Hardware Security against IP Piracy Using Forensic Biometrics

A THESIS

Submitted in partial fulfillment of the
requirements for the award of the degree

of

Master of Science (Research)

by

Ayush Kumar Singh



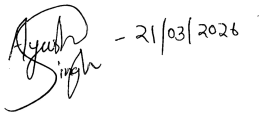
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
APRIL 2026

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Hardware Security against IP Piracy Using Forensic Biometrics** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF SCIENCE (RESEARCH)** and submitted in the **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**, Indian Institute of Technology Indore, is an authentic record of my own work carried out during the time period from July 2024 to May 2026 under the supervision of **Prof. Anirban Sengupta**.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.



Signature of the student with date

(Ayush Kumar Singh)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.



March 23, 2026

Signature of the Supervisor of

M.S. (Research) thesis with date

(Prof. Anirban Sengupta)

Ayush Kumar Singh has successfully given his/her M.S. (Research) Oral Examination held on June 5, 2026.



June 5, 2026

Signature of Supervisor of M.S. (Research) thesis

ACKNOWLEDGEMENTS

Completing this thesis has been a challenging yet immensely rewarding journey one that would not have been possible without the constant support, guidance, and encouragement of numerous individuals. I take this opportunity to express my heartfelt gratitude to everyone who played a part in this endeavour.

Firstly, I would like to express my deepest gratitude to my thesis advisor, **Prof. Anirban Sengupta**, for his exceptional mentorship, insightful guidance, and unwavering support throughout my research journey. His constructive feedback and constant encouragement were invaluable in shaping the direction and depth of my research work. I am truly grateful for the academic freedom, opportunities, and challenges he provided, which helped me grow both personally and professionally.

I would also like to extend my sincere thanks to my lab seniors **Dr. Aditya Anshul, Mr. Vishal Chaurasia, and Mr. Nabendu Bhui** for their consistent support, valuable advice, and encouragement during the course of my work. Their technical expertise, critical insights, and friendly guidance served as a strong foundation and constant motivation for me throughout my research journey.

I am thankful to my thesis committee members for taking the time to review my work and provide valuable feedback and suggestions that greatly improved the quality of this thesis. Their diverse perspectives and constructive evaluations helped me better understand the broader scope of my research.

I would like to acknowledge the entire **faculty and staff of the Computer Science and Engineering Department at IIT Indore** for creating a positive, resourceful, and intellectually stimulating environment. The academic infrastructure, learning

opportunities, and overall support provided by the department were crucial in facilitating my research work.

I am deeply grateful to my **family and friends** for their love, patience, and encouragement during all phases of this journey. Their emotional support and belief in my capabilities have been the cornerstone of my strength. I would especially like to thank my **parents and siblings** for standing by me during difficult times and always motivating me to push forward.

A special note of thanks to my **batchmates**, who have been a constant source of encouragement, camaraderie, and support. The shared experiences, discussions, and collaborative spirit have made this journey both enjoyable and fulfilling.

Lastly, I extend my gratitude to all my **peers, colleagues, and everyone** who has supported me in various ways be it through academic discussions, moral support, or simply being there during crucial times. Each contribution, no matter how small, has played a meaningful role in helping me reach this milestone.

This thesis stands as a testament to the collective efforts, support, and belief of all the individuals mentioned above. Thank you all for being part of my journey.

ABSTRACT

This thesis proposes three different practical methodologies aimed at strengthening the **hardware security** of intellectual property (IP) core designs, especially those generated using **high-level synthesis (HLS)** process. As hardware attacks and IP piracy become increasingly sophisticated, these contributions offer effective solutions to detect and prevent such threats, ensuring trustworthy integration of IPs into modern **consumer electronics (CE)** and **system-on-chip (SoC)** designs.

The first contribution of this presents a novel secure hardware intellectual property (IP) protection framework that integrates biometric-driven digital signatures into the high-level synthesis (HLS) design flow. The proposed approach generates a robust encrypted fused signature derived from the IP vendor's biological identity, including amino acid biometric samples and facial anthropometric features. To enhance security, the fused signature is encrypted using a secure non-invertible fingerprint-based key within an AES cryptographic framework. The encrypted signature is subsequently transformed into covert hardware security constraints that are embedded during the register allocation phase of the HLS process. By integrating biometric-derived signatures with design automation, the methodology establishes strong ownership traceability and enables reliable detection of IP piracy and false ownership claims. Experimental evaluation on image processing filter IPs demonstrates that the proposed approach generates a large set of covert constraints, providing high tamper tolerance, extremely low probability of coincidence, and strong entropy while maintaining negligible design overhead. The results indicate that incorporating multi-biometric encrypted signatures within the HLS workflow offers an effective and scalable solution for securing hardware IP cores against piracy and unauthorized reuse.

The second contribution of this thesis introduces Gen-Sign, a secure hardware watermarking framework that embeds an encrypted genomic signature of the IP vendor into hardware IP cores during the high-level synthesis process. The

methodology extracts genomic base-pair sequences from the vendor's biological sample and converts them into a binary genomic signature through an encoding scheme. The generated signature is then encrypted using a multi-round Feistel cipher network with vendor-specified secret keys to produce a highly secure encrypted genomic watermark. The resulting signature is converted into hardware security constraints and embedded within the register allocation phase of HLS, enabling imperceptible watermarking of compute-intensive IP cores such as convolutional neural network (CNN) layers and image processing filters. The approach strengthens ownership authentication and enhances resistance against common hardware threats including tampering, removal, and ghost signature search attacks. Experimental results show that the proposed technique achieves strong security properties, including low probability of coincidence, high tamper tolerance, and improved entropy, while incurring zero design overhead in performance and resource utilization. These results demonstrate that genomic-based watermarking integrated within the HLS workflow provides an efficient and reliable mechanism for protecting modern hardware IPs against piracy and unauthorized replication.

LIST OF PUBLICATIONS

JOURNAL:

- A. Sengupta, A. Anshul, A. K. Singh "Hardware Security against IP Piracy using Secure Fingerprint Encrypted Fused Amino-Acid Biometric with Facial Anthropometric Signature", **Elsevier Microprocessors and Microsystems**, Vol 112, 2025, 105131.

CONFERENCES:

- A. Sengupta, V. Chourasia and A. K. Singh, "Gen-Sign: HLS Based Watermarking Using IP Vendor's Feistel Cipher Encrypted Genomic Signature for Protecting CNN and Image Processing Filter Cores Against Piracy," **10th IEEE International Symposium on Smart Electronic Systems (iSES)**, New Delhi, India, 2024, pp. 128-133.

TABLE OF CONTENTS

LIST OF FIGURES . . . . . .……..11

LIST OF TABLES . . . . . .……..12

Chapter 1: Introduction

1.1 Architectural Foundations of Hardware Systems and IP Integration ……13

1.2 Hierarchical Design Abstractions in ASIC Development. . . . . …….. 14

1.3 Transformation of Integrated Circuit Design in the Era of Reusable Hardware. 15

1.4 Role of High-Level Design Abstractions in Modern Hardware Development 16

1.5 Security Risks in Contemporary IP-Centric Design Ecosystems. . . . ……17

1.6 Secure High-Level Design Methodology for Protected IP Generation. ……19

1.7 Organisation of the Thesis . . . . . .……...20

Chapter 2: Literature Review

2.1 Review of Past Work and Problem Formulation. . . . . .……..22

2.2 Literature Review on IP Watermarking for CNN IP core. . . . ……....23

2.3 Biometric and Genomic Encrypted Signatures for Hardware IP Protection. . . . . …...24

2.4 Different State-of-the-Art IP Security Mechanisms. . . . . .……...25

1.5 Summary . . . . . .……..26

Chapter 3: Biometric-Fusion Driven Hardware IP Protection Using Encrypted Fingerprint and Molecular Identity Signatures

3.1 Introduction. . . . .……27

3.1.1 Background and Context……..27

3.1.2 Hardware IP Piracy: Challenges and Risks.……28

3.1.3 Limitations of Conventional Protection Techniques. ……..28

3.1.4 Motivation for Biometric-Based Security.……29

3.1.5 Problem Statement.……...30

3.1.6 Proposed Approach Overview. ……30

3.2 Proposed Security Methodology . . . . . .……...31

3.2.1 Amino Acid Signature Generation (Module 1). ……..31

3.2.2 Facial Anthropometric Signature Generation (Module 2). ……..33

3.2.3 Secure Non-Invertible Fingerprint Processing (Module 3). ……35

3.2.4 Generation of Encrypted Fused Signature Using AES. ……...37

3.2.5 Embedding of Covert Encrypted Signature in Hardware IP (Module 4). ……38

3.3 Detection of IP Piracy and Ownership Validation. . . . . .……41
3.4 Summary . . . .……41

Chapter 4: Gen-Sign: HLS based Watermarking using IP Vendor's Feistel Cipher Encrypted Genomic Signature for Protecting CNN and Image Processing Filter Cores against Piracy

4.1 Introduction. . . . . .……43
4.2 Proposed hardware watermarking: gen-sign . . . . .……..43
4.2.1 Overview of Gen-Sign. ……43
4.3 Details of the proposed encrypted genomic signature: gen-sign. . . . . .……..45
4.3.1 Extraction of Genomic Sequence and Creation of Genomic Signature. ……..45
4.3.2 Generation of Encrypted Genomic Signature.……..46
4.4 Embedding of Encrypted Genomic Signature on CNN Convolutional Layer and Sharpening Filter. . . ..……....48
4.5 Verification and Validation. .. . .…….....52
4.6 Summary. . . . . .……..…52

Chapter 5: Results and Analysis

5.1 Evaluation Metrics. . . ……54
5.1.1 Probability of Coincidence (IC). ……....54
5.1.2 Tamper Tolerance (TT). ……..55
5.1.3 Entropy (EN). ……...56
5.1.4 Design Overhead. ……57
5.2 Results and analysis of paper-1 biometric–aes fused signature method . . . ……...57
5.2.1 Probability of Coincidence Analysis. ……..…57
5.2.2 Tamper Tolerance Analysis. ……....58
5.2.3 Entropy Analysis. ……60
5.2.4 Design Overhead Analysis. ……..61
5.2.5 Summary of Paper-1 Results. ……..62
5.3 Results and Analysis: Gen-Sign Feistel Encrypted Genomic Method **……**……62
5.3.1 - Probability of Coincidence Analysis. ……..62
5.3.2 - Tamper Tolerance Analysis. ……...64
5.3.3 Entropy Analysis. ……65

5.3.4 Design Overhead Analysis. ……..66

Chapter 6: Conclusion and Future Scope

6.1 Conclusion and Future Scope. …….……...…….67

REFERENCES ……..……..69

LIST OF FIGURES

Fig. 1.1 Abstraction Levels in IP Core Design.. ……14

Fig. 3.1.Generation of the Amino Acid Signature. ……..…….32

Fig. 3.2 Facial biometric feature extraction and binary signature generation process. ……34

Fig. 3.3 Minutiae Point in the Fingerprint Image. ……..35

Fig. 3.4 Fingerprint-based encryption key generation using multi-stage translation of minutiae points. …….36

Fig. 3.5 Scheduled data flow graph (SDFG) of the sharpening image filter showing operation scheduling using four adders and two multipliers. ……..……..39

Fig. 4.1 Overview of the proposed Gen-Sign methodology for HLS-based hardware IP watermarking and secure datapath synthesis. ……44

Fig. 4.2 Genomic sequence with base pairs and polynucleotide. ……..46

Fig. 4.3 Feistel cipher integrated in proposed Gen-Sign. ……..47

Fig 4.4 Scheduled data flow graph of loop unrolled CNN using 6 Multiplier and 2 Adder ……49

Fig. 4.5 Scheduled data flow graph for sharpening filter using 2M and 4A.……..50

LIST OF TABLES

Table 3.1. Minutiae points coordinate post-Translation 2 (T2), along with type, ridge angle, and their corresponding concatenated binary equivalents. …….38

Table 3.2. Register (Reg) Allocation Table: Pre- and Post-Embedding Secret Security Constraints Corresponding to the Sharpening Filter. ……38

Table 4.1 Encoded hardware security constraints mapped onto register pairs for watermark embedding during register allocation. ……51

Table 4.2 Comparison of the register allocation table (RAT) for the CNN convolution layer before and after embedding hardware security constraints. …….51

Table 5.1. Comparison of the Probability of Coincidence (IC) in the Proposed Approach……58

Table 5.2. Comparison of Tamper Tolerance (LO) in the Proposed Approach. ……..59

Table 5.3. Comparison of Tamper Tolerance (LO) in the Proposed Approach. ……..59

Table 5.4. Comparison of Entropy in the Proposed Approach. ……60

Table 5.5. Design Cost, Area, and Latency of the Proposed Approach Before and After Embedding Security Constraints. ……..61

Table 5.6. Comparison of the Probability of Coincidence with Previous Works. ……63

Table 5.7. Comparison of Tamper Tolerance with Previous Works. ……...64

Table 5.8. Comparison of Entropy with Previous Works. ……65

Table 5.9. Overhead Analysis of the Proposed Gen-Sign Methodology. …….66

Chapter 1

1. Introduction

1.1 Foundations of Application-Specific Hardware Systems and IP Core Design

The rapid evolution of modern technology has significantly transformed the way electronic systems are designed and deployed. Continuous advancements in computing and semiconductor technologies have enabled the development of intelligent, high-performance systems that are deeply integrated into everyday life. Today's digital ecosystem relies heavily on electronic platforms that deliver fast, reliable, and energy-efficient computation across a wide range of applications. Devices such as smartphones, wearable electronics, digital cameras, personal computers, and wireless communication tools have become essential components of modern living, supporting both consumer convenience and industrial functionality. Beyond consumer electronics, computing systems also play a vital role in domains such as healthcare, defense, and industrial automation, where accuracy, reliability, and efficiency are of paramount importance. For instance, healthcare technologies enable real-time patient monitoring and advanced diagnostic capabilities, while military systems depend on secure and high-performance computing for surveillance, communication, and strategic decision-making [1]-[4].

With the growing demand for performance optimization and energy efficiency, there has been a significant shift toward application-oriented computing paradigms. Unlike general-purpose processors, application-specific computing platforms are designed to execute dedicated tasks with greater speed, lower power consumption, and improved operational efficiency. These systems are particularly effective for computation-intensive applications such as image processing, multimedia analytics, cryptography, and machine learning. As a result, many modern computing solutions are implemented as Application-Specific Integrated Circuits (ASICs), often referred to as hardware accelerators, which are tailored to meet the performance requirements of targeted applications [12], [14], [19].

At the core of these hardware accelerators lies the System-on-Chip (SoC) architecture, which integrates multiple functional components—including processing elements, memory units, controllers, and peripheral interfaces—onto a single chip. Rather than designing every component from scratch, system designers commonly rely on pre-designed and reusable IP cores obtained from third-party vendors. This design paradigm, commonly known as IP-based design, has transformed the semiconductor industry by enabling faster development cycles, reduced engineering effort, and improved design reliability [13], [16], [18].

The development of modern SoCs is typically distributed across multiple organizations and geographic locations. Design activities such as IP development, system integration, verification, and fabrication are often carried out by specialized entities with domain-specific expertise. This globally distributed workflow is driven by factors such as access to specialized skills, cost optimization, and shortened time-to-market requirements. While this collaborative ecosystem accelerates innovation, it also introduces new challenges related to design complexity, trust, and security. Consequently, ensuring reliability, protection of intellectual property, and secure integration has become a critical concern throughout the hardware development lifecycle [5]-[11].

1.2 Hierarchical Design Abstractions in ASIC Development

To manage the growing complexity of modern hardware systems, designers rely on abstraction-based design methodologies. Working at higher abstraction levels simplifies system modeling, improves productivity, and enables early integration of performance and security considerations. In the ASIC design flow, abstraction typically progresses through multiple levels, including system-level, behavioral or algorithmic level, register-transfer level (RTL), gate level, and physical layout level. Each level provides a different view of the system, allowing designers to gradually refine functionality while managing complexity [14], [18].

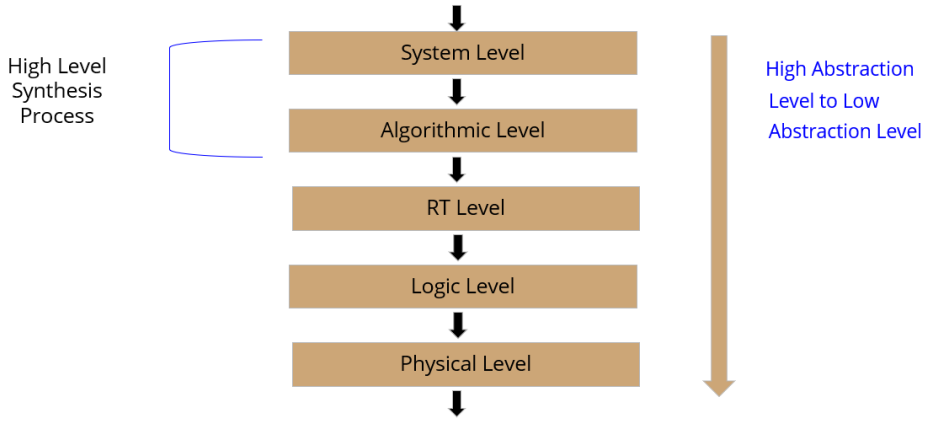


Fig. 1.1 Abstraction Levels in IP Core Design.

