 2014.

[3] P. Chaudhry, V. Cordell, and A. Zimmerman, "Modelling anti-counterfeiting strategies in response to protecting intellectual property rights in a global environment," *The Marketing Review*, vol. 5, no. 1, pp. 59–72, 2005.

[4] T. Staake, F. Thiesse, and E. Fleisch, "The emergence of counterfeit trade: a literature review," *European Journal of Marketing*, vol. 43, no. 3/4, pp. 320–349, 2009.

[5] M. Stevenson and J. Busby, "An exploratory analysis of counterfeiting strategies: Towards counterfeit-resilient supply chains," *International Journal of Operations & Production Management*, vol. 35, no. 1, pp. 110–144, 2015.

[6] A. Hoecht and P. Trott, "How should firms deal with counterfeiting? a review of the success conditions of anti-counterfeiting strategies," *International Journal of Emerging Markets*, vol. 9, no. 1, pp. 98–119, 2014.

[7] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: a rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014.

[8] P. Goods, "Intellectual property observations on efforts to quantify the economic effects of counterfeit and pirated goods," *Intellectual property*, 2010.

[9] W. Lucyshyn, J. S. Gansler, and J. Rigilano, "Addressing counterfeit parts in the dod supply chain," Monterey, California. Naval Postgraduate School, Tech. Rep., 2013.

[10] IACC. (2017) Counterfeiting statistics. [Online]. Available: http://www.iacc.org/resources/about/statistics

[11] R. Leal, F. Barreiros, L. Alves, F. Romeiro, J. Vasco, M. Santos, and C. Marto, "Additive manufacturing tooling for the automotive industry," *The International Journal of Advanced Manufacturing Technology*, vol. 92, no. 5-8, pp. 1671–1676, 2017.

[12] R. Liu, Z. Wang, T. Sparks, F. Liou, and J. Newkirk, "Aerospace applications of laser additive manufacturing," in *Laser additive manufacturing*. Elsevier, 2017, pp. 351–371.

[13] L. Chen, Y. He, Y. Yang, S. Niu, and H. Ren, "The research status and development trend of additive manufacturing technology," *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 9-12, pp. 3651–3660, 2017.

[14] M. Thakur and C. R. Hurburgh, "Framework for implementing traceability system in the bulk grain supply chain," *Journal of Food Engineering*, vol. 95, no. 4, pp. 617–626, 2009.

[15] J. Hu, X. Zhang, L. M. Moga, and M. Neculita, "Modeling and implementation of the vegetable supply chain traceability system," *Food Control*, vol. 30, no. 1, pp. 341–353, 2013.

[16] T. Moe, "Perspectives on traceability in food manufacture," *Trends in Food Science & Technology*, vol. 9, no. 5, pp. 211–214, 1998.

[17] M. H. Jansen-Vullers, C. A. van Dorp, and A. J. Beulens, "Managing traceability information in manufacture," *International journal of information management*, vol. 23, no. 5, pp. 395–413, 2003.

[18] M. Cheng and J. Simmons, "Traceability in manufacturing systems," *International Journal of Operations & Production Management*, 1994.

[19] A. Bechini, M. G. Cimino, F. Marcelloni, and A. Tomasi, "Patterns and technologies for enabling supply chain traceability through collaborative e-business," *Information and Software Technology*, vol. 50, no. 4, pp. 342–359, 2008.

[20] Merriam-Webster. (2017) Counterfeit. [Online]. Available: https://www.merriam-webster.com/dictionary/counterfeit

[21] B. Vlasic. (2017) Record 2016 for u.s. auto industry; long road back may be at end. [Online]. Available: https://www.nytimes.com/2017/01/04/business/2016-record-united-states-auto-sales

[22] D. Simchi-Levi, W. Schmidt, Y. Wei, P. Y. Zhang, K. Combs, Y. Ge, O. Gusikhin, M. Sanders, and D. Zhang, "Identifying risks and mitigating disruptions in the automotive supply chain," *Interfaces*, vol. 45, no. 5, pp. 375–390, 2015.

