A Survey of High-Level Synthesis-Based Hardware (IP) Watermarking Approaches

Published in IEEE Design & Test

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

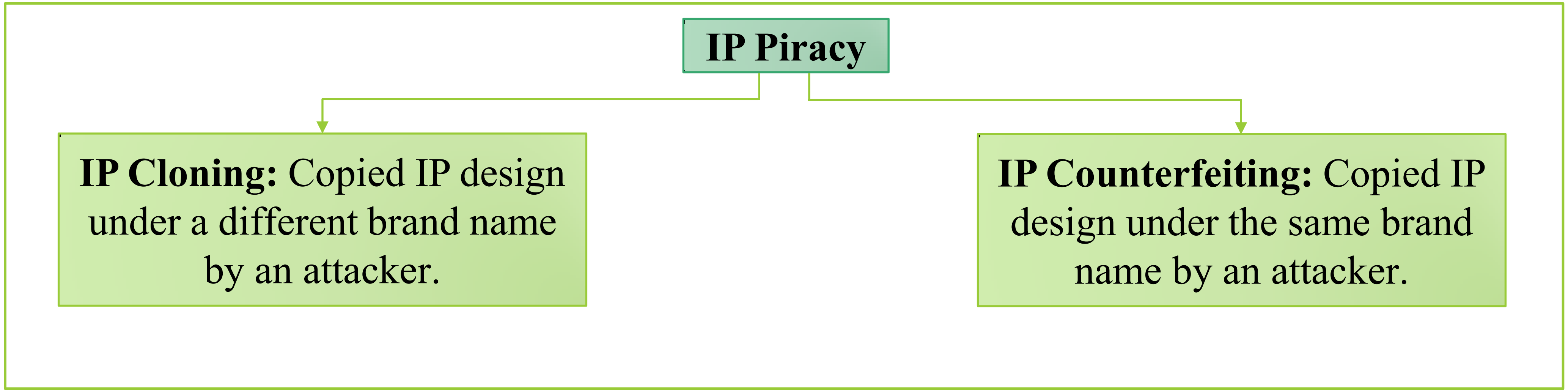
Introduction

- Hardware watermarking is a technique used for embedding hidden information within a hardware design.

- The demand for application-specific hardware systems is rising due to the increasing need for optimized performance, energy efficiency, and tailored solutions in various sectors.

Threat Model

- Securing hardware designs from security threats (such as IP piracy and false IP ownership claims) is crucial in the global supply chain, necessitating robust measures such as hardware watermarking and/or hardware steganography.



IP Piracy

IP Cloning: Copied IP design under a different brand name by an attacker.

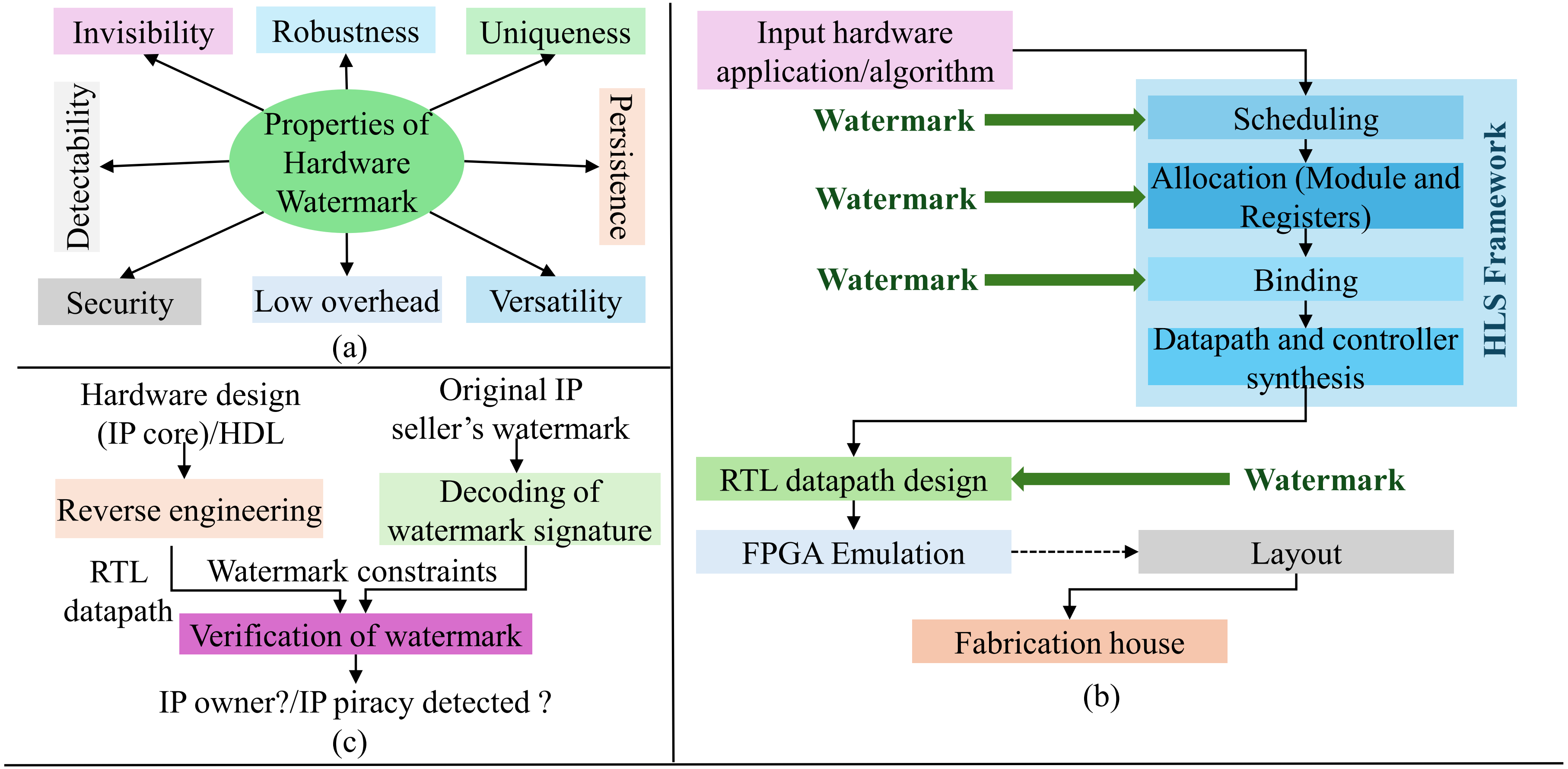
IP Counterfeiting: Copied IP design under the same brand name by an attacker.