Figure 1.1 illustrates the different abstraction levels involved in IP core design. It shows how a hardware system is developed progressively from higher conceptual levels to lower physical implementation levels. At the top level, the system or algorithm level describes the overall functionality and behavior of the design. This is followed by the behavioral or high-level description, where the functionality is modeled using programming or hardware description languages. The next stage is the register-transfer level (RTL), where data flow between registers and control logic are defined. After RTL design, the process moves to the gate-level representation, where the design is implemented using logic gates. Finally, the physical or layout level represents the actual hardware implementation on silicon, completing the IP core design flow.

At the system and algorithmic levels, the focus lies on functional behavior and computational modeling. Designs are often expressed using control/data flow representations, which describe how data moves and operations are executed. These high-level models serve as the foundation for HLS framework, where algorithmic descriptions are transformed into RTL implementations. The RTL level defines data paths, control logic, and storage elements, forming what is commonly referred to as a soft IP. Soft IPs offer flexibility, portability, and ease of customization, making them widely adopted in modern SoC designs [18], [20].

Beyond RTL, designs are synthesized into gate-level representations, producing firm IP cores that are more technology-dependent and less flexible. These are further transformed into physical layouts, known as hard IPs, which are optimized for a specific fabrication process and are not easily portable across foundries. While hard IPs offer higher security due to their fixed nature, they lack the adaptability of soft IPs.

IP cores are also categorized based on complexity and functionality. Micro-IP cores consist of small logic components such as registers and basic combinational circuits, whereas macro-IP cores represent larger functional blocks like processors, signal processing units, and multimedia accelerators. The selection between micro and macro-IPs depends on system complexity, performance requirements, and design constraints. Overall, abstraction-based design enables efficient development while balancing flexibility, performance, and security considerations across modern hardware systems [12], [14], [18].

1.3 Transformation of Integrated Circuit Design in the Era of Reusable Hardware

The evolution of integrated circuit (IC) design has undergone a fundamental transformation with the widespread adoption of reusable hardware components. Early IC development relied heavily on custom, monolithic designs tailored for specific applications, resulting in

long development cycles, high design costs, and limited scalability. As system complexity increased and market demands intensified, this traditional approach became impractical. The need for faster innovation, reduced time-to-market, and improved design reliability led to the emergence of reusable hardware paradigms [14], [16], [18].

Reusable hardware IP cores, has significantly reshaped modern design methodologies. Instead of building complete systems from scratch, designers now integrate pre-verified and configurable IP blocks to construct complex systems efficiently. This approach enables rapid prototyping, simplifies verification efforts, and enhances design productivity. The reuse of standardized components also promotes consistency across different platforms while allowing designers to focus on system-level innovation rather than low-level implementation details [13], [16], [18].

The integration of reusable IPs has further encouraged the adoption of modular and scalable system architectures. Modern SoC designs often consist of heterogeneous components such as processors, accelerators, memory subsystems, and communication interfaces, all interconnected through standardized protocols. This modular approach supports design portability across multiple technology nodes and manufacturing processes, making it suitable for diverse application domains including consumer electronics, automotive systems, healthcare devices, and artificial intelligence platforms.

However, the widespread reliance on reusable hardware has also introduced new design challenges. As IP cores are frequently sourced from third-party vendors and integrated across distributed design environments, concerns related to trust, ownership, and security have become increasingly critical. Ensuring the authenticity, integrity, and proper usage of IP components is now a central issue in modern IC design. Consequently, the transformation toward reusable hardware not only enhances efficiency and scalability but also necessitates the development of robust protection mechanisms to safeguard intellectual property throughout the design lifecycle [26], [29].

1.4 Role of High-Level Design Abstractions in Modern Hardware System Development

The increasing complexity of modern digital systems has made high-level design abstractions an essential component of contemporary hardware development. As system functionality continues to expand, designing hardware at low abstraction levels becomes time-consuming, error-prone, and difficult to scale. High-level abstractions enable designers to focus on functionality, system behavior, and performance objectives without being constrained by low-level implementation details during the early stages of development.

High-level design methodologies allow system functionality to be described using algorithmic or behavioral models, which can later be translated into hardware implementations through automated synthesis tools. This abstraction facilitates rapid design exploration, early validation, and efficient optimization of architectural parameters such as performance, power consumption, and resource utilization. By separating functionality from physical implementation, designers can explore multiple architectural alternatives before committing to a final hardware realization [16], [18], [20].

Another key advantage of high-level abstraction is improved design productivity and maintainability. High-level representations promote modular design practices, enabling reuse of functional components across different projects and platforms. This modularity simplifies system integration and enhances scalability, particularly in complex systems that incorporate heterogeneous computing elements such as processors, accelerators, and memory subsystems.

Furthermore, high-level abstractions play a crucial role in enabling the integration of security mechanisms early in the design process. Security-aware modeling at higher abstraction levels allows designers to incorporate security strategies such as access control, obfuscation, and secure data handling before hardware implementation begins. This early integration helps ensure consistent security enforcement throughout the design flow while minimizing overhead [29], [31].

Overall, high-level design abstractions serve as a foundation for efficient, flexible, and secure hardware development. By bridging the gap between algorithmic intent and physical implementation, they support the creation of scalable and reliable systems capable of meeting the growing demands of modern applications.

1.5 Security Risks in Contemporary IP-Centric Design Ecosystem

The increasing reliance on reusable IP cores in modern hardware design has significantly elevated concerns related to security and intellectual property protection. IP cores represent substantial investments in research, development, and engineering effort, making them highly valuable assets within the semiconductor ecosystem. As a result, they are frequent targets for unauthorized replication, reverse engineering, and illegal redistribution. These risks are further intensified by the widespread adoption of synthesizable RTL-based IPs, which can be easily copied, modified, or integrated into unauthorized systems without the owner’s consent [12], [16], [26].

To mitigate such threats, several protection mechanisms have been developed to safeguard IP integrity and ownership. Among the most widely adopted techniques are hardware watermarking and fingerprinting [16], [18], [20].

***IP Piracy*:** IP Piracy is a significant security threat that involves the unauthorized reuse, duplication, or distribution of protected intellectual property cores by third-party entities. This problem commonly arises in the global semiconductor design supply chain where reusable hardware modules are shared across multiple design stages. In such cases, a malicious integrator or designer may illegally copy a hardware IP core—such as a digital image processing filter—and incorporate it into their own System-on-Chip (SoC) design without obtaining proper authorization or licensing from the original creator. This unauthorized replication leads to substantial financial losses for IP vendors and undermines the incentive for innovation. In addition to economic damage, pirated hardware components may introduce functional and reliability risks; for instance, unauthorized or modified IP used in critical systems such as medical devices can result in incorrect data processing and potentially harmful outcomes. Therefore, protecting hardware IP from piracy is essential to maintain both design integrity and trust in electronic systems

***False claim of IP ownership*:** False Claim of IP Ownership is a critical security threat in which an attacker fraudulently asserts authorship over a hardware IP core that they did not originally design. This typically occurs when a rogue entity within the global design supply chain, such as a System-on-Chip (SoC) integrator, attempts to claim ownership of an existing reusable hardware module. Attackers often attempt to support these claims using techniques such as *ghost signature search attacks*, where they search for coincidental patterns or marks in unsecured designs and present them as fabricated proof of authorship. Such fraudulent claims can create legal disputes and complicate the verification of legitimate ownership. To address this challenge, hardware designers can embed unique and tamper-resistant digital evidence directly into the design. Techniques such as encrypted facial and amino-acid biometric signatures can serve as secure identifiers, providing verifiable proof of authorship and enabling designers to defend against unauthorized ownership claims.

Watermarking embeds an identifiable signature within the design to establish authorship and ownership, while fingerprinting introduces unique identifiers that enable tracing of individual IP instances in the event of leakage or misuse. These mechanisms are designed to remain covert, resilient to tampering, and minimally intrusive, ensuring that performance, power consumption, and area overheads remain within acceptable limits. Depending on design requirements, these protection features may be integrated at various abstraction levels, including behavioural, register-transfer level (RTL), or gate-level representations, with growing emphasis on early-stage integration during HLS to improve robustness and traceability [16], [18], [21], [52], [53].

Beyond watermarking and fingerprinting, additional protection strategies are also employed. IP metering allows vendors to regulate IP usage by embedding unique identifiers that enable licensing enforcement and usage tracking. Another emerging approach is Computational Forensic Engineering (CFE), which analyses structural and

statistical characteristics of hardware designs to infer authorship, detect unauthorized modifications, and identify synthesis tools used during implementation. While legal safeguards such as patents and copyrights offer formal protection, they are often insufficient against internal threats, piracy, reverse engineering, or unauthorized reuse within complex design ecosystems [12], [26], [29].

Collectively, these challenges highlight the growing need for integrated, security-aware design methodologies capable of protecting intellectual property throughout the entire hardware lifecycle. Addressing these concerns is critical to ensuring trust, accountability, and sustainability within modern IP-centric design environments [12], [29].

1.6 Secure High-Level Design Methodology for Secured IP Generation

The increasing complexity of modern digital systems has necessitated the adoption of secure design methodologies that integrate protection mechanisms early in the development cycle. HLS provides an effective platform for implementing such methodologies, as it enables designers to incorporate security features at higher abstraction levels before hardware realization. By embedding protection mechanisms during high-level design, security becomes an inherent part of the architecture rather than an afterthought added during later implementation stages [1], [2], [6], [16], [18].

A secure high-level design methodology focuses on integrating protection/security strategies directly into the functional and structural representation of a system. This includes embedding security-aware constraints, architectural transformations, and controlled design variations that preserve functional correctness while preventing unauthorized replication or manipulation. At this stage, design parameters such as scheduling, resource allocation, and data flow can be strategically influenced to incorporate protection features without introducing significant overhead [16], [18], [20].

One of the key advantages of high-level security integration is the ability to achieve strong protection with minimal impact on performance, power, and area. Since, design decisions at this level propagate throughout lower abstraction layers, security mechanisms introduced early naturally permeate the entire implementation. This approach also enhances portability, as protected designs can be synthesized across different platforms while maintaining consistent security characteristics [18], [20].

Additionally, high-level secure design methodologies enable seamless integration with cryptographic and identity-based protection techniques. Secure signature, encrypted design parameters, and authentication features can be embedded into the design flow, ensuring traceability and ownership verification throughout the IP lifecycle. Such integration

supports robust defense against threats such as unauthorized reuse/piracy, false claim of Ip ownership, and tampering [1], [2], [53].

Overall, a secure high-level design methodology provides a scalable and flexible framework for generating protected IP cores. By unifying functional correctness, performance optimization, and security enforcement within a single design flow, this approach offers a reliable foundation for developing trustworthy and resilient hardware systems in modern computing environments [1], [6].

1.7 Organization of the Thesis

This thesis is structured into six chapters, each addressing a specific aspect of secure hardware IP design with an emphasis on biometric-assisted protection and high-level synthesis based methodologies.

Chapter 1 introduces the motivation behind secure hardware IP development and discusses the evolution of reusable hardware design, high-level design abstractions, and security challenges in IP-centric ecosystems. It outlines the research objectives and contributions of the thesis.

Chapter 2 provides a detailed survey of existing research in the field of hardware security, including IP watermarking, steganography, authentication mechanisms, biometric-based security approaches, and HLS-driven protection techniques [1], [2], [6], [16], [18], [29], [52], [53].

Chapter 3 addresses the growing threat of hardware IP piracy in modern global design supply chains by introducing a biometric-driven security framework for robust ownership protection. The methodology generates a high-entropy fused signature by combining the IP seller's amino acid biometric data and facial anthropometric features, which is then encrypted using a multi-key, non-invertible fingerprint-derived AES key. The encrypted signature is embedded as covert security constraints during the high-level synthesis (HLS) process, ensuring seamless integration without hardware overhead. By leveraging multimodal biometrics and strong cryptographic diffusion, the approach significantly enhances tamper tolerance and ownership credibility. Experimental results demonstrate extremely high resistance to brute-force attacks, negligible probability of false ownership claims, and exceptionally strong entropy, making the proposed framework a reliable and legally defensible solution for hardware IP protection.

Chapter 4 introduces a secure hardware IP watermarking methodology implemented during high-level synthesis (HLS), where the IP vendor's genomic sequence is utilized to generate an encrypted and covert signature termed "Gen-Sign." The genomic data is encoded and cryptographically protected to serve as an imperceptible watermark embedded within convolutional neural network (CNN) and image processing filter IP cores. By integrating

the watermark at the synthesis stage, the approach ensures structural coupling with the hardware design while maintaining functional correctness. The proposed technique is evaluated using standard security metrics including design overhead, probability of coincidence, tamper tolerance, and entropy. Experimental results demonstrate enhanced robustness, strong resistance against piracy attempts, and negligible implementation overhead, establishing Gen-Sign as an effective and practical solution for hardware IP protection.

Chapter 5 analyses the robustness of the proposed framework under different attack scenarios, including forgery, spoofing, brute-force tampering, and false-positive detection. The chapter highlights the advantages of biometric-driven protection over conventional techniques. It also presents the experimental analysis and findings of the two different security approaches proposed in this thesis. It evaluates the effectiveness of the approaches using various benchmark designs and analyzes important design metrics such as area, latency, cost, entropy, and tamper tolerance. The chapter also compares the performance of the proposed methods with existing techniques to demonstrate their advantages in terms of security strength and implementation efficiency. The results validate that the proposed approaches provide enhanced hardware IP protection while maintaining minimal design overhead.

Chapter 6 concludes the thesis by summarizing the key research contributions and findings. It also outlines potential future research directions, such as extending the framework to multi-modal biometric systems, real-time security monitoring, and secure deployment across heterogeneous SoC platforms.

Chapter 2

2.1 Review of Past Work and Problem Formulation

The rapid adoption of reusable hardware IP cores has transformed modern IC design process, enabling faster development cycles and improved design productivity. However, this paradigm shift has also introduced serious security concerns, including IP piracy, unauthorized reuse, overproduction, and false ownership claim. As IP cores represent significant investments in research and development, protecting them from malicious exploitation has become a critical research problem [12], [16], [26].

Early efforts in hardware IP protection primarily relied on legal mechanisms such as patents and copyrights. While these methods provide legal ownership claim, however, they are insufficient for detecting piracy, tampering, or misuse within complex and globally distributed design supply chain. Consequently, technical solutions capable of embedding ownership information directly into hardware designs have gained increasing attention [12], [29].

Hardware watermarking emerged as one of the earliest technical countermeasures, enabling designers to embed covert signature into hardware design. Initial watermarking approaches focused on gate-level and layout-level implementations. Although these methods provided reasonable robustness, they suffered from limited portability, technology dependence, and high design overhead. To address these drawbacks, researchers explored register-transfer level and behavioral-level watermarking, allowing earlier integration of security features and improved flexibility [16], [18], [20].

With the emergence of HLS, watermarking techniques were further extended to higher abstraction levels. HLS-based watermarking enables security constraints to be embedded

during scheduling, binding, or register allocation stages, allowing security features to propagate naturally across lower design levels. However, many existing HLS-based watermarking approaches rely on limited encoding variables or static embedding rules [16], [18], [20], [21], [52], making them vulnerable to brute-force attacks, tampering, and ghost signature search attacks.

Recent research has explored biometric-driven hardware security [4], [5], [7], [8], [9] leveraging the inherent uniqueness of biological characteristics to strengthen ownership proof. Techniques based on fingerprinting [4], facial features [5], DNA [8], and amino acid biometrics [9] have demonstrated improved robustness compared to traditional watermarking. Nevertheless, single-biometric approaches may still suffer from limited entropy or vulnerability if partial biometric information is compromised. These observations motivate the development of multi-modal, encrypted, and HLS-integrated security frameworks capable of offering stronger protection with minimal overhead.

2.2 Literature Review on IP Watermarking for CNN IP core

The widespread deployment of convolutional neural networks (CNNs) in applications such as computer vision, medical imaging, and autonomous systems has increased the demand for high-performance CNN accelerators. These accelerators are often implemented as reusable hardware IP core using HLS, making them attractive targets for IP piracy and unauthorized reuse. As a result, securing CNN-based IP designs has become an important research focus [1], [2], [81].

Several watermarking techniques have been proposed to protect CNN IPs at different abstraction levels. Some approaches embed watermarks within the neural network parameters or weights [82], while others focus on architectural-level modifications [31]. Although parameter-based watermarking is effective for software models, it becomes less reliable in hardware implementations due to quantization, optimization, and synthesis transformations [32], [81].