[23] T. Melton, "The benefits of lean manufacturing: what lean thinking has to offer the process industries," *Chemical engineering research and design*, vol. 83, no. 6, pp. 662–673, 2005.

[24] CNNMoney. (2007) Fake parts reportedly cost ford $1b. [Online]. Available: http://money.cnn.com/2007/01/22/news/companies/ford_counterfeit_parts

[25] J. S. Gansler and W. Lucyshyn, "The dod's use of lowest price technically acceptable (lpta) price selection," 2013.

[26] R. McCormack. (2012) Boeing's planes are riddled with chinese counterfeit electronic components. [Online]. Available: http://www.manufacturingnews.com/news/counterfeits615121.html

[27] C. Levin, J. Lieberman, J. Reed, D. Akaka, E. B. Nelson, J. Webb, C. McCaskill, M. Udall, K. Hagan, M. Begich *et al.*, "Inquiry into counterfeit electronic parts in the department of defense supply chain," *Committee on Armed Services United States Senate.[online], http://www. gpo. gov/fdsys/pkg/CRPT-112srpt167/pdf/CRPT-112srpt167. pdf*, 2012.

[28] Reuters. (2014) Aston martin recalls 17,590 cars due to counterfeit material. [Online]. Available: http://www.reuters.com/article/autos-astonmartin-recall-idUSL2N0LA0O920140205

[29] J. Dowling. (2015) Toyota trying to locate thousands of counterfeit airbag parts that could prove deadly in a crash. [Online]. Available: https://www.news.com.au/news-story/1f12a255d1093e070b1c9be616ad65ff

[30] A. Tutu. (2015) Fake takata airbag makes this honda owner's recall story worse. [Online]. Available: https://www.autoevolution.com/news/fake-takata-airbag-makes-this-honda-owner-s-recall-story-worse-99616.html

[31] K. Domdouzis, B. Kumar, and C. Anumba, "Radio-frequency identification (rfid) applications: A brief introduction," *Advanced Engineering Informatics*, vol. 21, no. 4, pp. 350–355, 2007.

[32] M. Strassner and E. Fleisch, "The promise of auto-id in the automotive industry," 2003.

[33] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th annual Design Automation Conference*. ACM, 2007, pp. 9–14.

[34] sigNatureDNA. (2017) adnas.com. [Online]. Available: http://adnas.com/signature_dna/

[35] Covisint. (2017) portal.covisint.com. [Online]. Available: https://portal.covisint.com/web/portal/autportals

[36] I. Crimes, "Title 18—crimes and criminal procedure," pp. 531–535.

[37] IACC. (2017) www.iacc.org. [Online]. Available: http://www.iacc.org/

[38] SAE. (2017) Counterfeit electrical, electronic, and electromechanical (eee) parts; avoidance, detection, mitigation, and disposition. available: saemobilus.sae.org/content/as5553b. [Online]. Available: https://saemobilus.sae.org/content/as5553b

[39] WIPO. (2017) World intellectual property organization. available: wipo.int/about-wipo/en. [Online]. Available: https://saemobilus.sae.org/content/as5553b

[40] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[41] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 523–540.

[42] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M.-D. M. Yu, "Efficient fuzzy extraction of puf-induced secrets: Theory and applications," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 412–431.

[43] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Towards Hardware-Intrinsic Security*. Springer, 2010, pp. 3–37.

[44] P. Tuyls and L. Batina, "Rfid-tags for anti-counterfeiting," in *Cryptographers' Track at the RSA Conference*. Springer, 2006, pp. 115–131.

[45] J. D. Buchanan, R. P. Cowburn, A.-V. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. A. Allwood, and M. T. Bryan, "Forgery: fingerprinting documents and packaging," *Nature*, vol. 436, no. 7050, pp. 475–475, 2005.

[46] A. Sharma, L. Subramanian, and E. A. Brewer, "Paperspeckle: microscopic fingerprinting of paper," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 99–110.

[47] Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "Emerging physical unclonable functions with nanotechnology," *IEEE access*, vol. 4, pp. 61–80, 2016.