• Importance of hardware watermarking

➢ The importance of hardware watermarking in the field of hardware IP core protection includes the following.

- *Protection from IP piracy* – serves as a detective countermeasure.
- *Enhancing design integrity* – making it difficult for malicious actors.
- *Enabling traceability and accountability* – traced back to the original designer or manufacturer, which is essential for accountability.
- *Fostering trust in the market* – enhances trust among stakeholders.

• Properties of hardware watermark



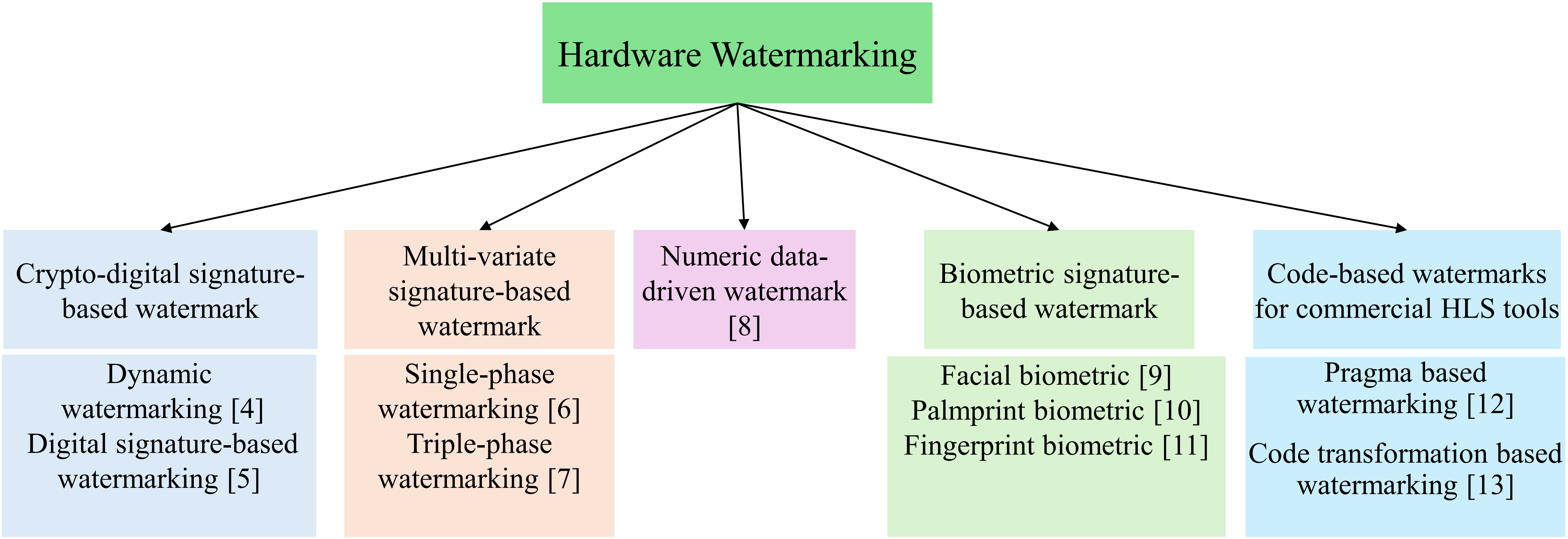
Invisibility
Robustness
Uniqueness
Detectability
Properties of Hardware Watermark
Persistence
Security
Low overhead
Versatility
(a)
Hardware design (IP core)/HDL
Original IP seller's watermark
Reverse engineering
Decoding of watermark signature
RTL datapath
Watermark constraints
Verification of watermark
IP owner?/IP piracy detected ?
(c)
Input hardware application/algorithm
Watermark
Scheduling
Watermark
Allocation (Module and Registers)
Watermark
Binding
Datapath and controller synthesis
HLS Framework
RTL datapath design
Watermark
FPGA Emulation
Layout
Fabrication house
(b)