HLS-based watermarking approaches for CNN accelerators address these limitations by embedding ownership information directly into the hardware architecture. Prior works have demonstrated watermarking at scheduling, resource binding, and register allocation stages to create covert structural modifications [16], [18]. These methods offer improved robustness against synthesis transformations and design optimizations. However, many existing approaches generate relatively small signature spaces, which limits their resistance to brute-force removal attack and false-positive detection [20], [21], [52].

Recent studies have shown that loop unrolling and architectural customization in CNN layers can be exploited to embed stronger and more resilient watermark. By integrating

security constraints during register allocation and datapath construction, watermarking becomes tightly coupled with the CNN architecture, making unauthorized removal extremely difficult. Nevertheless, there remains a need for watermarking techniques that combine strong cryptographic protection, biometric uniqueness, and low design overhead for CNN-based IPs [18], [53].

2.3 Biometric and Genomic Encrypted Signatures for Hardware IP Protection

The growing sophistication of hardware security threats has motivated the exploration of biometric and genomic information as strong identity anchors for intellectual property (IP) protection. Unlike conventional watermarking or cryptographic techniques that rely on static keys or design-dependent parameters, biometric and genomic signature approaches leverage inherently unique biological characteristics, making them highly resistant to replication and forgery. These properties make biometric-driven security particularly suitable for asserting authorship and resolving ownership disputes in reusable hardware IP designs [4]–[9].

Biometric-based hardware security approaches typically utilize physiological traits such as fingerprints, facial features, or behavioral patterns to generate digital signature that can be embedded within hardware designs [4], [5], [7], [13]. The uniqueness and permanence of biometric characteristics provide a higher degree of confidence in ownership verification compared to traditional watermarking methods. However, direct use of biometric data raises concerns related to privacy, spoofing, and vulnerability to inversion attacks. To address these issues, recent works have focused on transforming biometric data into secure, non-invertible digital representations prior to hardware embedding [7], [13].

Genomic signature, such as those derived from amino acid or DNA sequences, offer an additional layer of security due to their vast combinational space and extremely high entropy [8], [9]. Amino acid–based signature, in particular, benefit from the diversity of protein sequences, enabling the generation of large and complex signature spaces that significantly improve tamper tolerance [9]. When used for hardware security, genomic information is processed through encoding and encryption mechanisms to produce covert security constraints suitable for design-time integration [5], [8].

To further enhance robustness, biometric and genomic signatures are often combined with cryptographic encryption techniques. Encryption ensures that the embedded signature remains unintelligible to attackers, even if partial information about the embedding process is exposed. The use of secure and non-invertible encryption keys derived from biometric traits strengthens resistance against forgery, spoofing, and brute-force attacks. This

combination of biological uniqueness and cryptographic strength results in a highly reliable form of digital evidence for IP ownership [13], [17].

Recent research has demonstrated the effectiveness of embedding encrypted biometric and genomic signatures during HLS. By integrating the signature at early synthesis stages, such as scheduling or register allocation, the security features naturally propagate through subsequent design levels while maintaining low performance and area overhead. This approach also enhances portability and scalability, making it suitable for complex IP cores including CNN accelerators and image processing modules [1], [2], [6].

Despite their advantages, biometric and genomic encrypted signature–based approaches introduce challenges related to secure key management, ethical handling of biometric data, and design complexity. These challenges necessitate carefully designed frameworks that ensure privacy preservation, non-invertibility, and efficient integration within automated design flows [6], [12]. Addressing these issues forms a central motivation for the methodologies proposed in this thesis, which aim to provide robust, low-overhead, and synthesis-aware protection mechanisms for modern hardware IP designs.

2.4 Different State-of-the-Art IP Security Mechanisms

Beyond watermarking, a variety of hardware IP protection techniques have been proposed to counter piracy and unauthorized usage. Hardware steganography conceals ownership information within functional structures, improving imperceptibility but often lacking robustness if embedding parameters are compromised [14], [35]. Cryptographic authentication schemes using hash functions, digital signatures, and encryption algorithms provide stronger security guarantees, but their effectiveness depends heavily on key secrecy and secure key management [15], [43], [58], [59].

Another class of techniques involves IP metering and licensing enforcement, where unique identifiers are embedded into individual chip instances to control usage and detect overproduction [26], [35]. While effective for post-fabrication monitoring, these methods are less suitable for resolving ownership disputes related to design-time piracy. Physical Unclonable Functions (PUFs) have also been widely studied for device authentication and chip-level security [36], [38]. However, PUF-based approaches generate unique signatures per fabricated instance and do not directly address IP ownership identification across multiple copies of a design [36], [37].

Biometric-assisted hardware security has emerged as a promising direction for addressing these limitations. Prior works have demonstrated that biological signatures, such as amino acid sequences, facial features, and fingerprint biometrics, can be transformed into digital signatures and embedded into hardware designs. These approaches significantly enhance ownership proof by linking IP designs to the natural identity of the IP owner [4]–[9].

However, earlier biometric-based methods often relied on single biometric modalities or lacked strong encryption mechanisms, leaving room for further improvement [7], [13].

Our research advances this domain by introducing encrypted, fused biometric and genomic signatures embedded during HLS, enabling robust protection against forgery, tampering, and false ownership claims while maintaining negligible performance and area overhead.

2.5 Summary

This chapter reviewed prior research on hardware IP protection, with a focus on watermarking, cryptographic security, biometric-assisted methods, and HLS-based design approaches. While significant progress has been made, existing techniques often suffer from limited robustness, vulnerability to advanced attacks, or high implementation overhead. In particular, securing CNN-based and computation-intensive IP cores remains a challenging problem due to aggressive design optimization and architectural complexity [1], [2], [6], [16], [18].

The analysis of prior work highlights the need for a unified, secure-by-design methodology that integrates strong cryptographic protection, biometric uniqueness, and high-level synthesis-driven embedding mechanisms. Motivated by these gaps, the subsequent chapters present the proposed frameworks and methodologies that aim to advance the state of the art in secure hardware IP design [13], [17], [53].

Chapter 3

Biometric-Fusion Driven Hardware IP Protection Using Encrypted Fingerprint and Molecular Identity Signatures

3.1 Introduction

3.1.1 Background and Context

The rapid advancement of semiconductor technology and the increasing complexity of embedded computing platforms have led to an unprecedented demand for reusable hardware IP cores [12], [27]. In modern SoC design flow, hardware components are rarely designed from scratch. Instead, designers frequently integrate pre-verified IP blocks such as digital filters, encryption engines, signal processing modules, memory controllers, and communication interfaces to reduce development time, cost, and design risk [19], [33]. This reuse-centric approach enables faster time-to-market and improves productivity; however, it simultaneously introduces serious concerns related to security, ownership, and trust [12], [27], [28].

Among various hardware modules, digital signal and image processing IP cores are particularly valuable due to their widespread use in real-time and safety-critical applications [9], [17]. These include medical imaging systems, autonomous vehicles, industrial automation, defense electronics, robotics vision, and biometric authentication systems. Because such IP cores encapsulate significant design effort, algorithmic innovation, and optimization expertise, they represent substantial intellectual and commercial value to the IP vendor [9], [17].

Despite these benefits, the globalization of semiconductor manufacturing has drastically changed the design ecosystem [12], [27], [28]. Today, the design chain involves multiple geographically distributed and often mutually untrusted entities, including:

- IP vendors or sellers,
- SoC integrators,
- third-party verification companies,
- fabrication foundries,
- packaging and testing houses.

The participation of multiple stakeholders creates several points where sensitive design information may be exposed. Consequently, hardware IPs become vulnerable to unauthorized copying, tampering, reverse engineering, illegal redistribution/piracy, and false ownership claim [12], [27], [28].

This situation has elevated hardware IP piracy into one of the most critical challenges faced by the semiconductor industry [12], [26].

3.1.2 Hardware IP Piracy: Challenges and Risks

Hardware IP piracy refers to the illegal duplication, modification, or misuse of proprietary hardware designs without the authorization of the legitimate owner [12], [16]. Unlike software piracy, hardware piracy is often harder to detect because the stolen design is embedded inside silicon and may not exhibit obvious visual differences. The consequences of IP piracy are both economic and technical. From an economic perspective, piracy leads to:

- loss of licensing revenue,
- reduced competitive advantage,
- intellectual property theft,
- reputational damage.

From a technical and safety perspective, pirated or modified IPs may introduce:

- hidden malicious logic,
- degraded performance,
- reliability issues,
- increased power consumption,
- functional inaccuracies.

For example, if a pirated image processing filter is used inside a medical diagnostic device, incorrect computations could lead to false diagnoses [9], [59]**.** Similarly, compromised hardware inside autonomous vehicles or defence systems may cause catastrophic failures. Therefore, protecting hardware IP is not merely a financial concern but also a safety and reliability requirement [12], [25]**.**

In addition to piracy, another major problem is the false ownership claim, where an unauthorized entity claims authorship of an IP core. In the absence of strong ownership evidence, legal resolution becomes difficult. Hence, a mechanism capable of providing irrefutable proof of authorship is essential [42], [43]**.**

3.1.3 Limitations of Conventional Protection Techniques

As discussed in the related work chapter, to address hardware piracy, researchers have proposed several protection mechanisms such as watermarking, steganography, logic locking, and cryptographic signatures [16], [29], [30], [33]. Although these approaches offer partial protection, they possess inherent weaknesses. Traditional watermarking embeds a secret digital pattern inside the design structure [16], [20]. While useful, such watermarks rely on encoding parameters that may be reverse engineered. Once discovered, the watermark can be removed or replicated. Steganographic methods attempt to conceal information inside structural redundancies [14], [32]. However, the security of these approaches depends heavily on entropy thresholds and may not withstand sophisticated attacks. Cryptographic methods such as RSA, SHA, or AES generate digital signatures based on secret keys [15], [54]. Although mathematically strong, their security collapses if keys are leaked or stolen. Moreover, purely digital keys lack intrinsic linkage to the real identity of the IP owner. Therefore, most existing techniques fail to satisfy three important properties simultaneously:

- Non-replicability,
- Strong tamper resistance,
- Direct association with the owner's identity,
- This motivates the exploration of alternative protection strategies.

3.1.4 Motivation for Biometric-Based Security

Biometrics provide a natural and highly reliable mechanism for identity verification [4], [11], [13]. Biological traits such as fingerprint, facial features, and molecular characteristics exhibit extremely high uniqueness across individuals [8], [9], [45], [46]. These traits cannot be easily forged, guessed, or duplicated. Unlike conventional digital signatures, biometric traits inherently link the authentication mechanism to a real individual, providing stronger identity assurance [4], [11]. Unlike conventional digital signatures, biometric features offer several advantages:

- inherent randomness,
- extremely large entropy space,
- strong uniqueness,
- resistance to brute-force guessing,
- direct linkage to human identity.

Because of these properties, embedding biometric information into hardware designs provides a powerful method of ownership authentication [4], [5], [9]. Even if an attacker obtains the design, reproducing the original biometric signature becomes practically impossible. However, relying on a single biometric source may still be insufficient due to

noise, environmental variations, or limited entropy. Hence, combining multiple biometric modalities can significantly strengthen security [5], [17], [62]. This observation forms the core motivation of this research.

3.1.5 Problem Statement

Although numerous hardware protection methods exist, there is still no comprehensive solution that simultaneously provides:

- strong ownership authentication,
- high tamper tolerance,
- low false positive probability,
- resistance to forgery and spoofing,
- seamless embedding within standard synthesis flows.

Existing approaches either rely on weak digital signatures or single biometric features, both of which may be compromised [7], [16]. Therefore, the primary problem addressed in this thesis can be stated as follows:

How can we design a hardware security methodology that embeds a highly unique, non-replicable, encrypted, and biometric-based ownership signature into reusable IP cores while ensuring robustness against piracy, tampering, and false ownership claim [4], [5], [13], [17].

3.1.6 Overview of the Biometric Fusion driven Hardware Security methodology

To overcome the limitations of existing techniques, this thesis presents a novel multi-

layered security methodology that integrates multiple biometric traits with cryptographic protection and synthesis-level embedding [1], [5], [13].

The proposed framework employs three independent biometric modalities:

- Amino acid–based molecular signature [5], [17]
- Facial anthropometric measurements [48]
- Fingerprint minutiae features [3], [13].

These signatures are fused together to form a composite biometric template with extremely high entropy [5], [17]. The fused template is then encrypted using a secure non-invertible

fingerprint-derived AES key [7], [13]. The encrypted result is embedded into the hardware IP through covert structural constraints during the HLS process [15], [54].

Because the embedded signature originates from the natural biological characteristics of the IP vendor, it becomes inherently unique and practically impossible to forge. [4], [9]. Furthermore, encryption ensures that the signature cannot be extracted or interpreted by unauthorized users [13], [30].

Consequently, the proposed solution offers:

- stronger ownership proof
- improved tamper resistance
- stronger entropy
- reduced probability of coincidence
- reliable piracy detection.

3.2 Biometric-Fusion Driven Hardware Security Methodology

3.2.1 Amino Acid Signature Generation (Module 1)

The first stage of the proposed security methodology generates a biometric signature using the amino acid structure obtained from the IP seller's body sample [5], [17]. Amino acids constitute the fundamental building blocks of proteins in the human body. Since twenty chemically distinct amino acids exist, numerous possible combinations of polypeptide chains can be formed. For a chain of length n, the total number of possible combinations

equals 20^n. This large combinational space provides extremely high diversity, making amino acid sequences suitable for unique identification [5], [17].

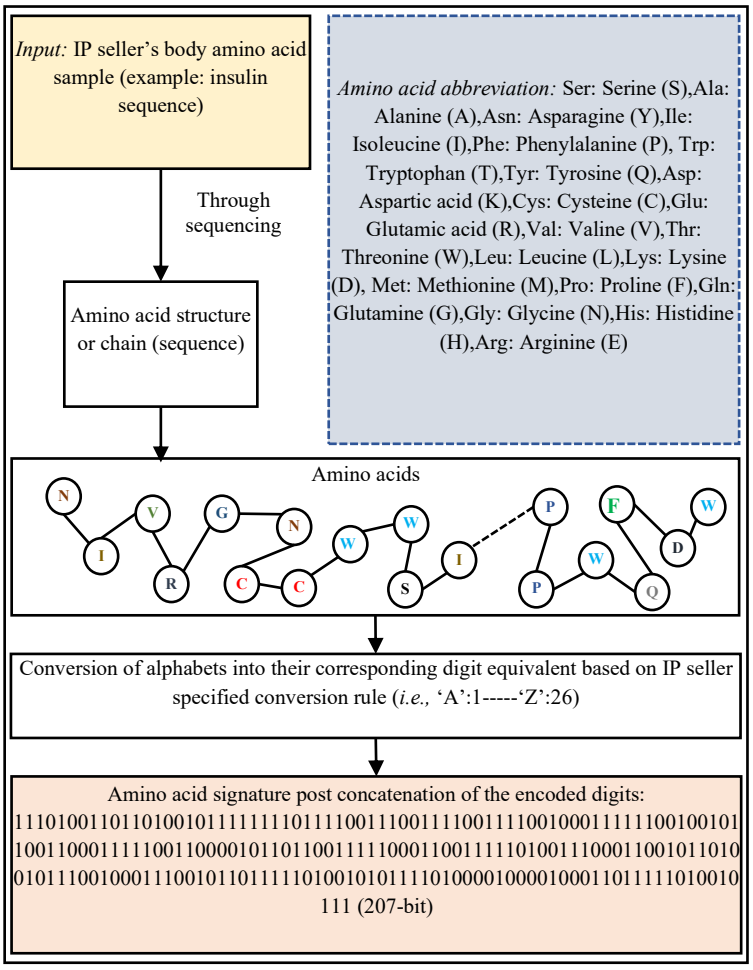


Fig. 3.1. Generation of the Amino Acid Signature

Fig. 3.1 Explains the The generation of an amino acid signature from an IP seller's biological sample (such as an insulin sequence) involves a systematic process to transform biological data into a digital format for hardware security.

In the proposed procedure, an amino acid sample is first collected from the IP seller. The sample may originate from biological sources such as hair, saliva, muscle, or blood. After collection, amino acid sequencing is performed using standard sequencing techniques. The sequencing process produces an ordered chain of amino acid symbols. For illustration, the insulin sequence of 51 amino acids is considered [5]. The obtained sequence is:

“NIVRGCCWSICSLQGLRYQCYPVYGHLCNSHLVRALQLVCNRENPPQWFDW”

Each alphabet in this chain is then converted into its corresponding numeric value according to an IP seller–specified encoding rule, where ‘A’ corresponds to 1 and ‘Z’ corresponds to 26. These decimal numbers are subsequently converted into their binary equivalents. Finally, all binary representations are concatenated sequentially to generate a long binary string. This concatenated binary sequence forms the **amino acid signature**, which serves as the first component of the fused biometric identity [5], [17]. Because different body samples and polypeptide chains produce different sequences, multiple variations of amino acid signatures can be derived from the same individual.