[48] T. Takahashi and R. Ishiyama, "Fibar: Fingerprint imaging by binary angular reflection for individual identification of metal parts," in *2014 Fifth International Conference on Emerging Security Technologies (EST)*. IEEE, 2014, pp. 46–51.

[49] Y. Cao, A. J. Robson, A. Alharbi, J. Roberts, C. S. Woodhead, Y. J. Noori, R. Bernardo-Gavito, D. Shahrjerdi, U. Roedig, V. I. Fal'ko *et al.*, "Optical identification using imperfections in 2d materials," *2D Materials*, vol. 4, no. 4, p. 045021, 2017.

[50] S. P. McGrew, "Anticounterfeiting method and device utilizing holograms and pseudorandom dot patterns," Mar. 7 1995, US Patent 5,396,559.

[51] L. A. Hackel, C. B. Dane, and F. Harris, "Identification marking by means of laser peening," Jul. 23 2002, US Patent 6,423,935.

[52] E. I. Pryakhin, E. V. Larionova, and M. G. Afon'kin, "Method of marking an object to identify same," May 13 2014, US Patent 8,723,077.

[53] C. N. Chong, D. Jiang, J. Zhang, and L. Guo, "Anti-counterfeiting with a random pattern," in *2008 Second International Conference on Emerging Security Information, Systems and Technologies*, Aug 2008, pp. 146–153.

[54] P. S. Ravikanth, "Physical one-way functions," Ph.D. dissertation, MIT, 2001.

[55] C. Di Venuto, M. Kutter, and F. Jordan, "Means for using microstructure of materials surface as a unique identifier," May 15 2012, US Patent 8,180,174.

[56] J.-C. Wu, "Fingerprint identifying system using a set of microstructure layers formed on one of top and bottom faces of light-transmissive finger press plate," Jun. 19 2012, US Patent 8,204,284.

[57] S. Voloshynovskiy, M. Diephuis, T. Holotyak, and N. Standardo, "Physical object identification using micro-structure images," dOI: 10.1117/2.1201411.005524.

[58] S. K. Decker, H. L. Brunk, J. S. Carr, and G. B. Rhoads, "Watermark holograms," Aug. 24 2004, US Patent 6,782,115.

[59] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings," in *Proceedings of the 8th International Conference on Cryptographic Hardware and Embedded Systems*, ser. CHES'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 369–383. [Online]. Available: http://dx.doi.org/10.1007/11894063_29

[60] D. Jiang and C. N. Chong, "Anti-counterfeiting using phosphor puf," in *2008 2nd International Conference on Anti-counterfeiting, Security and Identification*, Aug 2008, pp. 59–62.

[61] P. Bulens, F.-X. Standaert, and J.-J. Quisquater, "How to strongly link data and its medium: the paper case," *IET Information Security*, vol. 4, no. 3, p. 125, 2010. [Online]. Available: http://dx.doi.org/10.1049/iet-ifs.2009.0032

[62] J. D. R. Buchanan, R. P. Cowburn, A.-V. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. A. Allwood, and M. T. Bryan, "Forgery: 'fingerprinting' documents and packaging," *Nature*, vol. 436, no. 7050, pp. 475–475, jul 2005. [Online]. Available: http://dx.doi.org/10.1038/436475a

[63] U. Park, S. Pankanti, and A. K. Jain, "Fingerprint verification using sift features," in *Biometric Technology for Human Identification V*, vol. 6944. International Society for Optics and Photonics, 2008, p. 69440K.

[64] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," 2011.

[65] M. Muja and D. G. Lowe, "Fast matching of binary features," in *2012 Ninth Conference on Computer and Robot Vision*. IEEE, 2012, pp. 404–410.

[66] S. R. Niezgoda, Y. C. Yabansu, and S. R. Kalidindi, "Understanding and visualizing microstructure and microstructure variance as a stochastic process," , *Acta Materialia*, vol. 59, no. 16, pp. 6387 – 6400, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359645411004654

[67] A. Dachowicz, S. C. Chaduvula, M. Atallah, and J. H. Panchal, "Microstructure-based counterfeit detection in metal part manufacturing," *JOM*, vol. 69, no. 11, pp. 2390–2396, 2017.