Fig. 1.(a) Different properties of hardware watermark.

(b) Depiction of possible watermark insertion locations during the hardware design process.

(c) Hardware watermark detection process.

Properties of hardware watermark



Hardware Watermarking

Crypto-digital signature-based watermark

Dynamic watermarking [4]
Digital signature-based watermarking [5]

Multi-variate signature-based watermark

Single-phase watermarking [6]
Triple-phase watermarking [7]

Numeric data-driven watermark [8]

Biometric signature-based watermark

Facial biometric [9]
Palmprint biometric [10]
Fingerprint biometric [11]

Code-based watermarks for commercial HLS tools

Pragma based watermarking [12]
Code transformation based watermarking [13]

Fig. 2. Taxonomy of HLS-based hardware watermarking approaches.

• Taxonomy of HLS-based hardware watermarking approaches

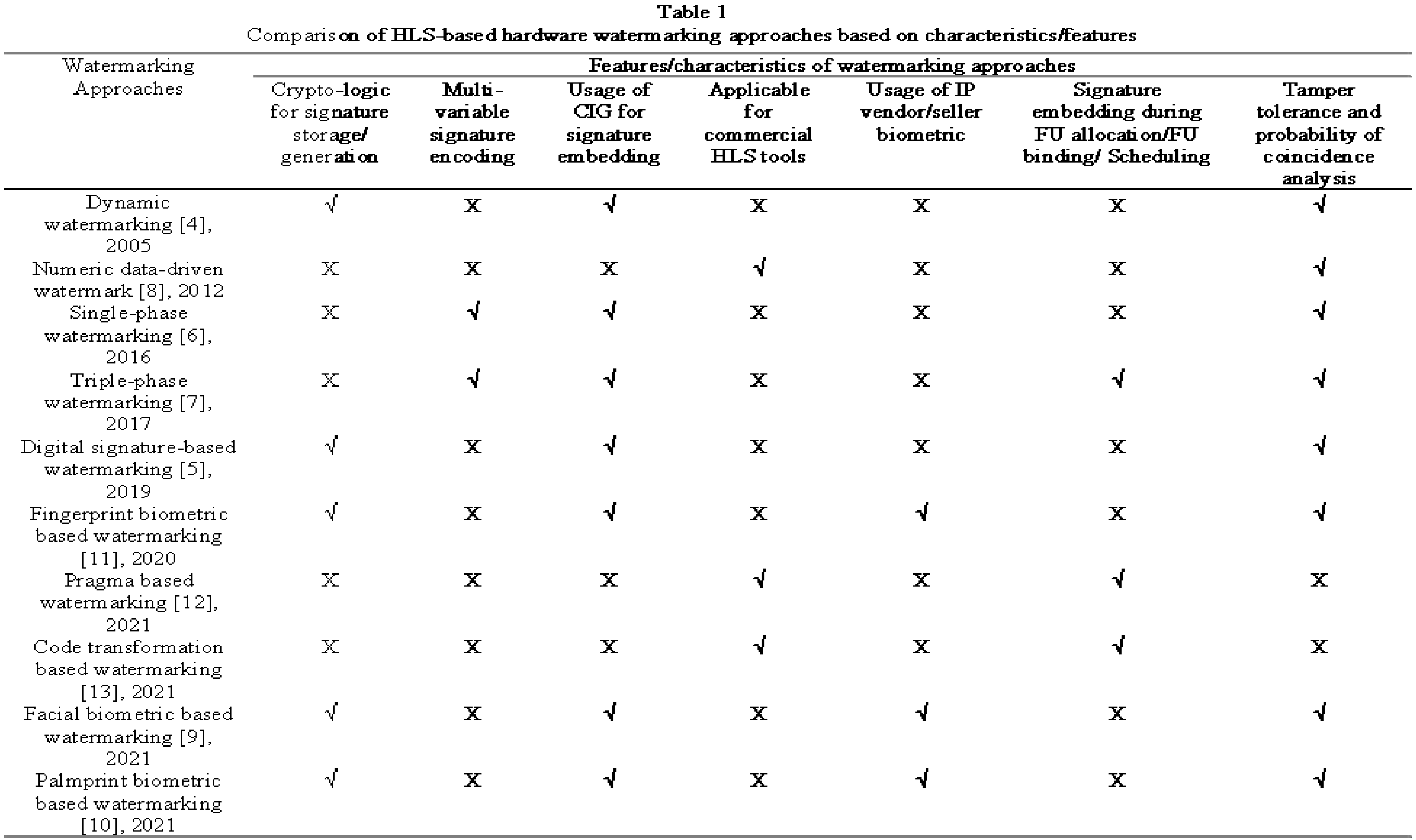


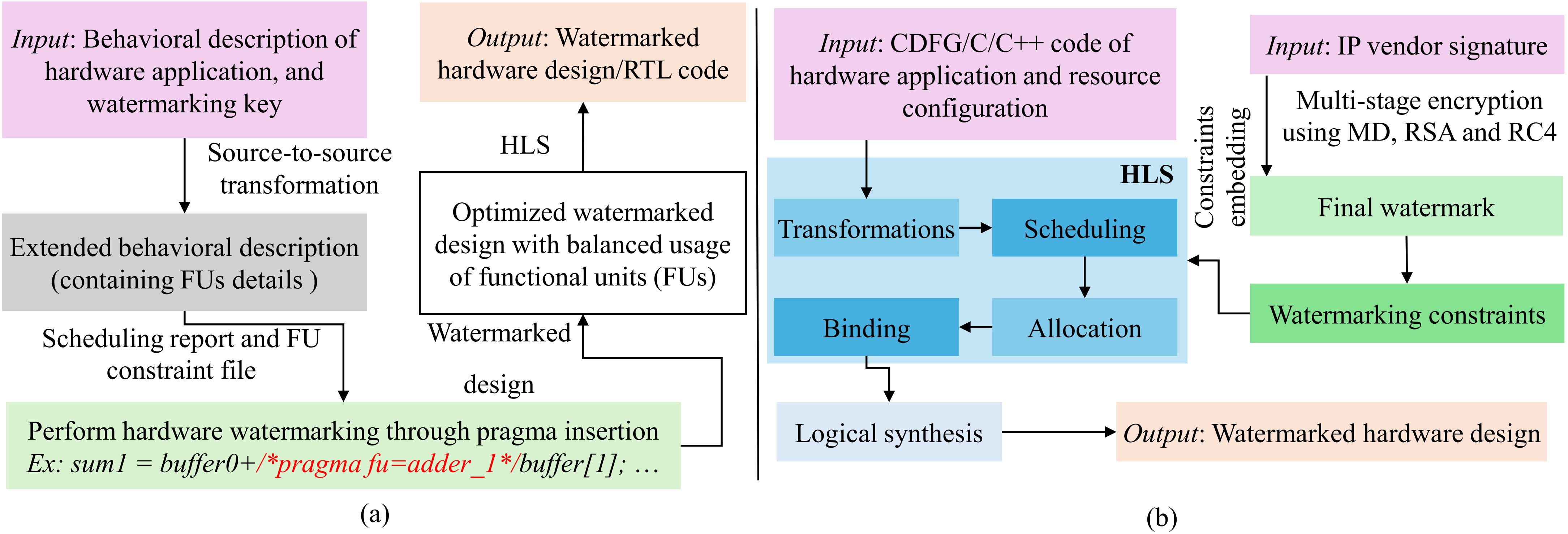
Table 1

Comparison of HLS-based hardware watermarking approaches based on characteristics/features

Watermarking Approaches	Features/characteristics of watermarking approaches						
	Crypto-logic for signature storage/ generation	Multi-variable signature encoding	Usage of CIG for signature embedding	Applicable for commercial HLS tools	Usage of IP vendor/seller biometric	Signature embedding during FU allocation/FU binding/ Scheduling	Tamper tolerance and probability of coincidence analysis
Dynamic watermarking [4], 2005	√	x	√	x	x	x	√
Numeric data-driven watermark [8], 2012	X	x	x	√	x	x	√
Single-phase watermarking [6], 2016	X	√	√	x	x	x	√
Triple-phase watermarking [7], 2017	X	√	√	x	x	√	√
Digital signature-based watermarking [5], 2019	√	x	√	x	x	x	√
Fingerprint biometric based watermarking [11], 2020	√	x	√	x	√	x	√
Pragma based watermarking [12], 2021	X	x	x	√	x	√	x
Code transformation based watermarking [13], 2021	X	x	x	√	x	√	x
Facial biometric based watermarking [9], 2021	√	x	√	x	√	x	√
Palmprint biometric based watermarking [10], 2021	√	x	√	x	√	x	√

• Details of hardware watermarking approaches

- **Key contributions of [4]:** Presents a dynamic hardware watermarking approach using the CIG framework of the HLS process.