3.2.2 Facial Anthropometric Signature Generation (Module 2)

The second stage of the methodology generates a facial anthropometric signature based on the geometric structure of the IP seller’s face [5], [7]. Instead of using general facial biometrics, the method relies on precise anthropometric measurements, which are commonly used in forensic identification due to their reliability and accuracy [7], [11]. Initially, the IP seller’s facial image is captured using a high-resolution digital camera. The captured image is placed under a predefined grid size and spacing selected by the IP seller. The grid ensures accurate determination of nodal points and spatial coordinates.

Seven anthropometric features are extracted:

- head breadth,
- chin width,
- Menton–nasal bridge length,
- face length,
- lower face length,
- lip width,
- Menton chin length.

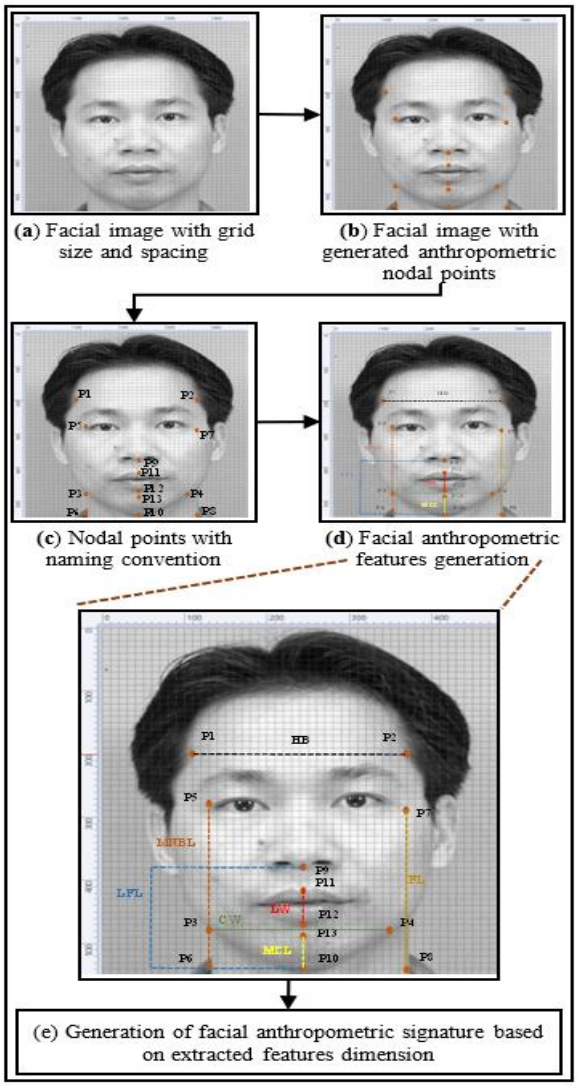


Fig. 3.2 Facial biometric feature extraction and binary signature generation process.

Fig. 3.2 Explains The generation process begins with capturing the IP seller's facial image and applying a grid with specific spacing to define anthropometric nodal points using a naming convention. Seven distinct features, such as head breadth, chin width, and face length, are then extracted as detailed measurements of physical proportions. These dimensions are converted into binary equivalents and concatenated in a secret order chosen by the seller.

For each feature, nodal points are marked on the face image. These nodal points are labelled using a naming convention. After identifying the coordinates of corresponding nodal pairs, the feature dimension is calculated using the Manhattan distance between the two points.

Each calculated dimension is then converted into its binary equivalent. The binary values of all extracted features are concatenated in an IP seller-specified order. The concatenated binary string forms the **facial anthropometric signature**. This signature provides a detailed geometric representation of the IP seller's facial structure and offers higher precision and robustness compared to traditional facial biometrics [5], [7], [11].

3.2.3 Secure Non-Invertible Fingerprint Processing (Module 3)

The third module produces a secure non-invertible fingerprint signature that functions as an encryption key. The fingerprint of the IP seller is first captured using an optical scanner. The captured image undergoes preprocessing operations, including binarization, thinning, and Fast Fourier Transform (FFT) enhancement. These steps improve ridge clarity and facilitate minutiae extraction [7], [13], [48].

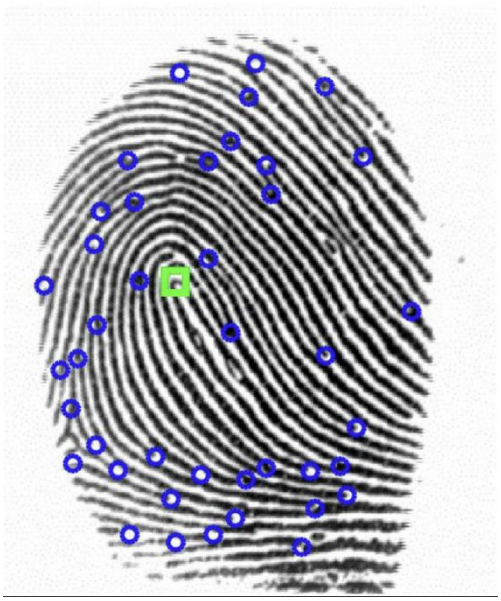


Fig. 3.3 Minutiae Point in the Fingerprint Image

Fig. 3.3 demonstrates the extraction of specific minutiae points ridge endings and ridge bifurcations from the IP seller's fingerprint.

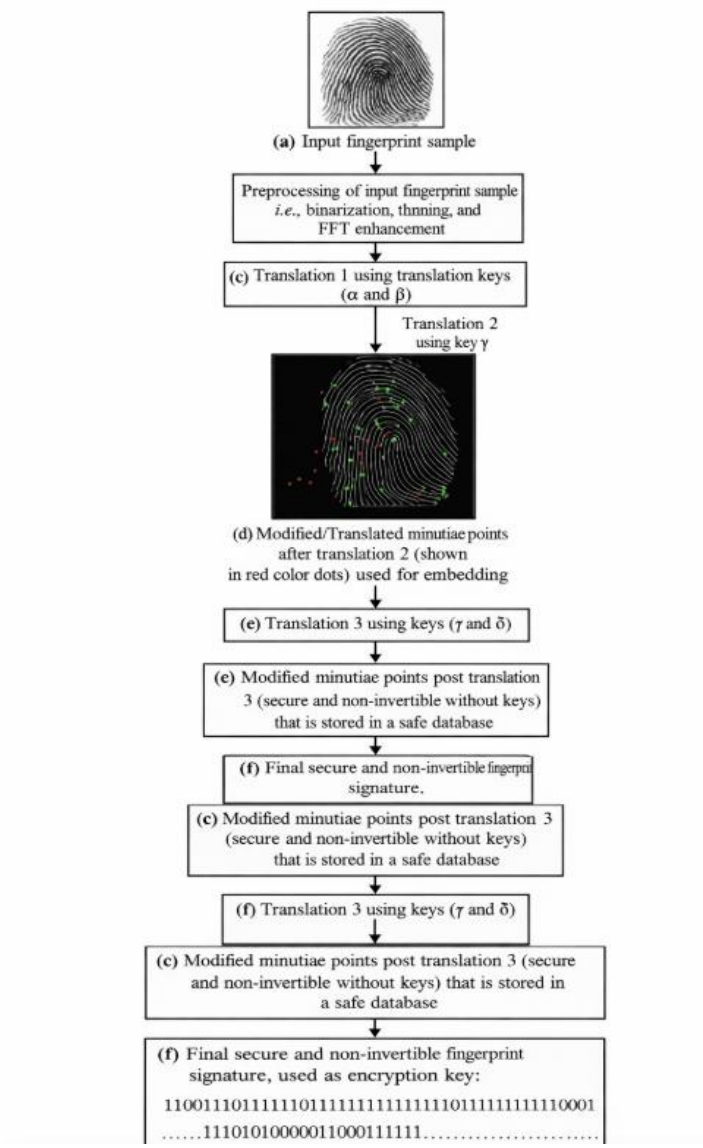


Fig. 3.4 Fingerprint-based encryption key generation using multi-stage translation of minutiae points.

Fig 3.4 This process illustrates the systematic transformation of an IP seller's raw fingerprint into a secure, non-invertible encryption key. By applying multi-stage translation keys to extracted minutiae points, the methodology ensures the final binary signature is cryptographically robust and mathematically shielded from reversal. This sequence provides a high-entropy, detective countermeasure that uniquely links the hardware design to its legitimate owner.

Minutiae points are then determined using the crossing number algorithm. Points with crossing number 1 are identified as ridge endings, while those with crossing number 3 correspond to bifurcations. For each minutiae point, parameters such as x-coordinate, y-

coordinate, ridge angle, and type are extracted. To ensure non-invertibility, three mathematical translations are applied:

Translation 1 (T1): modifies minutiae using keys α and β.

Translation 2 (T2): rotates points using key γ.

Translation 3 (T3): applies final transformation using keys γ and δ.

Minutiae after T2 are used to generate the signature, while T3 points are stored securely. Storing T3 points prevents attackers from reconstructing the original signature without the secret keys. Each T2 parameter is converted to binary and concatenated according to the rule: *x-coordinate # y-coordinate # minutiae type # ridge angle.*

The resulting binary stream forms the **secure non-invertible fingerprint signature**, which is divided into 128-bit segments. A selected segment is used as the encryption key.

3.2.4 Generation of Encrypted Fused Signature Using AES

After obtaining the amino acid signature and the facial anthropometric signature, both binary templates are concatenated to form a unified fused signature [5], [58]. This combined signature acts as the primary data input to the cryptographic block. The fused signature is then encrypted using the Advanced Encryption Standard (AES) [58]. The encryption key is derived from the secure non-invertible fingerprint signature generated in the previous module. A selected 128-bit segment of the fingerprint signature is used as the AES key [58].

Inside the AES block, the encryption process proceeds through multiple rounds. Initially, substitution operations introduce non-linear transformations to provide confusion. This is followed by row shifting and mix-column transformations to achieve diffusion across the bitstream. Subsequently, XOR operations are performed between the transformed data and the fingerprint-derived key. The output of one round becomes the input to the next, and after all rounds are completed, the final encrypted fused signature is produced. This encrypted signature conceals the raw biometric information and provides cryptographic protection before embedding [58].

3.2.5 Embedding of Covert Encrypted Signature in Hardware IP (Module 4)

The encrypted fused signature is next utilized to derive covert hardware security constraints [1], [15]. These constraints are embedded during the HLS design process. Initially, the algorithmic description of the image filter is converted into a control data flow graph (CDFG). Using architectural constraints and scheduling techniques (LIST scheduling), a scheduled data flow graph is generated. From this representation, a register allocation table is created.

X-axis	*Y-axis*	*Type*	*Angle (in degree)*	*Binary equivalent*
205	106	3	23	11001101111010101110111
90	98	1	120	10110101100010111111000
203	116	3	162	11001011111010011101000010
221	130	3	5	11011101100000101101
229	182	3	198	11100101101101101111000110
217	176	1	44	11011001101100001101100
200	183	1	290	11001000101101111100100010
217	219	1	120	11011001110110111111000
185	206	1	31	10111001110011101111111
165	218	1	53	10100101110110101110101
192	202	3	128	11000000110010101110000000
117	203	1	303	11101011100101111100101111
169	234	3	39	10101001111010101100111
167	255	1	284	10100111111111111100011100
149	241	1	119	10010101111100011110111
82	238	1	297	10100101110111011001010001
285	350	3	282	100011101101011110110011010
84	280	3	272	10101001000110001110001000
75	315	3	52	10010111001110111110100
49	304	1	268	11000110011000001100001100
204	383	1	262	11001100101111111110000010
34	314	3	272	10001010011101011100010000
200	392	3	71	11001000110001000111000111
196	401	3	279	11000100110010001111000101
205	423	3	58	11001101110100111111111010
64	398	1	26	10000001100011101110

Table 3.1 Minutiae points coordinates post-translation 2 (T2), along with type, ridge angle, and their corresponding concatenated binary equivalents.

Reg	T0	T1	T2	T3	T4	T5	T6
R	C_0	C_{22}/**C_{23}**	C_{22}/**C_{23}**	C_{32}/**C_{33}**	C_{34}/**C_{39}**	C_{35}	C_{36}/**C_{41}**
Gr	C_1	C_{23}/**C_{22}**	C_{23}/**C_{22}**	**C_{32}**	**C_{34}**	**C_{40}**	**C_{36}**
In	C_2	C_{24}/**C_{25}**	C_{24}/**C_{25}**	C_{33}/**C_{37}**	-	-	-
BL	C_3	C_{25}/C_{24}	C_{25}/**C_{24}**	**C_{38}**	-	-	-
Y	C_4	C_{26}	C_{26}/**C_{27}**	C_{26}	C_{26}	C_{26}	-
Bla	C_5	C_5	C_{27}/**C_{28}**	C_{37}	C_{39}	C_{40}	C_{41}
V	C_6	C_6	C_{28}/**C_{29}**	-	-	-	-
Pi	C_7	C_7	C_{29}/**C_{30}**	C_{38}	-	-	-
LI	C_8	C_8	C_{30}	-	-	-	-
Or	C_9	C_{31}/**C_{26}**	C_{31}/**C_{26}**	C_{31}/**C_{26}**	C_{31}/**C_{26}**	C_{31}/**C_{26}**	-
Aq	C_{10}	**C_{31}**	**C_{31}**	**C_{31}**	**C_{31}**	**C_{31}**	-
T	C_{11}	-	-	-	-	-	-
Gry	C_{12}	-	-	-	-	-	-
Ma	C_{13}	C_{13}	C_{13}	C_{13}	C_{13}	-	-
Si	C_{14}	-	-	-	-	-	-
Kh	C_{15}	-	-	-	-	-	-
Ol	C_{16}	-	-	-	-	-	-
Cy	C_{17}	-	-	-	-	-	-
Wh	C_{18}	C_{18}	C_{18}	C_{18}	C_{18}	-	-
Br	C_{19}	-	-	-	-	-	-
Be	C_{20}	-	-	-	-	-	-
Lb	C_{21}	-	-	-	-	-	-

Table 3.2 Register (Reg) allocation table: pre and post embedding secret security constraints corresponding to sharpening filter.

Table 3.1 lists the minutiae point coordinates after the second translation (T2) along with their type and ridge angle. These features are converted into binary form and concatenated to generate a unique binary representation for each minutiae point.

Table 3.2 presents the register (Reg) allocation for the sharpening filter before and after embedding the secret security constraints. It shows how registers are assigned in the design and how the allocation changes after incorporating the security information.

Watermark constraints generation rules: The embedding rule maps each bit of the encrypted signature to artificial edges between storage variables. When the bit value corresponds to '0', an edge is inserted between variable pairs with even indices. When the bit value corresponds to '1', the edge is inserted between odd-indexed pairs. These edges enforce specific register allocation relationships [1], [18], [33].

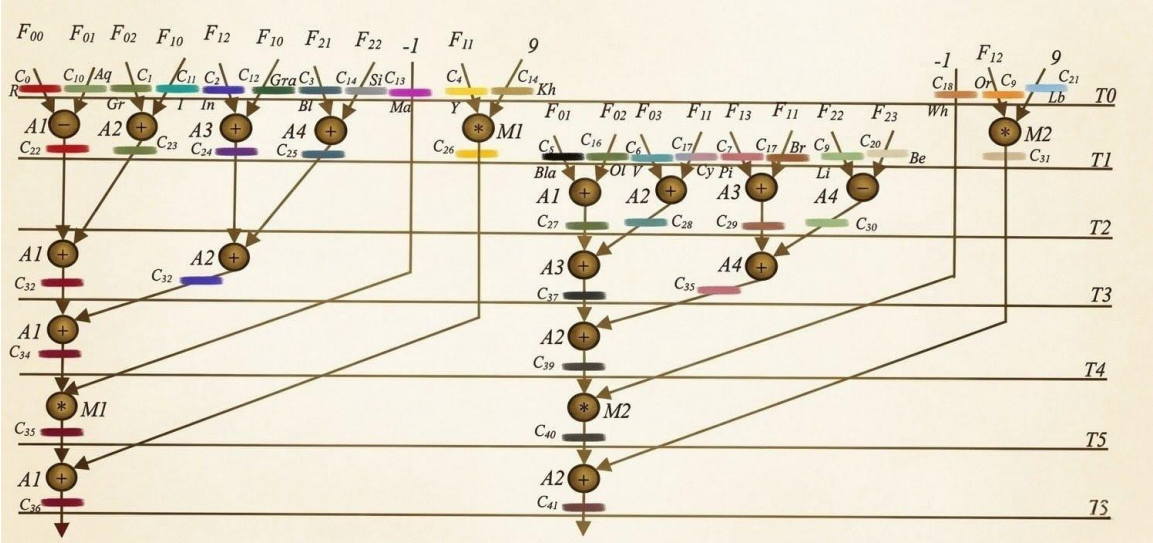


Fig. 3.5 Scheduled data flow graph (SDFG) of the sharpening image filter showing operation scheduling using four adders and two multipliers.