[68] A. Dachowicz, M. Atallah, and J. H. Panchal, "Optical puf design for anti-counterfeiting in manufacturing of metallic goods," in *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2018, pp. V01BT02A035–V01BT02A035.

[69] C. A. Schuh, M. Kumar, and W. E. King, "Analysis of grain boundary networks and their evolution during grain boundary engineering," *Acta Materialia*, vol. 51, no. 3, pp. 687–700, 2003.

[70] D. T. Fullwood, S. R. Niezgoda, and S. R. Kalidindi, "Microstructure reconstructions from 2-point statistics using phase-recovery algorithms," *Acta Materialia*, vol. 56, no. 5, pp. 942–948, 2008.

[71] S. E. Zeltmann, N. Gupta, N. G. Tsoutsos, M. Maniatakos, J. Rajendran, and R. Karri, "Manufacturing and security challenges in 3D printing," *JOM*, vol. 68, no. 7, pp. 1872–1881, 2016.

[72] M. Yampolskiy, W. E. King, J. Gatlin, S. Belikovetsky, A. Brown, A. Skjellum, and Y. Elovici, "Security of additive manufacturing: Attack taxonomy and survey," *Additive Manufacturing*, vol. 21, pp. 431–457, 2018.

[73] J. H. Panchal, S. R. Kalidindi, and D. L. McDowell, "Key computational modeling issues in integrated computational materials engineering," *Computer-Aided Design*, vol. 45, no. 1, pp. 4–25, Jan. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.cad.2012.06.006

[74] B. L. Adams, S. R. Kalidindi, and D. T. Fullwood, "Chapter 4 - description of the microstructure," in *Microstructure Sensitive Design for Performance Optimization*, B. L. Adams, S. R. Kalidindi, and D. T. Fullwood, Eds. Boston: Butterworth-Heinemann, 2013, pp. 67 – 87. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123969897000046

[75] W.-H. Tsai, "Moment-preserving thresholding: A new approach," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 377–393, 1985.

[76] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid *et al.*, "Fiji: an open-source platform for biological-image analysis," *Nature methods*, vol. 9, no. 7, pp. 676–682, 2012.

[77] S. Ghosh and D. M. Dimiduk, *Computational methods for microstructure-property relationships*. Springer, 2011.

[78] R. T. DeHoff and F. N. Rhines, *Quantitative Microscopy*. New York, Mcgraw-Hill, 1968.

[79] G. Pellissier and S. Purdy, "Stereology and quantitative metallography." American Society for Testing & Materials, 1972.

[80] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[81] A. Dachowicz, M. Atallah, and J. H. Panchal, "Extraction and analysis of spatial correlation micrograph features for traceability in manufacturing," in *ASME 2019 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2019)*, ASME. Anaheim, CA: ASME, August 2019.

[82] A. Dachowicz, M. Atallah, and J. Panchal, "Extraction and analysis of spatial correlation micrograph features for traceability in manufacturing," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 5, 2020.

[83] D. Wheeler, D. Brough, T. Fast, S. Kalidindi, and A. Reid, "Pymks: Materials knowledge system in python," 2014, doi:10.6084/m9.figshare.1015761.

[84] S. Huber, "libstick. software package for persistent homology calculation." *https://www.sthu.org/code/libstick/*, 2019.

[85] S. Niezgoda, D. Fullwood, and S. Kalidindi, "Delineation of the space of 2-point correlations in a composite material system," *Acta Materialia*, vol. 56, no. 18, pp. 5285–5292, 2008.

[86] A. Cecen, T. Fast, and S. R. Kalidindi, "Versatile algorithms for the computation of 2-point spatial correlations in quantifying material structure," *Integrating Materials and Manufacturing Innovation*, vol. 5, no. 1, p. 1, 2016.

[87] S. M. Qidwai, D. M. Turner, S. R. Niezgoda, A. C. Lewis, A. B. Geltmacher, D. J. Rowenhorst, and S. R. Kalidindi, "Estimating the response of polycrystalline materials using sets of weighted statistical volume elements," *Acta Materialia*, vol. 60, no. 13-14, pp. 5284–5299, 2012.