Input: Behavioral description of hardware application, and watermarking key

Source-to-source transformation

Extended behavioral description (containing FUs details)

Scheduling report and FU constraint file

Perform hardware watermarking through pragma insertion

*Ex: sum1 = buffer0+/*pragma fu=adder_1*/buffer[1]; …*

Watermarked design

Optimized watermarked design with balanced usage of functional units (FUs)

HLS

Output: Watermarked hardware design/RTL code

Input: CDFG/C/C++ code of hardware application and resource configuration

HLS

Transformations

Scheduling

Allocation

Binding

Logical synthesis

Output: Watermarked hardware design

Input: IP vendor signature

Multi-stage encryption using MD, RSA and RC4

Final watermark

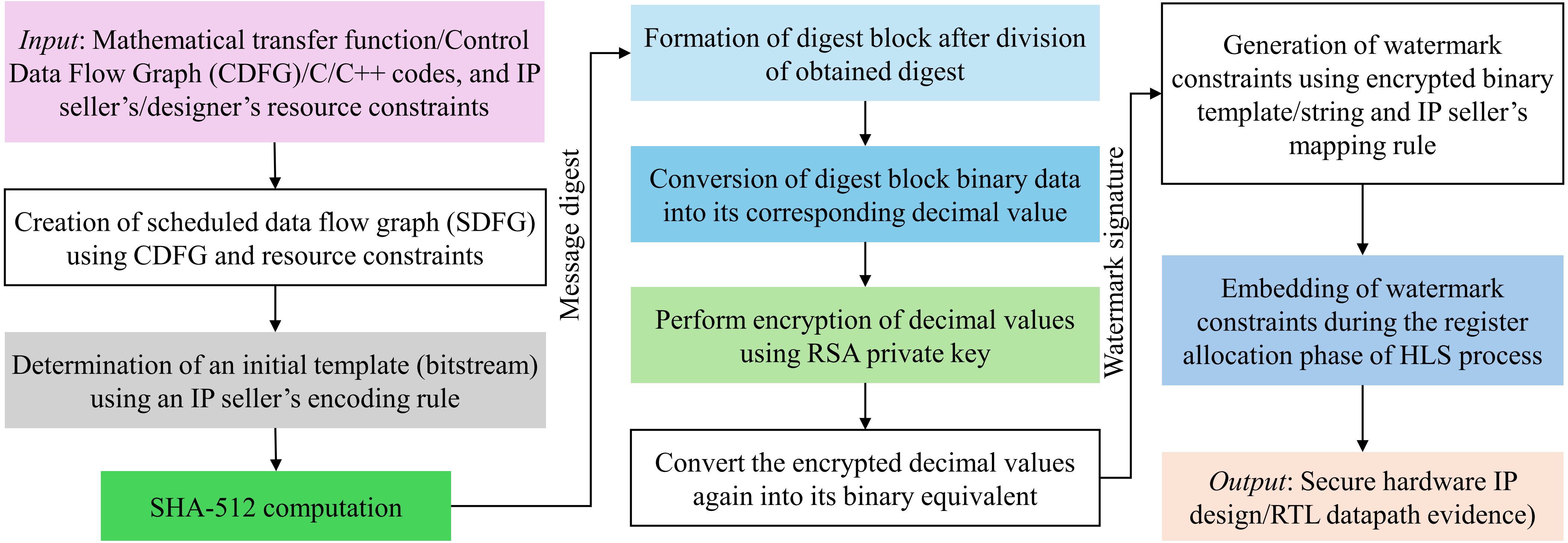
Watermarking constraints

Constraints embedding

Fig. 3. (a) Details of pragma-based hardware watermarking approach [12], (b). Details of dynamic hardware watermarking approach [4]

• Details of hardware watermarking approaches

- **Key contributions of [5]:** Demonstrates the generation of a robust hardware watermark using SHA-512, RSA cryptosystem, and IP seller-selected encoding rule.



Input: Mathematical transfer function/Control Data Flow Graph (CDFG)/C/C++ codes, and IP seller's/designer's resource constraints

Creation of scheduled data flow graph (SDFG) using CDFG and resource constraints

Determination of an initial template (bitstream) using an IP seller's encoding rule

SHA-512 computation

Message digest

Formation of digest block after division of obtained digest

Conversion of digest block binary data into its corresponding decimal value

Perform encryption of decimal values using RSA private key

Convert the encrypted decimal values again into its binary equivalent

Watermark signature

Generation of watermark constraints using encrypted binary template/string and IP seller's mapping rule

Embedding of watermark constraints during the register allocation phase of HLS process

Output: Secure hardware IP design/RTL datapath evidence)

Fig. 4. Details of HLS-digital signature-based watermark generation [5]

• Details of hardware watermarking approaches

- Multivariate signature-based watermark [6], [7]