Fig. 3.5 Scheduled data flow graph (SDFG) of sharpening image filter using 4 adder and 2 multipliers, An SDFG is a graphical representation of an algorithm where nodes represent operations and edges represent the flow of data, specifically mapped onto a timeline of control steps. In hardware synthesis, it allows designers to visualize resource sharing and scheduling, ensuring that the operations (like additions and multiplications) are executed within specific clock cycles based on available hardware constraints.

By incorporating these additional constraints into the register allocation phase, the encrypted biometric signature becomes structurally integrated into the hardware design. After synthesis, the resulting RTL datapath inherently contains the hidden ownership information [1]. Thus, the final hardware IP core carries covert security evidence without affecting normal functionality. Embedding refers to the process of implanting a unique, secret, and permanent digital mark (often called a watermark) into a design to prove ownership and detect piracy [16], [18]. The following explanation details the embedding process based on the methodology presented in the research paper:

1. Conceptual Overview of Embedding

The goal of embedding is to create "digital evidence" within the hardware design that is:

- **Covert:** It should be hidden within the design and not easily detectable by an attacker.
- **Robust:** It must resist attempts at removal or tampering, such as brute-force attacks.

- **Non-intrusive:** Ideally, it should not add significant overhead in terms of area, power, or latency to the final design [1], [16].

2. Preparation: Generating the Fused Signature

Before embedding occurs, a unique signature is generated by fusing two distinct biometric features of the IP seller:

- **Amino Acid Biometric:** Derived from the IP seller's protein sequences (e.g., insulin), which offers a massive signature space due to the billions of possible combinations [5], [17].
- **Facial Anthropometric Signature:** Created by mapping specific facial features (like head breadth and face length) into a digital string [4].
- **AES Encryption:** These features are fused and then encrypted using an AES-128 block. The encryption key is derived from the seller's **secure non-invertible fingerprint**, ensuring that only the legitimate owner can generate the specific signature [7], [33].

3. The Embedding Phase: High-Level Synthesis (HLS)

The actual embedding happens during **High-Level Synthesis (HLS)**, specifically in the **register allocation phase** [1], [16].

Steps in the Embedding Process:

1. **Generation of Security Constraints:** The encrypted fused signature is used to generate secret "hardware security constraints".
2. **Mapping to Register Allocation:** The methodology utilizes a Register Allocation Table (RAT) framework [1].
3. **Embedding Rule:** The paper describes a specific rule where an **artificial edge** (a constraint) is added between storage variable pairs.
    - For example, if a bit in the signature is ‘0’, an additional edge is embedded between variables with even indices during the register allocation phase [1], [16] .
4. **Impact on Design:** This artificial constraint forces the HLS tool to assign these variables to different registers than it otherwise would have. This "fingerprint" is now woven into the very structure of the hardware's datapath.

4. Characteristics and Advantages of this Embedding

- **Zero Design Overhead:** Remarkably, this approach introduces **zero design cost overhead**. It utilizes existing resources more specifically rather than adding new hardware, meaning the final chip is no larger or slower than an unsecured version [1] .
- **High Tamper Tolerance:** Because the signature is derived from complex biometrics and encrypted, the "signature space" is massive (up to 1.15E+77 for certain filters) [5]. This makes it practically impossible for a pirate to find and remove the watermark through brute force.
- **Seamless Verification:** To prove ownership, the IP seller does not need to recapture their biometrics. They use pre-stored, encrypted "nodal points" and "translated minutiae" along with their secret keys to reconstruct the signature and verify its presence in the design [7], [16].

3.3 Detection of IP Piracy and Ownership Validation

For piracy detection, the legitimate IP seller regenerates the encrypted fused signature using the originally stored biometric information and the secret translation keys. From this regenerated signature, the corresponding hardware security constraints are computed. These regenerated constraints are then compared with the constraints extracted from the suspected IP core. If the two sets match, the embedded signature confirms authorship and establishes ownership [16], [18], [51].

During this validation procedure, recapturing of biometric samples is not required. Fingerprint minutiae points after the third translation and the necessary anthropometric data are already stored securely. This eliminates dependence on environmental factors such as lighting conditions or physical variations and ensures consistent verification. Through this comparison mechanism, unauthorized duplication or false ownership claims can be reliably detected.

3.4 Summary

This paper addresses the critical challenge of hardware intellectual property (IP) piracy within the global semiconductor design supply chain, where increasing reliance on third-party IP cores in System-on-Chip (SoC) integration has amplified risks of unauthorized reuse and false ownership claims. Conventional watermarking approaches often suffer from limited entropy and weak tamper tolerance, making them vulnerable to sophisticated attacks such as ghost signature searches. To overcome these limitations, the proposed methodology introduces a multi-modal biometric hardware security framework

that establishes a strong and verifiable link between the IP seller and the digital design. The approach generates a high-entropy fused biometric signature by combining two biological datasets: the IP seller's amino acid sequence derived from an insulin sample and their facial anthropometric measurements. The amino acid sequence is encoded into a 207-bit digital signature, while seven facial measurements are converted into a 56-bit representation; these are concatenated to form a 263-bit fused signature that acts as digital ownership evidence. To further enhance security, the fused signature is encrypted using a non-invertible fingerprint-based mechanism in which fingerprint minutiae points undergo three levels of mathematical translation before being used as the encryption key for an Advanced Encryption Standard (AES) engine, preventing reconstruction of the original biometric data. The encrypted signature is then embedded during the high-level synthesis (HLS) stage as covert security constraints within the behavioural description of the hardware design, ensuring that the watermark becomes an intrinsic part of the logic structure without significant performance or power overhead. By integrating biological identity, cryptographic protection, and design-time embedding within the HLS workflow, the methodology provides a robust multi-layered defence against IP piracy and false ownership claims, enabling reliable digital ownership verification and strengthening security within the global hardware IP ecosystem.

Chapter 4

Gen-Sign: HLS based Watermarking using IP Vendor's Feistel Cipher Encrypted Genomic Signature for Protecting CNN and Image Processing Filter Cores against Piracy

4.1 Introduction

A CNN is a class of deep neural networks; most commonly applied for analyzing visual imagery [81]. In machine learning, CNNs have been a cornerstone for tasks such as image and video recognition, image classification, medical image analysis and many others, demonstrating remarkable performance in extracting patterns/features from visual data, which is highly computationally intensive [7], [75] [81].

Image processing encompasses techniques used to perform operations on images, including enhancing, altering, and analyzing them. One major operation is the application of a sharpening filter for image enhancement [17], [75].

Both CNN feature extraction and sharpening filters are computationally intensive; therefore, hardware IP designs are required [66], [81]. It is also imperative to safeguard the IP design against piracy, which may lead to revenue losses, goodwill damage, and malicious logic insertion [66], [75], [81].

The proposed approach presents a security-aware HLS watermarking methodology using IP vendor's Feistel cipher encrypted genomic signature for protecting CNN and image processing filter IP cores against piracy [17], [81].

4.2 Hardware Watermarking Technique: Gen-Sign

4.2.1 Overview of Gen-Sign

This paper presents a triple layered hardware security approach using Feistel cipher based encrypted genomic signature to protect IP cores against piracy [8], [17], [18]. The input block consists of:

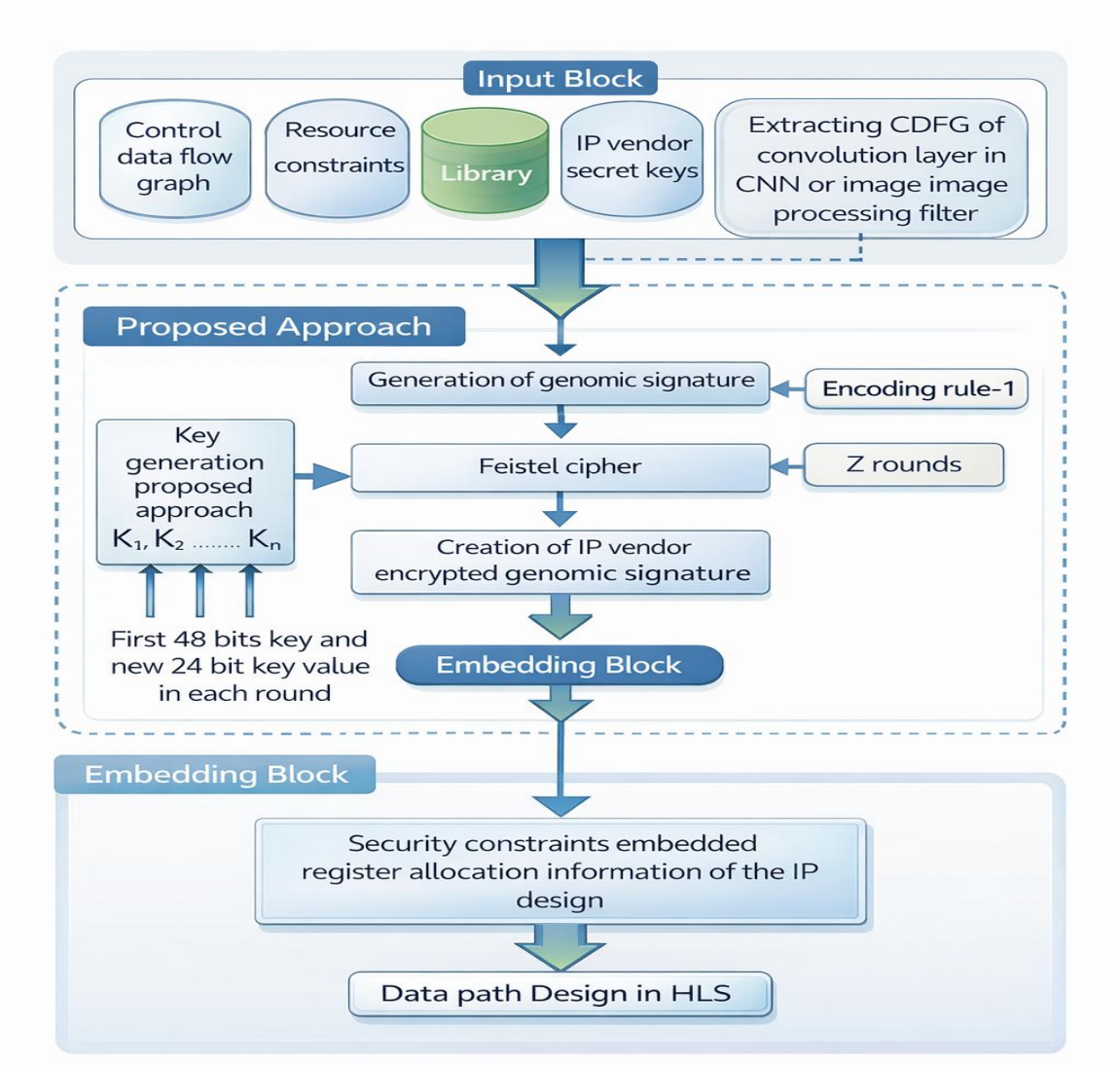


Fig. 4.1 Overview of the proposed Gen-Sign methodology for HLS-based hardware IP watermarking and secure datapath synthesis.

- resource constraints,
- library,
- IP vendor secret key,
- IP vendor genomic base pair,
- CDFG of CNN convolution layer or image filter [7], [75].

The output block consists of a secure datapath design using HLS.

• The first block is responsible for the extraction of genomic sequence from the human body sample of IP vendor [36]. The genomic base pair comprises of the following chemical elements: Adenine: A, Thymine: T, Guanine: G, Cytosine: C. The genomic sequence comprising base pairs of distinct types is encoded using encoding rule-1, to produce the genomic signature.

• The second block is accountable for the encryption of the base pair using Feistel cipher. This encryption process uses IP vendor's key (K1, K2…..Kz), generated by the key generation process based on number of rounds (R) chosen by the IP vendor.

• The third block is accountable for generation of IP vendor encrypted genomic signature.

• The encrypted genomic signature is further used to produce the encoded security constraints using the encoding rule-2. The encoded security constraint is embedded in the register allocation phase of HLS [16], [18], [33].

The security constraints post embedded register allocation information is used to generate the IP design data path using classical high level synthesis design flow.

4.3 Details of the Presents Encrypted Genomic Signature: Gen-Sign

4.3.1 Extraction of Genomic Sequence and Creation of Genomic Signature

The IP designer extracts genomic sequence, which could include alternative base pairs of the distinct types [1], [8]. This sequence is then converted into binary digits following a specific encoding rule defined by the IP designer, known as encoding rule-1.

To provide an accurate response of encoding rule-1 as it pertains to the chemical components (A, T, G, C, and S) is represented by a 2-bit binary code used in forming the final genomic base pair. "S" represents a polynucleotide (Sugar phosphate backbone). Polynucleotide has been incorporated as both the leading and lagging strands in the genomic sequence.

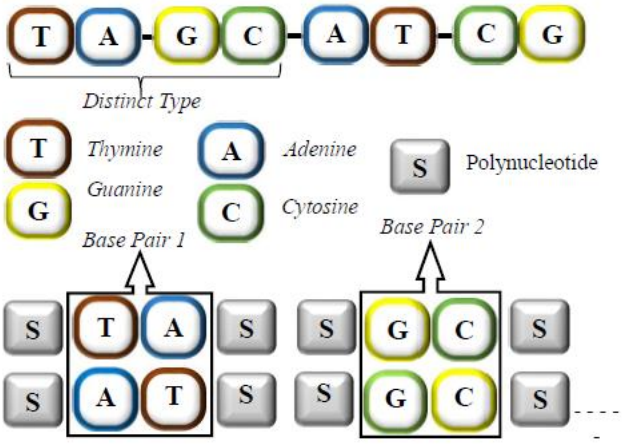


Fig. 4.2 Genomic sequence with base pairs and polynucleotide

Fig. 4.2 shows Representation of the genomic sequence composed of DNA base pairs (A, T, G, C) and the polynucleotide backbone used for generating the vendor-specific genomic signature.

Illustration of encoding rule:

Element ‘A’ is encoded in binary as ‘1’

Element ‘T’ as ‘10100’

Element ‘G’ as ‘111’

Element ‘C’ as ‘11’

Element ‘S’ as ‘10011’

Illustration of a final genomic sequence featuring alternative base pairing of distinct type:

STAS-SGCS-SATS-SCGS-STAS-SGCS-SATS-SCGS-S……

Binary-encoded genomic representation (128-bit example):

1001110100110011–100111111110011–1001111010010011–

100111111110011–1001110100110011–100111111110011–

1001111010010011–100111111110011–1001

4.3.2 Generation of Encrypted Genomic Signature

This binary-encoded genomic sequence is then processed using the Feistel cipher to encrypt the extracted data [8].

The Feistel cipher begins by selecting 64 bits from the 128 bits of the binary coded genomic sequences as input. These 64 bits are then split into two sections: the left 32 bits and the right 32 bits.

The encryption function unfolds in three main steps:

• The right 32 bits are processed through an expanded P-box, which transforms these 32 bits into 48 bits through permutation/diffusion.
• The 48 bits undergo an XOR operation with a 48-bit key supplied by the IP vendor.

• The output is fed into an S-box which transforms the 48 bits into 32 bits through substitution/confusion.

These 32 bits are then:

Used as left bits for next round

XORed with the initial left bits to produce next right bits

The encryption continues for 100 rounds (R) [36].