[88] N. H. Paulson, M. W. Priddy, D. L. McDowell, and S. R. Kalidindi, "Reduced-order structure-property linkages for polycrystalline microstructures based on 2-point statistics," *Acta Materialia*, vol. 129, pp. 428–438, 2017.

[89] T. Mueller, A. G. Kusne, and R. Ramprasad, "Machine learning in materials science: Recent progress and emerging applications," *Reviews in Computational Chemistry*, vol. 29, pp. 186–273, 2016.

[90] S. R. Kalidindi, S. R. Niezgoda, G. Landi, S. Vachhani, and T. Fast, "A novel framework for building materials knowledge systems," *Computers, Materials, & Continua*, vol. 17, no. 2, pp. 103–125, 2010.

[91] M. I. Latypov and S. R. Kalidindi, "Data-driven reduced order models for effective yield strength and partitioning of strain in multiphase materials," *Journal of Computational Physics*, vol. 346, pp. 242–261, 2017.

[92] S. R. Kalidindi, A. J. Medford, and D. L. McDowell, "Vision for data and informatics in the future materials innovation ecosystem," *JOM*, vol. 68, no. 8, pp. 2126–2137, 2016.

[93] R. Ghrist, "Barcodes: the persistent topology of data," *Bulletin of the American Mathematical Society*, vol. 45, no. 1, pp. 61–75, 2008.

[94] G. Carlsson, "Topology and data," *Bulletin of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.

[95] H. Edelsbrunner and J. Harer, *Computational topology: an introduction*. American Mathematical Soc., 2010.

[96] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[97] A. Unler, A. Murat, and R. B. Chinnam, "mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification," *Information Sciences*, vol. 181, no. 20, pp. 4625–4641, 2011.

[98] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, p. 066138, Jun 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.69.066138

[99] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[100] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[101] S. Grantham, B. Lane, J. Neira, S. Mekhontsev, M. Vlasea, and L. Hanssen, "Optical design and initial results from nist's ammt/temps facility," in *Laser 3D Manufacturing III*, vol. 9738. International Society for Optics and Photonics, 2016, p. 97380S.

[102] H. Yeung, B. Lane, and J. Fox, "Part geometry and conduction-based laser power control for powder bed fusion additive manufacturing," *Additive Manufacturing*, vol. 30, p. 100844, 2019.

[103] J. C. Fox, B. M. Lane, and H. Yeung, "Measurement of process dynamics through coaxially aligned high speed near-infrared imaging in laser powder bed fusion additive manufacturing," in *Thermosense: Thermal Infrared Applications XXXIX*, vol. 10214. International Society for Optics and Photonics, 2017, p. 1021407.

[104] J. A. Slotwinski and E. J. Garboczi, "Metrology needs for metal additive manufacturing powders," *JOM*, vol. 67, no. 3, pp. 538–543, 2015.

[105] A. T. Sutton, C. S. Kriewall, M. C. Leu, and J. W. Newkirk, "Powder characterisation techniques and effects of powder characteristics on part properties in powder-bed fusion processes," *Virtual and physical prototyping*, vol. 12, no. 1, pp. 3–29, 2017.

[106] D. Mies, W. Marsden, and S. Warde, "Overview of additive manufacturing informatics:"a digital thread"," *Integrating Materials and Manufacturing Innovation*, vol. 5, no. 1, pp. 114–142, 2016.

[107] L. Thames and D. Schaefer, *Cybersecurity for industry 4.0*. Springer, 2017.

[108] A. Padmanabhan and J. Zhang, "Cybersecurity risks and mitigation strategies in additive manufacturing," *Progress in Additive Manufacturing*, vol. 3, no. 1-2, pp. 87–93, 2018.

[109] K. Wolter and P. Reinecke, "Performance and security tradeoff," in *International School on Formal Methods for the Design of Computer, Communication and Software Systems*. Springer, 2010, pp. 135–167.

[110] R. Candell, K. Stouffer, and D. Anand, "A cybersecurity testbed for industrial control systems," in *Proceedings of the 2014 Process Control and Safety Symposium*, 2014, pp. 1–16.