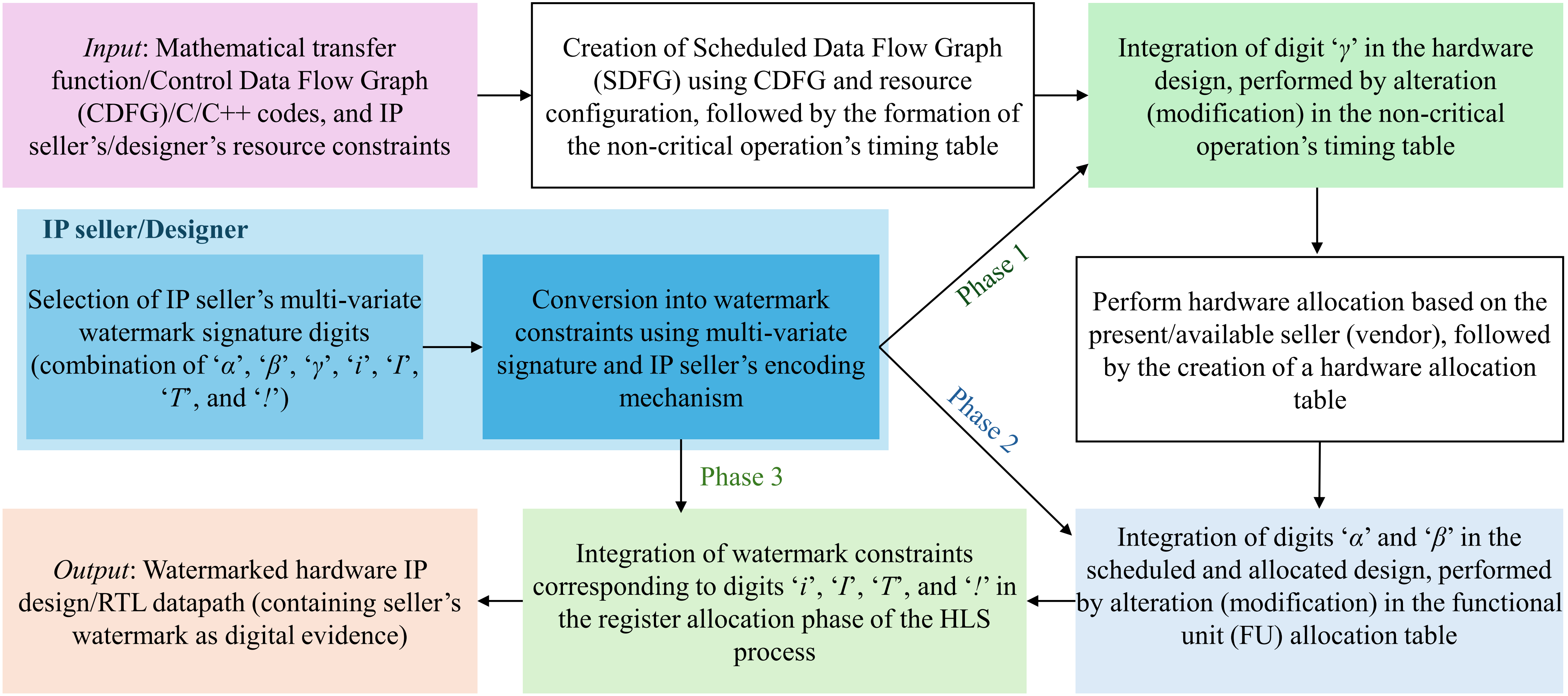
➢ **Key contributions of [6]:**

- Presents a quadruple variable-based hardware watermarking methodology.
- Exploits the register allocation phase of the HLS

➢ **Key contributions of [7]:**

- Presents a multivariate (using a seven-variable encoding scheme) signature-based hardware watermarking approach.
- Exploits scheduling, hardware allocation, and register allocation phases of the HLS process to embed hardware watermark.

• Details of hardware watermarking approaches



Input: Mathematical transfer function/Control Data Flow Graph (CDFG)/C/C++ codes, and IP seller's/designer's resource constraints

Creation of Scheduled Data Flow Graph (SDFG) using CDFG and resource configuration, followed by the formation of the non-critical operation's timing table

Integration of digit '*γ*' in the hardware design, performed by alteration (modification) in the non-critical operation's timing table

IP seller/Designer

Selection of IP seller's multi-variate watermark signature digits (combination of '*α*', '*β*', '*γ*', '*i*', '*I*', '*T*', and '*!*')

Conversion into watermark constraints using multi-variate signature and IP seller's encoding mechanism

Perform hardware allocation based on the present/available seller (vendor), followed by the creation of a hardware allocation table

Output: Watermarked hardware IP design/RTL datapath (containing seller's watermark as digital evidence)

Integration of watermark constraints corresponding to digits '*i*', '*I*', '*T*', and '*!*' in the register allocation phase of the HLS process

Integration of digits '*α*' and '*β*' in the scheduled and allocated design, performed by alteration (modification) in the functional unit (FU) allocation table

Fig. 4. High level synthesis (HLS) flow depicting the triple-phase hardware watermarking technique [7]

• Details of hardware watermarking approaches

- Numeric data-driven watermark [8]

➢ **Key contributions of** [8]**:**

• Presents a numeric data-driven watermarking approach by exploiting mathematical relations between input and output during behavioral synthesis of the HLS process.