Key generation:

- Initial 48-bit key selection
- Split into two 24-bit parts
- Circular shifts
- XOR operation
- Concatenation
- Generate next 48-bit key

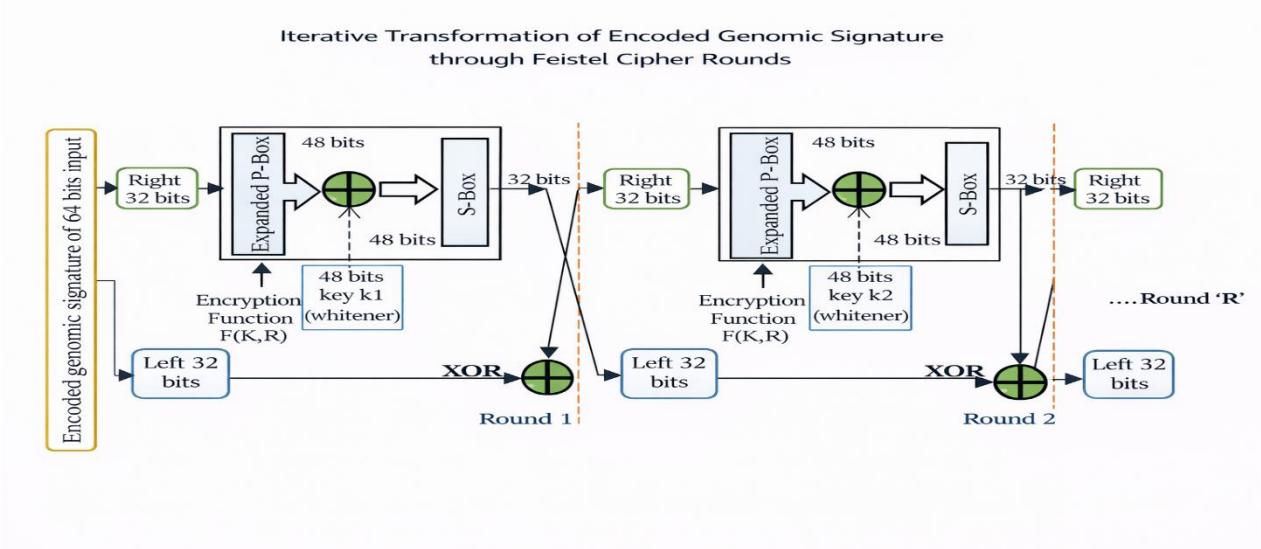


Fig. 4.3 Feistel cipher integrated in proposed Gen-Sign

Fig. 4.3 shows the Multi-round Feistel cipher structure employed in the Gen-Sign methodology, illustrating the division of the 64-bit input into left and right halves, expansion through P-box, key mixing via XOR operation, S-box substitution, and iterative round-based encryption using vendor-specific keys .

Final encrypted 128-bit genomic signature:

11001011010011110111010000001001111110100111110011

01001011010001010101000001010100010011111000110100

01111011111100001100010101 01.

4.4 Embedding of Encrypted Genomic Signature on CNN Convolutional Layer and Sharpening Filter

The convolutional layer in CNN is defined as follows:

$O = \Sigma \Sigma I_{ij} \times H_{ij}$

Where "I" represents the input image matrix of dimension M × N, and "H" denotes the convolution filter kernel matrix with dimensions m × n.

The sharpening filter (using mean or box kernel) can be defined as follows:

$O_0 = [(I_{00} + I_{01} + I_{02} + I_{10} + I_{12} + I_{20} + I_{21} + I_{22}) \times (-1)] + (I_{11} \times 9)$

$O_1 = [(I_{01} + I_{02} + I_{03} + I_{11} + I_{13} + I_{21} + I_{22} + I_{23}) \times (-1)] + (I_{12} \times 9)$

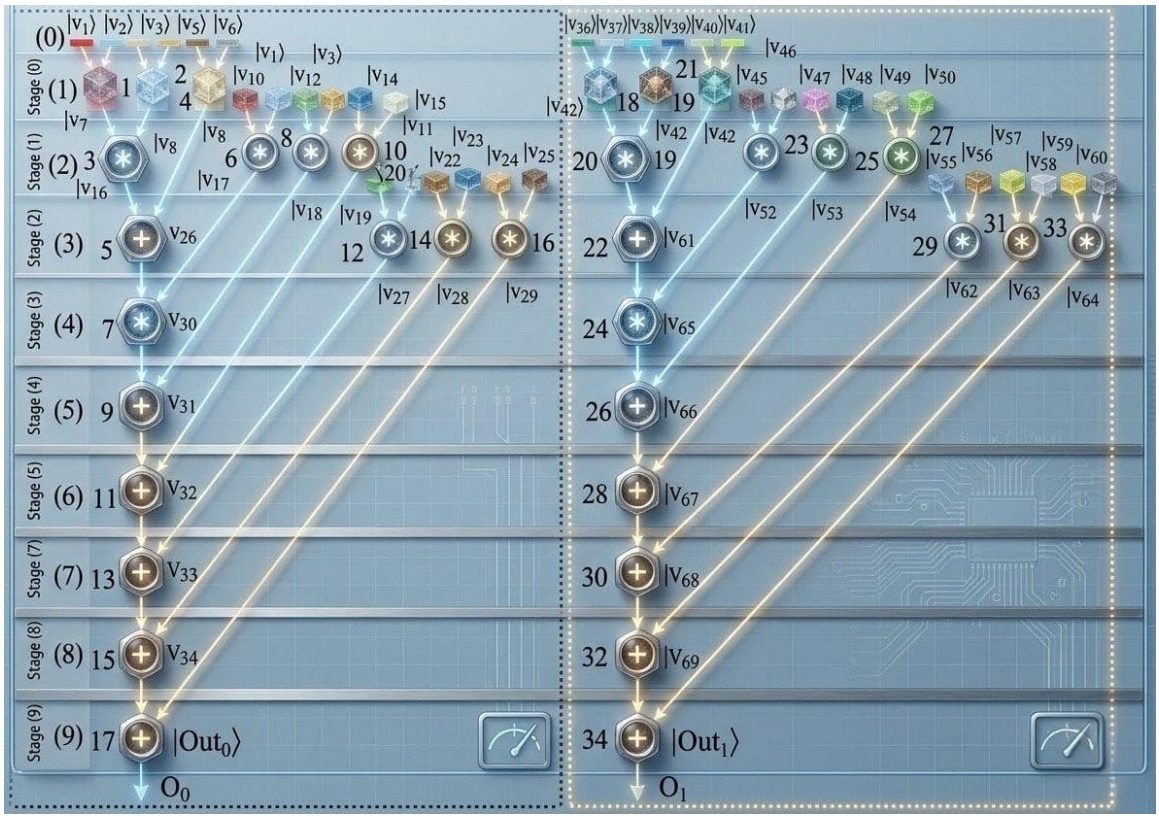


Fig 4.4 Scheduled data flow graph of loop unrolled CNN using 6 M and 2 A

Fig. 4.4 shows the Scheduled data flow graph of the loop-unrolled CNN convolution layer showing operation-level scheduling across control steps using six multipliers (M) and two adders (A). The graph illustrates task dependencies, parallel execution of arithmetic operations, and register utilization for efficient high-level synthesis

The scheduled data flow graph for a loop-unrolled CNN, with an unrolling factor (UF) of 2, is characterized by the use of 6 multipliers (M) and 2 adders (A). To schedule the 34 operations outlined, nine control steps are necessary. The diagram shows that 36 registers distinguished by various colors are required to store the variables Vn (ranging from V1, V2 up to V70) [7], [16], [81].

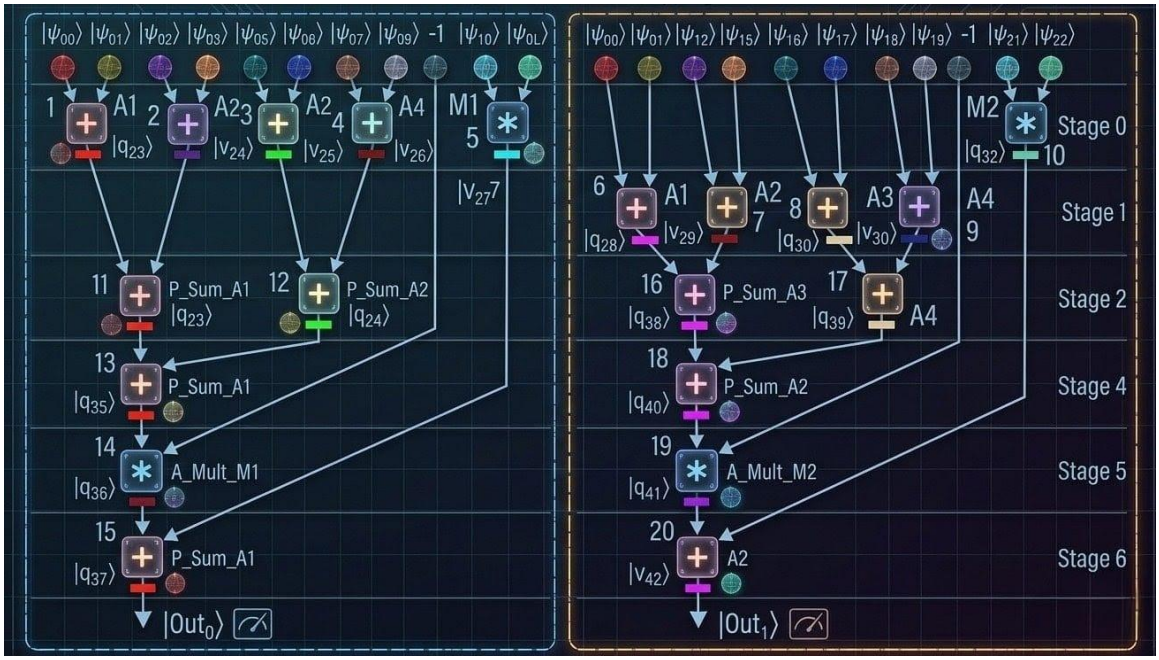


Fig. 4.5 Scheduled data flow graph for sharpening filter using 2M and 4A

Fig. 4.5 Scheduled data flow graph of the image sharpening filter illustrating operation scheduling across control steps with two multipliers (M) and four adders (A). The graph highlights computational dependencies, parallel arithmetic execution, and register allocation for efficient hardware realization through high-level synthesis.

The scheduled data flow graph for an image processing sharpening filter is designed with UF of 2. It operates under the resource constraints of 2M and 4A. To arrange the execution of 20 operations, a total of 7 control steps is necessary. Additionally, 22 registers, each identified by a unique color, are needed to hold the variables Vn, which vary from V0, V1 up to V42.

Based on the scheduled CNN and sharpening image processing filter obtained, its corresponding RAT is created. This phase indicates the register allocation phase of the HLS design process. Subsequently, the IP vendor's encoded encrypted genomic sequence is embedded into this RAT of CNN and sharpening image filter [17].

The encrypted genomic sequence is converted into its respective hardware security constraints using encoding rule-2. According to encoding rule-2, a '0' bit indicates the insertion of an artificial edge between pairs of storage variable nodes labelled as even-even within both the CNN and the sharpening filter. On the other hand, a '1' bit indicates the insertion of an artificial edge between pairs of storage variable nodes marked as odd-odd

within both systems. These constraints are subsequently integrated into the convolution layer of CNN and the image processing sharpening filter, following the encoding rule-2 defined by the IP vendor [81].

For instance, the hardware security constraints derived from the 128-bit encrypted genomic sequence corresponding to the sharpening filter IP is implemented using this encoding rule-2. The generated security constraints corresponding to CNN and sharpening filter for 128 bits encrypted signature are used. The register RAT pre and post the implementation of embedded security constraints for convolution layer CNN are outlined. Following the incorporation of the designed security measures, any conflicts arising between registers are addressed by either locally adjusting the involved registers or by introducing a new register [1], [17], [18].

CNN convolutional layer IP																										
'0' bits	(V_2,V_4)	(V_2,V_6)	(V_2,V_8)	(V_2,V_{10})	(V_2,V_{12})	(V_2,V_{14})	(V_2,V_{16})	(V_2,V_{18})	(V_2,V_{20})	(V_2,V_{22})	.....	(V_4,V_{44})	(V_4,V_{46})	(V_4,V_{48})	(V_4,V_{50})	(V_4,V_{52})	(V_4,V_{54})	(V_4,V_{56})	(V_4,V_{58})	(V_4,V_{60})	(V_4,V_{62})
'1' bits	(V_1,V_3)	(V_1,V_5)	(V_1,V_7)	(V_1,V_9)	(V_1,V_{11})	(V_1,V_{13})	(V_1,V_{15})	(V_1,V_{17})	(V_1,V_{19})	(V_1,V_{21})	.....	(V_3,V_{49})	(V_3,V_{51})	(V_3,V_{53})	(V_3,V_{55})	(V_3,V_{57})	(V_3,V_{59})	(V_3,V_{61})	(V_3,V_{63})	(V_3,V_{65})	(V_3,V_{67})
Sharpening Filter IP																					
'0' bits	(V_2,V_4)	(V_2,V_6)	(V_2,V_8)	(V_2,V_{10})	(V_2,V_{12})	(V_2,V_{14})	(V_2,V_{16})	(V_2,V_{18})	(V_2,V_{20})	(V_2,V_{22})	.....	(V_6,V_{42})	(V_8,V_{10})	(V_8,V_{12})	(V_8,V_{14})	(V_8,V_{16})	(V_8,V_{18})	(V_8,V_{20})	(V_8,V_{16})	(V_8,V_{18})	(V_8,V_{20})
'1' bits	(V_1,V_3)	(V_1,V_5)	(V_1,V_7)	(V_1,V_9)	(V_1,V_{11})	(V_1,V_{13})	(V_1,V_{15})	(V_1,V_{17})	(V_1,V_{19})	(V_1,V_{21})	.....	(V_7,V_7)	(V_7,V_9)	(V_7,V_{11})	(V_7,V_{13})	(V_7,V_{15})	(V_7,V_{17})	(V_7,V_{19})	(V_7,V_{21})	(V_7,V_{23})	(V_7,V_{25})

Table 4.1 Encoded hardware security constraints mapped onto register pairs for watermark embedding during register allocation.

Table 4.1 explain the Encoded hardware security constraints derived from the Feistel-cipher-encrypted genomic signature and mapped onto register pairs of the CNN convolution layer and sharpening filter. The constraints specify even–even ('0' bit) and odd–odd ('1' bit) register associations used for watermark embedding during register allocation.

C.S.																																
0	V1	V2	V3	V4	V5	V6												V36	V37	V38	V39	V40	V41									
1	V7	V7	V8		V9		V10	V11	V12	V13	V14	V15						V42		V43		V44		V45	V46	V47	V48	V49	V50				
2	V16				V9		V17		V18		V19		V20	V21	V22	V23	V24	V25	V51			V44		V52		V53		V54		V55	V56	V57	V58	V59	V60
3	V26						V17		V18		V19		V27		V28		V29		V61					V52		V53		V54		V62		V63		V64	
4	V30								V18		V19		V27		V28		V29		V65							V53		V54		V62		V63		V64	
5	V31	V31									V19		V27		V28		V29		V66									V54		V62		V63		V64	
6	V32												V27		V28		V29		V67											V62		V63		V64	
7	V33	V33													V28		V29		V68													V63		V64	
8	V34																V29		V69															V64	
9	V35	V35																	V70																

Table 4.2 Comparison of the register allocation table (RAT) for the CNN convolution layer before and after embedding hardware security constraints.

Table 4.2 shows the Comparison of the Register Allocation Table (RAT) for the CNN convolution layer before and after embedding hardware security constraints derived from the encrypted genomic signature. The post-embedding RAT reflects register reassignments performed to satisfy watermark constraints during high-level synthesis.

4.5 Verification and Validation

The corroboration of the authentic IP core is performed by matching the embedded watermarking constraints with the regenerated encrypted genomic signature [16], [18], [48]. In case of accurate matching, the IP under test is confirmed genuine.

To detect IP piracy, the genuine IP vendor matches the original genomic signature-based watermark with the extracted watermark of the suspected chip. During IP ownership conflict, the ownership is awarded to the genuine IP vendor as he/she is able to corroborate the ownership by effectively decrypting the genomic signature to obtain the covert watermark constraints using the IP vendor encryption key and matching the bit-wise positions of the regenerated genomic signature with the inserted watermark signature [42], [48], [51].

4.6 Summary

This paper proposes Gen-Sign, a secure and low-overhead hardware intellectual property (IP) watermarking methodology designed to protect convolutional neural network (CNN) accelerators and image processing filter cores against piracy and unauthorized duplication [8], [17], [72]. As hardware accelerators become widely deployed in consumer electronics and embedded systems, protecting proprietary designs from reverse engineering, cloning, and malicious modification has become a major concern. Traditional watermarking approaches rely on physical layout changes, synthesis directives, or structural signatures, which may either introduce design overhead or provide limited security. To overcome these limitations, the authors introduce a novel biometric-driven watermarking strategy that leverages the IP vendor's genomic sequence as a unique and covert signature, thereby offering stronger ownership proof and higher resistance to attacks [5], [16]. The proposed method begins by extracting a genomic base-pair sequence composed of Adenine (A), Thymine (T), Guanine (G), Cytosine (C), and the polynucleotide backbone [5]. This biological information is digitally encoded using a predefined encoding rule to generate a binary genomic signature. Since DNA sequences are inherently unique and difficult to replicate, the generated signature provides a highly distinctive watermark source. To further strengthen protection, the binary sequence is encrypted using a multi-round Feistel cipher. The encryption process divides the data into left and right halves and performs

iterative permutation, substitution, and XOR-based key mixing operations. Vendor-specific keys are generated dynamically for each round, ensuring strong confusion and diffusion properties [36]. This step produces an encrypted genomic signature that is resistant to reverse engineering and cryptanalysis. After encryption, the watermark is converted into hardware security constraints through a second encoding mechanism. Each bit of the encrypted signature is mapped to artificial dependencies between register pairs during the register allocation phase of high-level synthesis. Specifically, '0' bits create even–even register constraints and '1' bits create odd–odd constraints. These dependencies are embedded into the Register Allocation Table (RAT), allowing the watermark to be inserted invisibly without affecting functional correctness. Because the watermark is integrated within scheduling and allocation rather than extra hardware logic, the design maintains identical performance and resource usage. The methodology is validated using two representative case studies: a loop-unrolled CNN convolution layer and an image sharpening filter. Scheduled Data Flow Graphs (SDFGs) are constructed under defined resource constraints to enable efficient operation scheduling. The encrypted genomic constraints are embedded into the corresponding RATs, and any conflicts are resolved through register reassignment. Experimental results show that the proposed technique introduces zero area and latency overhead, demonstrating that security is achieved without compromising efficiency. Security evaluation is conducted using metrics such as tamper tolerance, probability of coincidence, and entropy. The Gen-Sign approach achieves significantly higher tamper tolerance due to its large signature space, making brute-force removal extremely difficult. It also reports very low probability of coincidence, minimizing false ownership claims, and high entropy, indicating strong randomness and unpredictability. Compared to existing watermarking techniques, the proposed framework provides superior robustness, stronger security guarantees, and broader compatibility with both academic and commercial HLS tools [1], [48], [51]. Overall, Gen-Sign integrates biometric uniqueness, cryptographic encryption, and HLS-based constraint embedding to deliver a practical and highly secure watermarking solution. The framework effectively detects IP piracy, verifies ownership, and protects hardware accelerators while maintaining zero design overhead, making it a reliable and scalable approach for modern hardware IP protection [18], [20].