[111] K. Stouffer and R. Candell, "Measuring impact of cybersecurity on the performance of industrial control systems," *Mechanical Engineering*, vol. 136, no. 12, pp. S4–S7, 2014.

[112] H. T. Maia, D. Li, Y. Yang, and C. Zheng, "LayerCode: optical barcodes for 3D printed shapes," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[113] A. Delmotte, K. Tanaka, H. Kubo, T. Funatomi, and Y. Mukaigawa, "Blind watermarking for 3D printed objects by locally modifying layer thickness," *IEEE Transactions on Multimedia*, pp. 1–1, 2019. [Online]. Available: https://doi.org/10.1109/tmm.2019.2962306

[114] C. Wei, Z. Sun, Y. Huang, and L. Li, "Embedding anti-counterfeiting features in metallic components via multiple material additive manufacturing," *Additive Manufacturing*, vol. 24, pp. 1–12, Dec. 2018. [Online]. Available: https://doi.org/10.1016/j.addma.2018.09.003

[115] Z. C. Kennedy, D. E. Stephenson, J. F. Christ, T. R. Pope, B. W. Arey, C. A. Barrett, and M. G. Warner, "Enhanced anti-counterfeiting measures for additive manufacturing: coupling lanthanide nanomaterial chemical signatures with blockchain technology," *Journal of Materials Chemistry C*, vol. 5, no. 37, pp. 9570–9578, 2017. [Online]. Available: https://doi.org/10.1039/c7tc03348f

[116] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9-12, pp. 3563–3576, 2018.

[117] D. G. Aliaga and M. J. Atallah, "Genuinity signatures: Designing signatures for verifying 3D object genuinity," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 437–446.

[118] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan Kaufmann, 2007.

[119] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, 2019, pp. 241–264.

[120] S. Gupta, A. Goyal, and B. Bhushan, "Information hiding using least significant bit steganography and cryptography," *International Journal of Modern Education and Computer Science*, vol. 4, no. 6, p. 27, 2012.

[121] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977, vol. 16.

[122] C. E. Shannon, "Certain results in coding theory for noisy channels," *Information and control*, vol. 1, no. 1, pp. 6–25, 1957.

[123] F. Gray, "Pulse code communication," Mar. 17 1953, US Patent 2,632,058.

[124] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.

[125] G. Forney, "Convolutional codes I: Algebraic structure," *IEEE Transactions on Information Theory*, vol. 16, no. 6, pp. 720–738, 1970.

[126] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE transactions on communications*, vol. 36, no. 4, pp. 389–400, 1988.

[127] A. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Transactions on Communication Technology*, vol. 19, no. 5, pp. 751–772, 1971.

[128] A. Lapidoth, "The performance of convolutional codes on the block erasure channel using various finite interleaving techniques," *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1459–1473, 1994.

[129] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *TDAPR*, vol. 126, pp. 1–26, 1996.

[130] D. Du, F. K. Hwang, and F. Hwang, *Combinatorial group testing and its applications*. World Scientific, 2000, vol. 12.

[131] M. Aldridge, L. Baldassini, and O. Johnson, "Group testing algorithms: Bounds and simulations," *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3671–3687, 2014.

[132] A. Ranellucci, "Reprap/slic3r and the future of 3D printing," *Canessa, E., Fonda, C., and Zennaro, ed.,"Low-cost 3D Printing for Science, Education, and Sustainable Development*, pp. 75–82, 2013.

VITA

VITA

ADAM P. DACHOWICZ

School of Mechanical Engineering
585 Purdue Mall, West Lafayette, IN 47907

Email: adachowi@purdue.edu
linkedin.com/in/adam-dachowicz/

EDUCATION

Ph.D. in Mechanical Engineering — Expected August 2020
Focus Area: Secure Design and Manufacturing — West Lafayette, IN
Purdue University

BS in Mechanical Engineering — May 2015
The University of Oklahoma — Norman, OK

RESEARCH INTERESTS

My research interests include topics in hardware-intrinsic traceability and security in manufacturing, addressed primarily through image modeling and processing, cryptography, application of physically unclonable functions, and the study of information exchange in collaborative design. The research presented in my dissertation focuses on the design and analysis of tailorable, hardware-intrinsic traceability schemes for anti-counterfeiting, and methods for securely embedding information in additive manufacturing.