• Discusses two different mathematical watermark generation processes:

1) a low-cost watermark and
2) a costless watermark.

• Details of hardware watermarking approaches

Code-based watermark for commercial HLS tools [12], [13]

➢ **Key contributions of [12]:**

• Presents a pragma insertion-based hardware watermarking approach using a commercial HLS tool.

• The core watermarking step involves devising a unique FU binding solution, which guarantees that the resulting RTL code is unique.

➢ **Key contributions of [13]:**

• Presents a C-code obfuscation-based watermarking approach to generate a unique hardware birthmark.

• Details of hardware watermarking approaches

Biometric signature-based watermark [9], [10], [11]

➢ **Key contributions of [9]:**

• Presents a facial biometric-based hardware watermarking methodology using the IP seller’s facial biometric characteristics.

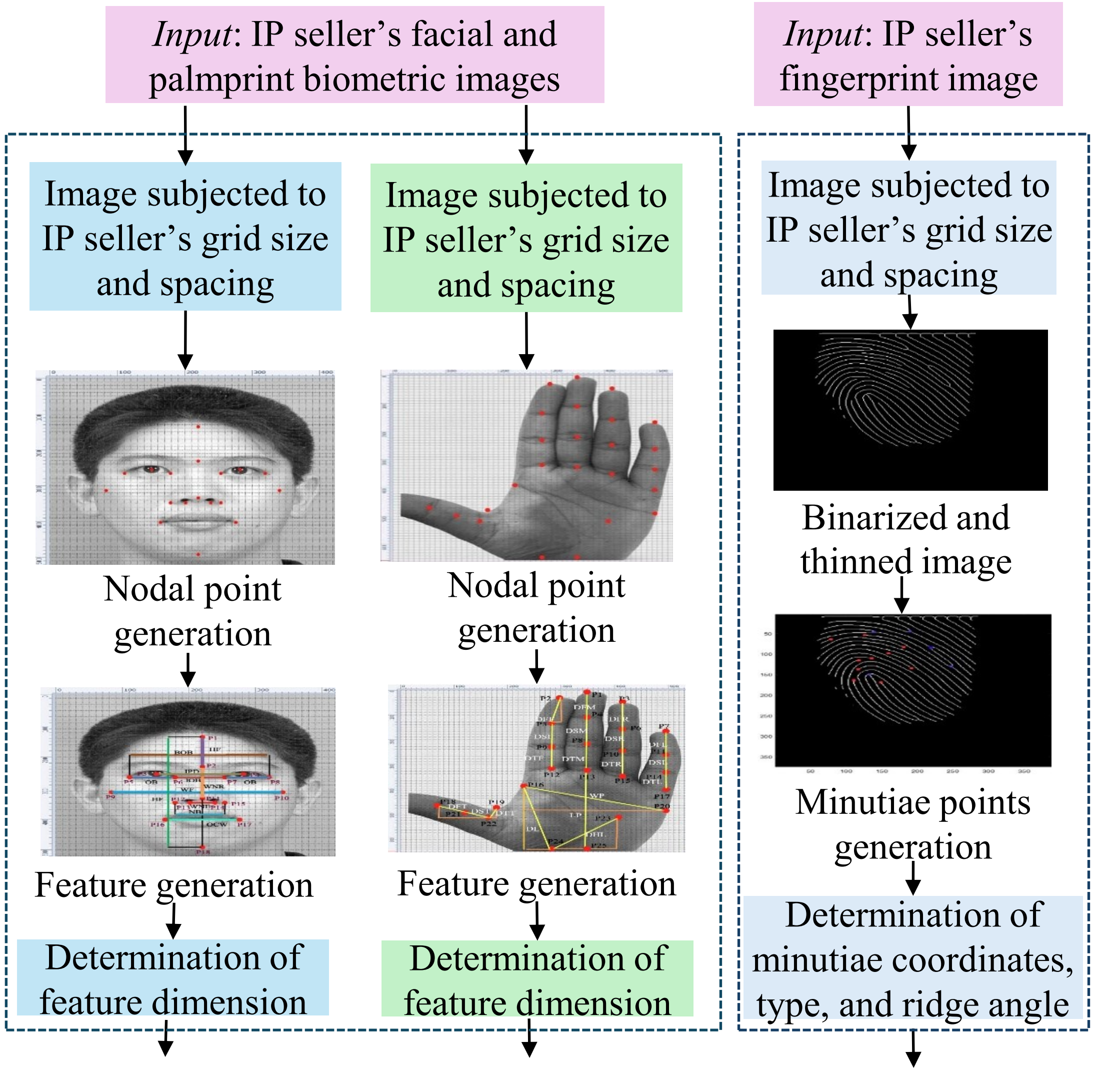
• **Key contributions of [10]:**

• Presents a palmprint biometric-based hardware watermarking methodology using the IP seller’s palmprint biometric characteristics.