Chapter 5
Results and Analysis

5.1 Evaluation Metrics

Before presenting the experimental results of the proposed hardware security methodologies, it is essential to clearly define and mathematically explain the evaluation metrics used to assess robustness, credibility, and implementation efficiency. These metrics are widely accepted in hardware watermarking literature for validating ownership proof and resistance against attacks [33], [34], [41], [44],[45].

The following four primary metrics are considered:

1. Probability of Coincidence (IC)
2. Tamper Tolerance (LO)
3. Entropy (EN)
4. Design Overhead (Area and Latency)

Each metric is explained in detail below.

5.1.1 Probability of Coincidence (IC)

Probability of Coincidence measures the likelihood that two independent watermark generation processes accidentally produce identical security constraints [18], [44]. It represents the probability of **false ownership validation**.

Mathematical Expression

$$IC = \left(1-\frac{1}{b}\right)^{s}$$

Where:

- b= number of available storage/register variables
- s= number of embedded watermark constraints

Interpretation

- As the number of constraints $s$increases, IC decreases exponentially.
- A smaller IC value indicates stronger ownership proof.
- Lower IC means lower probability of ghost signature attack [48].

Significance: In hardware IP protection, false positive claims must be statistically impossible [2], [6]. If IC is large, an attacker may claim accidental watermark match. Therefore, lower IC depicts stronger legal credibility and strong protection against ghost signature search attacks.

5.1.2 Tamper Tolerance (TT)

Definition

Tamper Tolerance represents the total number of possible watermark combinations that an attacker must search to remove or alter the watermark [1], [17], [44], [45]. It also quantifies resistance against brute-force tampering.

Mathematical Expression

$$LO = v^s$$

Where:

- v= number of encoding variables
- s= number of embedded constraints

Interpretation

- LO grows exponentially with constraint count.
- A higher LO indicates larger search space.
- Extremely high LO makes brute-force attacks infeasible [12], [29], [50].

Practical Meaning

If LO = 10^{77}, even testing one billion combinations per second would require:

$$10^{77}/10^{9} = 10^{68} \text{ seconds}$$

Which exceeds computational feasibility. Thus, higher LO means strong anti-tampering capability.

5.1.3 Entropy (EN)

Entropy measures the unpredictability and randomness of the generated signature [5], [16]. It reflects how difficult it is for an attacker to reconstruct or guess the watermark [44], [54], [70].

Mathematical Form

For biometric-fused encryption:

$$EN = \frac{1}{2^{w} \cdot m! \cdot n! \cdot 2^{k}}$$

Where:

- w= signature length
- m= facial features
- n= amino acid combinations
- k= encryption key size

Interpretation

- Lower EN value means stronger randomness.
- Extremely small entropy probability indicates unpredictability.
- High entropy ensures anti-forgery robustness [5], [61].

5.1.4 Design Overhead

Design overhead evaluates:

- Area increase (μm^2)
- Latency increase (ps)
- Additional hardware resources

Importance

Security mechanisms must not degrade:

- Performance
- Power efficiency
- Silicon cost

Thus: minimal or 0% overhead is ideal industrial applicability [18], [44], [61].

5.2 Results and Analysis of paper-1 (Biometric–AES Fused Signature Method)

5.2.1 Probability of Coincidence Analysis

The biometric-AES framework embeds up to 256 covert security constraints during register allocation [1], [17], [23]. Because the probability of coincidence decreases exponentially with the number of constraints, the resulting IC values become extremely small [20].

For the image filter benchmarks, IC values are observed in the order of:

$$10^{-6} \text{ to } 10^{-5}$$

Benchmarks	Proposed approach		[32]		[11]		[8]	
	Embedded security constraints	(I_c)	Embedded security constraints	(I_c)	Embedded security constraints	(I_c)	Embedded security constraints	(I_c)
Blur Filter	256	1.98E-06	128	1.40E-03	81	1.56E-02	128	1.40E-03
Sharpening Filter	256	6.72E-06	128	2.59E-03	81	2.30E-02	128	2.59E-03
LED Filter	136	4.19E-05	128	7.59E-05	81	2.47E-03	128	7.59E-05
JPEG-CODEC	256	1.53E-01	128	3.91E-01	81	5.52E-01	128	3.91E-01

Benchmarks	Proposed approach		[15]		[34]		[66]	
	Embedded security constraints	(I_c)	Embedded security constraints	(I_c)	Embedded security constraints	(I_c)	Embedded security constraints	(I_c)
Blur Filter (BF)	256	1.98E-06	160	2.72E-04	240	4.50E-06	235	5.82E-06
Sharpening Filter (SF)	256	6.72E-06	160	5.85E-04	240	1.41E-05	235	1.78E-05
Laplacian (LED) Filter	136	4.19E-05	136	4.19E-05	136	4.19E-05	136	4.19E-05
JPEG-CODEC	256	1.53E-01	160	3.09E-01	240	1.72E-01	235	1.78E-01

Table 5.1. Comparison of the Probability of Coincidence (IC) in the Proposed Approach

For larger designs such as JPEG-CODEC, the IC slightly increases but still remains statistically low [16].

Interpretation

This indicates that:

- accidental match probability is nearly zero
- watermark uniqueness is extremely high
- ownership proof is highly credible

Security reasoning

Since each signature is generated using:

- amino acid biometrics
- facial anthropometric measurements
- secure fingerprint
- AES encryption

the likelihood that another IP designer produces identical constraints becomes negligible.

Hence, ghost signature attacks are practically impossible.

5.2.2 Tamper Tolerance Analysis

Benchmarks	Proposed approach	[32]	[11]	[8]
Blur Filter (BF)	1.15E+77	3.40E+38	2.41E+24	3.40E+38
Sharpening Filter (SF)	1.15E+77	3.40E+38	2.41E+24	3.40E+38
Laplace Edge Detection (LED) Filter	8.71E+40	3.40E+38	2.41E+24	3.40E+38
JPEG-CODEC	1.15E+77	3.40E+38	2.41E+24	3.40E+38

Table 5.2 Comparison of tamper tolerance (LO) between the proposed approach

Benchmarks	Proposed approach	[15]	[34]
Blur Filter (BF)	1.15E+77	1.46E+48	1.76E+72
Sharpening Filter (SF)	1.15E+77	1.46E+48	1.76E+72
Laplace Edge Detection (LED) Filter	8.71E+40	8.71E+40	8.71E+40
JPEG-CODEC	1.15E+77	1.46E+48	1.76E+72

Table 5.3 Comparison of tamper tolerance (LO) between the proposed approach

Observations

Tamper tolerance reaches:

$$10^{77}$$

Interpretation

This value represents an astronomically large search space. Even with extremely powerful computational systems, enumerating such a large number of combinations is infeasible [29].

Practical reasoning

If an attacker attempts:

- 1 billion guesses per second

Time required:

$$10^{68} \text{ seconds}$$

This exceeds the lifetime of the universe.

Security implication

Therefore:

- brute-force removal is infeasible
- embedded watermark remains permanent
- design modification attempts become detectable

5.2.3 Entropy Analysis

Benchmarks	Entropy
Proposed approach	2.06E-138
[34]	2.01E-87
[15]	8.63E-78
[32]	1.66E-111
[11]	1.03E-32
[8]	2.9E-39

Table 5.4 Comparison of entropy between the proposed approach

Observations

Entropy is approximately:

$$2.06 \times 10^{-138}$$

Interpretation

This extremely small probability indicates exceptionally strong randomness [44], [54], [55].

Source of high entropy

Entropy arises from multiple independent components:

1. Amino acid chain randomness
2. Facial anthropometric variations
3. Non-invertible fingerprint minutiae transformation
4. AES cryptographic diffusion

These components multiply together, producing massive unpredictability.

Security implication

Thus:

- forging signature is infeasible
- reconstruction without secret key impossible
- extremely strong anti-replication property

5.2.4 Design Overhead Analysis

Benchmarks	IP seller selected resource configuration for scheduling	Initial design (*i.e.*, pre-signature embedded unsecured design)			Security constraints embedded secured design			Register count (pre and post embedding)	Design cost overhead %
		Area (um^2)	Latency (ps)	Design cost	Area (um^2)	Latency (ps)	Design cost		
Blur Filter (BF)	3(+), 1(*)	147.84	927.39	0.537	147.84	927.39	0.537	20	0
Sharpening Filter	4(+), 2(*)	243.79	794.91	0.588	243.79	794.91	0.588	22	0
LED Filter	2(+), 2(*)	199.75	728.67	0.71	199.75	728.67	0.71	14	0
JPEG-CODEC	6(+), 8(*)	824.96	3245.89	0.157	824.96	3245.89	0.157	137	0

Table 5.5 Design cost, area, and latency of the proposed approach before and after embedding security constraints

Observations

Measured results show:

- no increase in area
- no increase in latency
- no extra hardware

Reason

Embedding is done during:

register allocation phase [18], [33]

No additional logic is introduced.

Practical significance

This ensures:

- no cost penalty
- no performance degradation
- high industrial feasibility

5.2.5 Summary of Paper-1 Results

The biometric-AES approach demonstrates:

- extremely low coincidence probability
- very high tamper tolerance
- strong entropy
- zero hardware overhead

Thus ensuring robust and practical hardware IP protection [17], [61].

5.3 Results and Analysis of paper-2

Gen-Sign Feistel Encrypted Genomic Method

5.3.1 Probability of Coincidence Analysis

Benchmarks	CNN	Sharpening Filter	Blur Filter	LED Filter
Proposed Gen-Sign	2.71E-02	2.59E-03	1.40E-03	7.59E-05
Physical-Level Watermarking [76]	7.98E-01	6.89E-01	6.63E-01	5.52E-01
Multilevel Watermark [77]	6.74E-01	5.21E-01	4.87E-01	3.54E-01
Facial Biometric [7]	9.65E-02	2.10E-02	1.41E-02	2.10E-03
Automated Watermark Methodology [78]	1.84E-01	6.13E-02	4.60E-02	1.17E-02

Table 5.6 comparison of probability of coincidence with previous works

Observations: Gen-Sign embeds a 128-bit encrypted genomic signature converted into numerous covert constraints. Due to this large constraint set, the probability of coincidence becomes extremely small [48].

Interpretation

Because each genomic signature originates from:

- unique base-pair sequence
- custom encoding rule
- Feistel cipher encryption

the chance of two identical signatures occurring accidentally is negligible.

Security implication

Hence:

- ownership proof remains statistically reliable
- ghost search attack becomes infeasible

5.3.2 Tamper Tolerance Analysis

Benchmarks	CNN	Sharpening Filter	Blur Filter	LED Filter
Proposed Gen-Sign	3.40E+38	3.40E+38	3.40E+38	3.40E+38
Physical-Level Watermarking [76]	256	256	256	256
Multilevel Watermark [77]	16384	16384	16384	16384
Facial Biometric [7]	9.67E+24	9.67E+24	9.67E+24	9.67E+24
Automated watermark Methodology [78]	1.15E+18	1.15E+18	1.15E+18	1.15E+18

Table 5.7 comparison of Temper Tolerance with previous works

Observations

The encryption combined with large constraint space produces extremely large tamper tolerance [23], [61].

Interpretation

An attacker would have to test enormous combinations to modify constraints without breaking functionality [29], [50].

Security implication

Thus:

- brute-force removal infeasible
- constraints persistent
- watermark remains intact

5.3.3 Entropy Analysis

Benchmarks	Probability of Watermark Decoding Attack
Proposed Gen-Sign	2.71E-02
Physical-Level Watermarking [76]	7.98E-01
Multilevel Watermark [77]	6.74E-01
Facial Biometric [7]	9.65E-02
Automated Watermark Methodology [78]	1.84E-01

Table 5.8 comparison of entropy with previous works

Observations

High entropy results from:

- genomic sequence variability
- 100 Feistel rounds
- round-wise key generation
- nonlinear substitution and permutation

Interpretation

Each encryption round increases diffusion and confusion, strengthening randomness.

Security implication

Therefore:

- signature unpredictable
- reverse engineering extremely difficult
- forgery resistance strong

5.3.3 Design Overhead Analysis

Benchmark	Register Count	Baseline Design Area (µm²)	Baseline Design Latency (ps)	Baseline Design Cost	Embedded Design Area (µm²)	Embedded Design Latency (ps)	Embedded Design Cost	Design Cost Overhead
CNN	36	625.11	1192.36	0.33	625.11	1192.36	0.33	0%
Sharpening Filter	22	243.62	794.9	0.58	243.62	794.9	0.58	0%
Blur Filter	20	109.96	1523.54	0.83	109.96	1523.54	0.83	0%
LED Filter	14	199.64	728.66	0.68	199.64	728.66	0.68	0%

Table 5.9 overhead analysis of proposed gen-sign

Observations

Similar to the first method:

- no area increase
- no delay penalty
- no extra hardware

Reason

Constraints embedded within HLS scheduling and RAT.

Practical significance

Ensures cost-free security.

Chapter 6

6.1 Conclusion and Future Scope

The increasing globalization of semiconductor design supply chains has significantly exposed reusable hardware intellectual property (IP) cores to threats such as piracy, unauthorized reuse, and false ownership claims. Conventional watermarking and steganographic approaches often fail to provide strong authorship credibility and tamper resistance when confronted with sophisticated adversaries capable of reverse engineering or structural manipulation. To address these challenges, this work introduced two biologically inspired hardware security frameworks: Biometric–AES fused signature watermarking and Gen-Sign Feistel encrypted genomic watermarking, both embedded during the high-level synthesis stage to provide robust and covert IP protection.

The first approach integrates multimodal human biometrics—including amino acid sequences, facial anthropometric features, and fingerprint minutiae—with AES cryptographic encryption to generate a secure fused signature. By combining multiple biometric sources, the framework significantly increases entropy and establishes a natural and legally defensible connection between the IP core and its rightful owner. The encryption process further strengthens unpredictability, ensuring that even minor variations in biometric inputs produce entirely different signatures. Experimental evaluation demonstrates extremely low probability of coincidence, strong resistance to ghost-signature attacks, and high tamper tolerance due to the large number of covert constraints generated during register allocation. Importantly, the watermark is embedded without introducing additional hardware overhead, preserving silicon area, power consumption, and performance.

The second approach, Gen-Sign, explores genomic information as a novel signature source for hardware watermarking. In this method, encoded genomic sequences are encrypted using a multi-round Feistel cipher to produce highly randomized signatures that are embedded into register allocation constraints. The combination of biological diversity and cryptographic diffusion significantly enhances unpredictability and resistance to reverse engineering. Similar to the biometric-AES method, Gen-Sign achieves strong authorship credibility, minimal coincidence probability, and high tamper persistence while maintaining zero hardware overhead. The coupling between watermark constraints and functional datapath structures further discourages malicious removal attempts, ensuring reliable ownership verification even after design modifications.

Collectively, the results of this research demonstrate that biologically inspired signatures combined with cryptographic protection provide a powerful and practical solution for hardware IP protection. Both frameworks effectively address key challenges of authorship authentication, watermark robustness, tamper tolerance, and implementation efficiency. By embedding covert security constraints during high-level synthesis, the proposed techniques maintain complete transparency in hardware cost while significantly strengthening protection against piracy and false ownership claims.

Despite these promising outcomes, several avenues remain open for future exploration. One potential direction involves incorporating additional biometric modalities such as iris patterns, voice characteristics, or electrocardiogram signals to further increase signature diversity and entropy. Expanding genomic watermarking to include other biological encodings such as proteomic or epigenetic patterns may also broaden the signature space and enhance uniqueness.