PUBLICATIONS

PhD Dissertation

Dachowicz, A., "Tailored Traceability and Provenance Determination in Manufacturing." Presented to the faculty of the School of Mechanical Engineering, August 2020, Purdue University, West Lafayette, Indiana, USA.

Advisors: Dr. Jitesh H. Panchal, Dr. Mikhail Atallah.

Committee Members: Dr. Karthik Ramani, Dr. Ilias Bilionis.

Journal Articles

J1 **Dachowicz, A.**, Atallah, M., Panchal, J. H. 2020. "Extraction and Analysis of Spatial Correlation Micrograph Features for Traceability in Manufacturing." Journal of Computing and Information Science in Engineering, 20(5). DOI: 10.1115/1.4046891

J2 Chaduvula, S.C., **Dachowicz, A.**, Atallah, M., and Panchal, J. H. 2018. "Security in Cyber-Enabled Design and Manufacturing: A Survey." J. Comput. Inf. Sci. Eng, Vol. 18, No. 4, pp. 040802. DOI: 10.1115/1.4040341.

J3 **Dachowicz, A.**, Chaduvula, S. C., Atallah, M. J., Bilionis, I., Panchal, J. H. 2018. "Strategic information revelation in collaborative design." Advanced Engineering Informatics, Vol. 36, pp. 242-253. DOI: 10.1016/j.aei.2018.04.004.

J4 **Dachowicz, A.**, Chaduvula, S. C., Atallah, M., and Panchal, J. H. 2017. "Microstructure-Based Counterfeit Detection in Metal Part Manufacturing." JOM, Vol. 69, No. 11, pp. 2390-2396. DOI: 10.1007/s11837-017-2502-8.

Refereed Conference Papers

C1 **Dachowicz, A.**, Atallah, M, and Panchal, J. H., 2019. "Extraction and Analysis of Spatial Correlation Micrograph Features for Traceability in Manufacturing." ASME 2019 International Design Engineering Technical Conferences

and Computers and Information in Engineering Conference. Paper Number: DETC2019-98378.

C2 **Dachowicz, A.**, Atallah, M, and Panchal, J. H., 2018. "Optical PUF Design for Anti-Counterfeiting in Manufacturing of Metallic Goods." ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Paper Number: DETC2018-85714.

C3 **Dachowicz, A.**, Chaduvula, S. C., Panchal, J. H., and Atallah, M., 2016. "Confidentiality Management in Collaborative Design." ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Paper Number: DETC2016-59222.

C4 Goh, C. H., **Dachowicz, A.**, Allen, J. K., and Mistree, F. 2015. "Exploring the Performance-Property-Structure Solution Space in Friction Stir Welding." 3rd World Congress on Integrated Computational Materials Engineering (ICME 2015). DOI: 10.1007/978-3-319-48170-8_41.

C5 Goh, C. H., Ahmed, S., **Dachowicz, A.**, Allen, J. K., and Mistree, F. 2014. "Integrated Multiscale Robust Design Considering Microstructure Evolution and Material Properties in the Hot Rolling Process." ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Paper Number: DETC2014-34157.

RESEARCH EXPERIENCE

Graduate Research Assistant, August 2015 – August 2020.

Design Engineering Lab at Purdue (DELP), Purdue University School of Mechanical Engineering, West Lafayette, IN.

Advisors: Dr. Jitesh H. Panchal, Dr. Mikhail Atallah.

Research Activities:

- Designed and applied optical physically unclonable function (PUF) schemes for counterfeit prevention and traceability in manufacturing, and proposed a framework for scheme design and evaluation.
- Designed cryptographic schemes for embedding information in 3d-printed parts.
- Analyzed methods for studying the performance of cognitive agents in space systems.
- Implemented convolutional neural network models for studying game balance in real-time strategy games.
- Implemented a method for quantifying value and risk of revealing information in collaborative design.
- Contributed to grant proposal writing in the areas of hardware-intrinsic anti-counterfeiting and security in additive manufacturing.

Undergraduate Research Fellow, May 2014 - August 2014.

National Institute of Standards and Technology (NIST) Information Technology Laboratory, Gaithersburg, MD.

Advisor: Dr. Alden Dima.

Research Activities:

- Implemented natural language processing (NLP) techniques for keyword extraction and analysis from a corpus of scientific literature.
- Contributed to building a materials science ontology for use by researchers involved in the Materials Genome Initiative.

Undergraduate Research Assistant, January 2013 – May 2015.

Systems Realization Laboratory, The University of Oklahoma, Norman, OK.

Advisors: Dr. Farrokh Mistree, Dr. Janet K. Allen.

Research Activities:

- Implemented finite element models of shot peening and hot rolling processes.

- Assisted lab mates with data collection, analysis, and manuscript preparation.

TEACHING EXPERIENCE

Gifted Education Resource Institute (GERI) Summer Instructor, July 2018.

GERI, Purdue University, West Lafayette, IN.

Course: Rise of the Internet of Things.

Advisor: Dr. Jitesh H. Panchal.

Activities:

- Developed and ran a 2-week, intensive course for high school students on the Internet of Things and cyber-physical systems in general.
- Instructed 3-hour class sessions over the two weeks of the program.

Undergraduate Student Mentor, August 2013 – May 2015.

College of Engineering, The University of Oklahoma, Norman, OK.

Supervisor: Dr. John Antonio.

Activities:

- Organized bi-weekly classes for groups of 20 first-year engineering students with the goal of assisting their transition to college and increasing their understanding of various engineering disciplines.

Teaching Assistant, August 2014 – December 2014.

College of Engineering, The University of Oklahoma, Norman, OK.

Course: Freshman Engineering Orientation.

Supervisor: Dr. John Antonio.

Activities:

- Conducted weekly office hours, graded homework assignments, and answered student questions via email and during class.
- Assisted with faculty-led lectures.

PROFESSIONAL EXPERIENCE

Co-Founder and Chief Operating Officer, November 2018 – Present.

RightFit Analytics, Inc. West Lafayette, IN.

Activities:

- Co-founded a healthcare data analytics start-up with a Purdue graduate and three Purdue faculty members aiming to provide evidence-driven patient-provider matching technology.
- Contributed to the design and implementation of data analysis and machine learning pipelines for recommending patient-provider matches for elective healthcare procedures.
- Contributed to fund-raising and grant writing activities, including a successful NSF SBIR Phase I grant proposal (Award 1938405, project period May 2020-October 2020).
- Provided operational (financial and employment) management for the company.

AWARDS

A1 Finalist, Burton D. Morgan Business Model Competition 2019, awarded by the Burton D. Morgan Center for Entrepreneurship, Purdue University.

A2 Purdue Doctoral Fellowship Recipient, 2015-2017, awarded by the Purdue University Graduate School.

A3 Outstanding Senior in Mechanical Engineering, 2015, awarded by the University of Oklahoma.

A4 IDETC Student Design Essay and Travel Grant Winner, 2013 and 2014. Awarded by the ASME.

OTHER PRESENTATIONS AND POSTERS

P1 **Dachowicz, A.**, November 2019. "Customized Traceability in Modern Supply Chains." Presented as an invited seminar speaker at the National Institute of Standards and Technology (NIST).

P2 **Dachowicz, A.**, January 2018. "Hardware-Intrinsic Security Schemes for Counterfeit Detection." Presented at the Graduate Student Seminar Series, Purdue University School of Mechanical Engineering.

P3 **Dachowicz, A.**, August 2014. "Looking to the Future of Design and Manufacturing: To 2030 and Beyond." Poster presented at the 2014 ASME IDETC/COE Conference, Buffalo, NY.

P4 **Dachowicz, A.**, August 2014. "A Comparative Assessment of the LPV Algorithm for Keyword Extraction." Presented at the 2014 NIST SURF Research Symposium, Gaithersburg, MD.