Future research could also focus on developing automated pipelines for biometric feature extraction, signature generation, and encryption, enabling seamless integration with industrial HLS toolchains. Extending the proposed watermarking strategies to FPGA-based and reconfigurable platforms would expand their applicability to embedded systems, IoT devices, and edge computing architectures.

Another promising direction involves the use of machine learning techniques for adaptive watermark placement, allowing intelligent selection of embedding locations that maximize stealthiness and resilience against structural attacks. Additionally, exploring post-quantum cryptographic primitives as alternatives to classical encryption algorithms may help future-proof the watermarking framework against emerging quantum threats.

Further studies may also examine the legal and forensic implications of biometric and genomic hardware signatures, including standardized procedures for watermark extraction, validation, and courtroom admissibility. Finally, large-scale evaluations on complex system-on-chip designs containing numerous IP blocks will be essential to validate scalability and facilitate real-world industrial adoption.

In summary, the integration of biometric and genomic signatures with cryptographic watermarking techniques establishes a new paradigm for secure hardware IP protection. With further advancements in automation, cryptography, and large-scale integration, these biologically inspired methodologies have the potential to become a cornerstone of next-generation hardware security frameworks.

REFERENCES

[1] A. Sengupta, N. Bhui and A. Anshul, "High-Level Synthesis-Based Forensic Watermarking of Hardware IPs Using IP Vendor's DNA Signature," in IEEE Embedded Systems Letters, vol. 18, no. 1, pp. 19-22, Feb. 2026.

[2] A. Sengupta and A. Anshul, "A Survey of High-Level Synthesis-Based Hardware (IP) Watermarking Approaches," in IEEE Design & Test, vol. 41, no. 6, pp. 70-83, Dec. 2024.

[3] A. Sengupta and R. Chaurasia, "Secure FFT IP Using C-Way Partitioning-Based Obfuscation and Fingerprint," in IEEE Design & Test, vol. 41, no. 5, pp. 55-64, Oct. 2024.

[4] M. Rathor, A. Anshul and A. Sengupta, "Securing Reusable IP Cores Using Voice Biometric Based Watermark," in IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 4, pp. 2735-2749, July-Aug. 2024.

[5] A. Sengupta, R. Chaurasia and A. Anshul, "Robust Security of Hardware Accelerators Using Protein Molecular Biometric Signature and Facial Biometric Encryption Key," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 6, pp. 826-839, June 2023.

[6] A. Anshul and A. Sengupta, "A Survey of High Level Synthesis Based Hardware Security Approaches for Reusable IP Cores [Feature]," in IEEE Circuits and Systems Magazine, vol. 23, no. 4, pp. 44-62, Fourthquarter 2023.

[7] A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," in IEEE Transactions on Consumer Electronics, vol. 68, no. 3, pp. 291-306, Aug. 2022.

[8] A. Sengupta and R. Chaurasia, "Securing IP Cores for DSP Applications Using Structural Obfuscation and Chromosomal DNA Impression," in IEEE Access, vol. 10, pp. 50903-50913, 2022.

[9] R. Chaurasia, A. Anshul, A. Sengupta and S. Gupta, "Palmprint Biometric Versus Encrypted Hash Based Digital Signature for Securing DSP Cores Used in CE Systems," in IEEE Consumer Electronics Magazine, vol. 11, no. 5, pp. 73-80, 1 Sept. 2022.

[10] A. Sengupta, R. Chaurasia and T. Reddy, "Contact-Less Palmprint Biometric for Securing DSP Coprocessors Used in CE Systems," in IEEE Transactions on Consumer Electronics, vol. 67, no. 3, pp. 202-213, Aug. 2021.

[11] A. Sengupta and M. Rathor, "Facial Biometric for Securing Hardware Accelerators," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 1, pp. 112-123, Jan. 2021.

[12] W. Hu, C. -H. Chang, A. Sengupta, S. Bhunia, R. Kastner and H. Li, "An Overview of Hardware Security and Trust: Threats, Countermeasures, and Design Tools," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 40, no. 6, pp. 1010-1038, June 2021.

[13] A. Sengupta and M. Rathor, "Securing Hardware Accelerators for CE Systems Using Biometric Fingerprinting," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 9, pp. 1979-1992, Sept. 2020.

[14] A. Sengupta and M. Rathor, "IP Core Steganography for Protecting DSP Kernels Used in CE Systems," in IEEE Transactions on Consumer Electronics, vol. 65, no. 4, pp. 506-515, Nov. 2019.

[15] A. Sengupta, E. R. Kumar and N. P. Chandra, "Embedding Digital Signature Using Encrypted-Hashing for Protection of DSP Cores in CE," in IEEE Transactions on Consumer Electronics, vol. 65, no. 3, pp. 398-407, Aug. 2019.

[16] A. Sengupta, D. Roy and S. P. Mohanty, "Triple-Phase Watermarking for Reusable IP Core Protection During Architecture Synthesis," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 4, pp. 742-755, April 2018.

[17] A. Sengupta, R. Chaurasia and A. Anshul, "Hardware Security of Digital Image Filter IP Cores against Piracy using IP Seller's Fingerprint Encrypted Amino Acid Biometric Sample," 2023 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), Tianjin, China, 2023.

[18] A. Sengupta, S. Bhadauria and S. P. Mohanty, "Embedding low cost optimal watermark during high level synthesis for reusable IP core protection," 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, 2016.

[19] A. Sengupta and V. K. Mishra, "Swarm Intelligence Driven Simultaneous Adaptive Exploration of Datapath and Loop Unrolling Factor during Area-Performance Tradeoff," 2014 IEEE Computer Society Annual Symposium on VLSI, Tampa, FL, USA, 2014.

[20] A. Sengupta and S. Bhadauria, "Exploring Low Cost Optimal Watermark for Reusable IP Cores During High Level Synthesis," in IEEE Access, vol. 4, pp. 2198-2215, 2016.

[21] T. Yu and Y. Zhu, "A new watermarking method for soft IP protection," 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), Xianning, China, 2011, pp. 3839-3842.

[22] Y. Zhao et al., "Side Channel Security Oriented Evaluation and Protection on Hardware Implementations of Kyber," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 12, pp. 5025-5035, Dec. 2023.

[23] H. R. Mahdiany, A. Hormati and S. M. Fakhraie, "A hardware accelerator for DSP system design," in Proc. ICM, Morocco, 2001, pp. 141-144.

[24] M. Nagata, T. Miki and N. Miura, "Physical Attack Protection Techniques for IC Chip Level Hardware Security," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30, no. 1, pp. 5-14, Jan. 2022.

[25] D. Kim, J. -P. Hong, J. Lee and J. -W. Nam, "High-Speed Light Detection Sensor for Hardware Security in Standard CMOS Technology," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 70, no. 10, pp. 3917-3921, Oct. 2023.

[26] F. Koushanfar et al., "Can EDA combat the rise of electronic counterfeiting?," Design Automation Conference 2012, San Francisco, CA, USA, 2012, pp. 133-138.

[27] D. Zhang, X. Wang, M. T. Rahman and M. Tehranipoor, "An On-Chip Dynamically Obfuscated Wrapper for Protecting Supply Chain Against IP and IC Piracies," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 11, pp. 2456-2469, Nov. 2018.

[28] M. Yasin, J. J. Rajendran, O. Sinanoglu and R. Karri, "On Improving the Security of Logic Locking," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 9, pp. 1411-1424, Sept. 2016.

[29] A. Cui and C. Chang, "Watermarking for IP protection through template substitution at logic synthesis level," in Proc. IEEE ISCAS, New Orleans, LA, 2007, pp. 3687-3690.

[30] A. Cui, C. -H. Chang, S. Tahar and A. T. Abdel-Hamid, "A Robust FSM Watermarking Scheme for IP Protection of Sequential Circuit Design," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, no. 5, pp. 678-690, May 2011.

[31] S. Rizzo, F. Bertini and D. Montesi, "Fine-grain watermarking for intellectual property protection," EURASIP Journal on Information Security, 2019.

[32] F. Koushanfar, I. Hong and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," ACM Trans. Des. Autom. Electron. Syst., vol. 10, pp. 523–545, 2005.

[33] B. L. Gal and L. Bossuet, "Automatic low-cost IP watermarking technique based on output mark insertions," Des. Autom. Embedded Syst., vol. 16, pp. 71–92, 2012.

[34] E. Castillo et al., "Automated Signature Insertion in Combinational Logic Patterns for HDL IP Core Protection," 2008 4th Southern Conference on Programmable Logic, Bariloche, Argentina, 2008, pp. 183-186.

[35] F. Koushanfar, "Hardware Metering: A Survey," in Introduction to Hardware Security and Trust, Springer, New York, 2012.

[36] J. Zhang, Y. Lin, Y. Lyu and G. Qu, "A PUF-FSM Binding Scheme for FPGA IP Protection and Pay-Per-Device Licensing," IEEE Transactions on Information Forensics and Security, vol. 10, no. 6, pp. 1137-1150, June 2015.

[37] D. B. Roy et al., "Combining PUF with RLUTs: A Two-party Pay-per-device IP Licensing Scheme on FPGAs," ACM Trans. Embed. Comput. Syst., vol. 18, 2019.

[38] B. Halak, M. Zwolinski and M. S. Mispan, "Overview of PUF-based hardware security solutions for the internet of things," 2016 IEEE MWSCAS, Abu Dhabi, UAE, 2016.

[39] Yilmaz, Y., Aniello, L. & Halak, B., "ASSURE: A Hardware-Based Security Protocol for Resource-Constrained IoT Systems," J Hardw Syst Secur, vol. 5, 2021.

[40] A. Anshul and A. Sengupta, "Exploration of optimal crypto-chain signature embedded secure JPEG-CODEC hardware IP during high level synthesis," Microprocessors and Microsystems, vol. 102, 2023.

[41] 15 nm open cell library. [Online]. Available: https://si2.org/open-cell-library/.

[42] M. Potkonjak, "Methods and systems for the identification of circuits and circuit designs," USPTO, US7017043B1, 2006.

[43] A. T. Abdel-Hamid, S. Tahar and E. M. Aboulhamid, "A Public-Key Watermarking Technique for IP Designs," DATE, Munich, Germany, 2005.

[44] NIST Computer Security Resource Center, Glossary, https://csrc.nist.gov/glossary/term/entropy.

[45] University of California Santa Barbara Express Group Benchmark. [Online]. Available: http://express.ece.ucsb.edu/benchmark/.

[46] J. Liu et al., "Min-entropy estimation for semiconductor superlattice true random number generators," Sci Rep, vol. 12, 2022.

[47] F. Zhao and X. Tang, "Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction," Pattern Recognition, vol. 40, no. 4, pp. 1270-1281, 2007.

[48] A. S. M. Sohail and P. Bhattacharya, "Detection of Facial Feature Points Using Anthropometric Face Model," Springer, 2008.

[49] National Library of Medicine – National Institutes of Health, https://www.ncbi.nlm.nih.gov/books/NBK26830/.

[50] K. Steendam et al., "Mass spectrometry-based proteomics as a tool to identify biological matrices in forensic science," Int J Legal Med, vol. 127, pp. 287–298, 2013.

[51] N. N. Anandakumar et al., "Rethinking Watermark: Providing Proof of IP Ownership in Modern SoCs," Cryptology ePrint Archive, 2022.

[52] U. Das et al., "PSC-Watermark: Power Side Channel Based IP Watermarking Using Clock Gates," IEEE ETS, 2023.

[53] Z. Ibnat et al., "ActiWate: Adaptive and Design-agnostic Active Watermarking for IP Ownership in Modern SoCs," DAC, 2023.

[54] P. SLPSK et al., "SIGNED: A Challenge-Response Scheme for Electronic Hardware Watermarking," IEEE Transactions on Computers, vol. 72, no. 6, pp. 1763-1777, 2023.

[55] M. Anastasova et al., "Highly Optimized Curve448 and Ed448 design in wolfSSL and Side-Channel Evaluation on Cortex-M4," IEEE DSC, 2023.

[56] M. B. Niasar et al., "Optimized Architectures for Elliptic Curve Cryptography over Curve448," Cryptology ePrint Archive, 2020.

[57] M. Anastasova et al., "Fast Strategies for the Implementation of SIKE Round 3 on ARM Cortex-M4," IEEE TCAS-I, vol. 68, no. 10, pp. 4129-4141, Oct. 2021.

[58] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard," IEEE Transactions on Computers, vol. 59, no. 5, pp. 608-622, May 2010.

[59] B. Koziel et al., "Post-Quantum Cryptography on FPGA Based on Isogenies on Elliptic Curves," IEEE TCAS-I, vol. 64, no. 1, pp. 86-99, Jan. 2017.

[60] M. Bisheh-Niasar et al., "High-Speed NTT-based Polynomial Multiplication Accelerator for Post-Quantum Cryptography," IEEE ARITH, 2021.

[61] M. Mozaffari-Kermani et al., "Systematic Poisoning Attacks on and Defenses for Machine Learning in Healthcare," IEEE JBHI, vol. 19, no. 6, pp. 1893-1905, 2015.

[62] M. M. Kermani et al., "Education and Research Integration of Emerging Multidisciplinary Medical Devices Security," 2016.

[63] A. M. Nia et al., "Energy-Efficient Long-term Continuous Personal Health Monitoring," IEEE TMSC, vol. 1, no. 2, pp. 85-98, 2015.

[64] M. Rathor and G. P. Rathor, "Hard-Sign: A Hardware Watermarking Scheme Using Dated Handwritten Signature," IEEE Design & Test, vol. 41, no. 2, pp. 75-83, 2024.

[65] A. Sengupta and A. Anshul, "Watermarking Hardware IPs Using Design Parameter Driven Encrypted Dispersion Matrix With Eigen Decomposition Based Security Framework," IEEE Access, vol. 12, 2024.

[66] M. Rathor et al., "Exploring Handwritten Signature Image Features for Hardware Security," IEEE TDSC, vol. 20, no. 5, pp. 3687-3698, 2023.

[67] S. K. Khudhair et al., "Secure Reversible Data Hiding Using Block-Wise Histogram Shifting," Electronics, vol. 12, 2023.

[68] A. Khaldi et al., "A Medical Image Watermarking System Based on Redundant Wavelets for Secure Transmission in Telemedicine Applications," Wireless Pers Commun, 2023.

[69] M. M. Sayah et al., "A Blind and High-Capacity Data Hiding Scheme for Medical Information Security," Circuits Syst Signal Process, 2024.

[70] E. Salah et al., "Survey of imperceptible and robust digital audio watermarking systems," Multimed Tools Appl, 2024.

[71] M. M. Sayah et al., "Medical image protection using a data-hiding technique based on integer wavelets," Multimed Tools Appl, 2024.

[72] S. Thomas et al., "A novel image compression method using wavelet coefficients and Huffman coding," Journal of Engineering Research, 2023.

[73] J. Hemalatha et al., "Towards improving the performance of blind image steganalyzer using third-order SPAM features and ensemble classifier," Journal of Information Security and Applications, vol. 76, 2023.

[74] R. Elkhatib et al., "Cryptographic Engineering a Fast and Efficient SIKE in FPGA," ACM TECS, vol. 23, no. 2, 2024.

[75] S. P. G and P. N, "Texture Image Classification with Dilated Convolution Layers," IEEE WiSPNET, Chennai, India, 2023.

[76] A. Sengupta and M. Rathor, "Enhanced Security of DSP Circuits Using Multi-Key Based Structural Obfuscation and Physical-Level Watermarking for Consumer Electronics Systems," IEEE TCE, vol. 66, no. 2, pp. 163-172, 2020.

[77] D. Roy and A. Sengupta, "Multilevel Watermark for Protecting DSP Kernel in CE Systems," IEEE Consumer Electronics Magazine, vol. 8, no. 2, pp. 100-102, Mar. 2019.

[78] A. Sengupta, D. Roy, "Automated low cost scheduling driven watermarking methodology for modern CAD high-level synthesis tools," Advances in Engineering Software, vol. 110, pp. 26-33, 2017.

[79] A. Sengupta and M. Rathor, "Obfuscated Hardware Accelerators for Image Processing Filters - Application Specific and Functionally Reconfigurable Processors," IEEE TCE, vol. 66, no. 4, pp. 386-395, Nov. 2020.

[80] Z. Zhao et al., "Object detection with deep learning: A review," IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[81] M. Arredondo-Velázquez et al., "Review of prominent strategies for mapping CNNs onto embedded systems," IEEE Latin Amer. Trans., vol. 18, no. 5, pp. 971–982, May 2020.

[82] J. Chen and B. C. Schafer, "Watermarking of Behavioral IPs: A Practical Approach," Design, Automation & Test in Europe Conference (DATE), Grenoble, France, 2021, pp. 1266-1271.