➢ **Key contributions of [11]:**

• Presents a fingerprint biometric-based hardware watermarking methodology using the IP seller’s fingerprint biometric characteristics.

• Details of hardware watermarking approaches



Input: IP seller's facial and palmprint biometric images

Image subjected to IP seller's grid size and spacing

Nodal point generation

Feature generation

Determination of feature dimension

Image subjected to IP seller's grid size and spacing

Nodal point generation

Feature generation

Determination of feature dimension

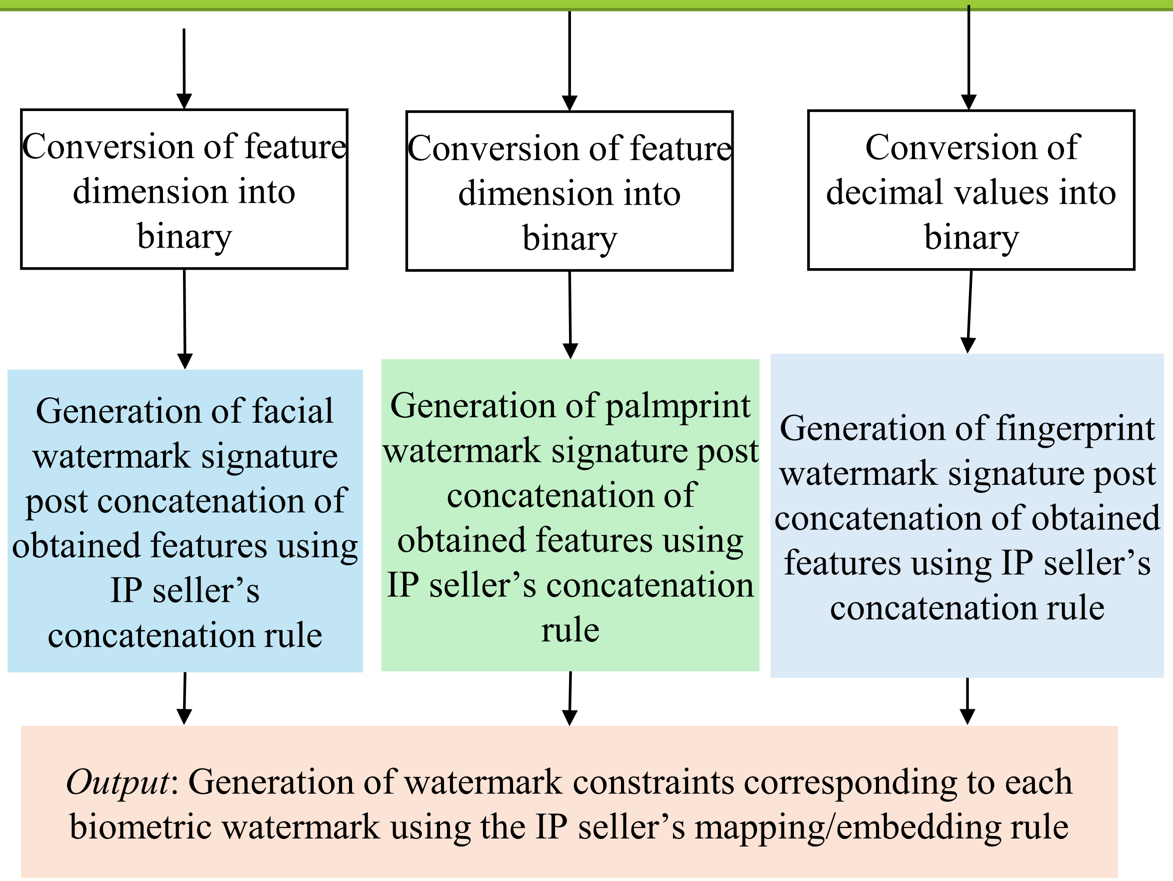
Input: IP seller's fingerprint image

Image subjected to IP seller's grid size and spacing

Binarized and thinned image

Minutiae points generation

Determination of minutiae coordinates, type, and ridge angle



Conversion of feature dimension into binary

Generation of facial watermark signature post concatenation of obtained features using IP seller's concatenation rule

Conversion of feature dimension into binary

Generation of palmprint watermark signature post concatenation of obtained features using IP seller's concatenation rule

Conversion of decimal values into binary

Generation of fingerprint watermark signature post concatenation of obtained features using IP seller's concatenation rule

Output: Generation of watermark constraints corresponding to each biometric watermark using the IP seller's mapping/embedding rule

Fig. 5. Biometric-based watermarking methodologies (such as facial biometric [9], palmprint biometric [10], and fingerprint biometric [11] based hardware security approaches)

- # Details of hardware watermarking approaches

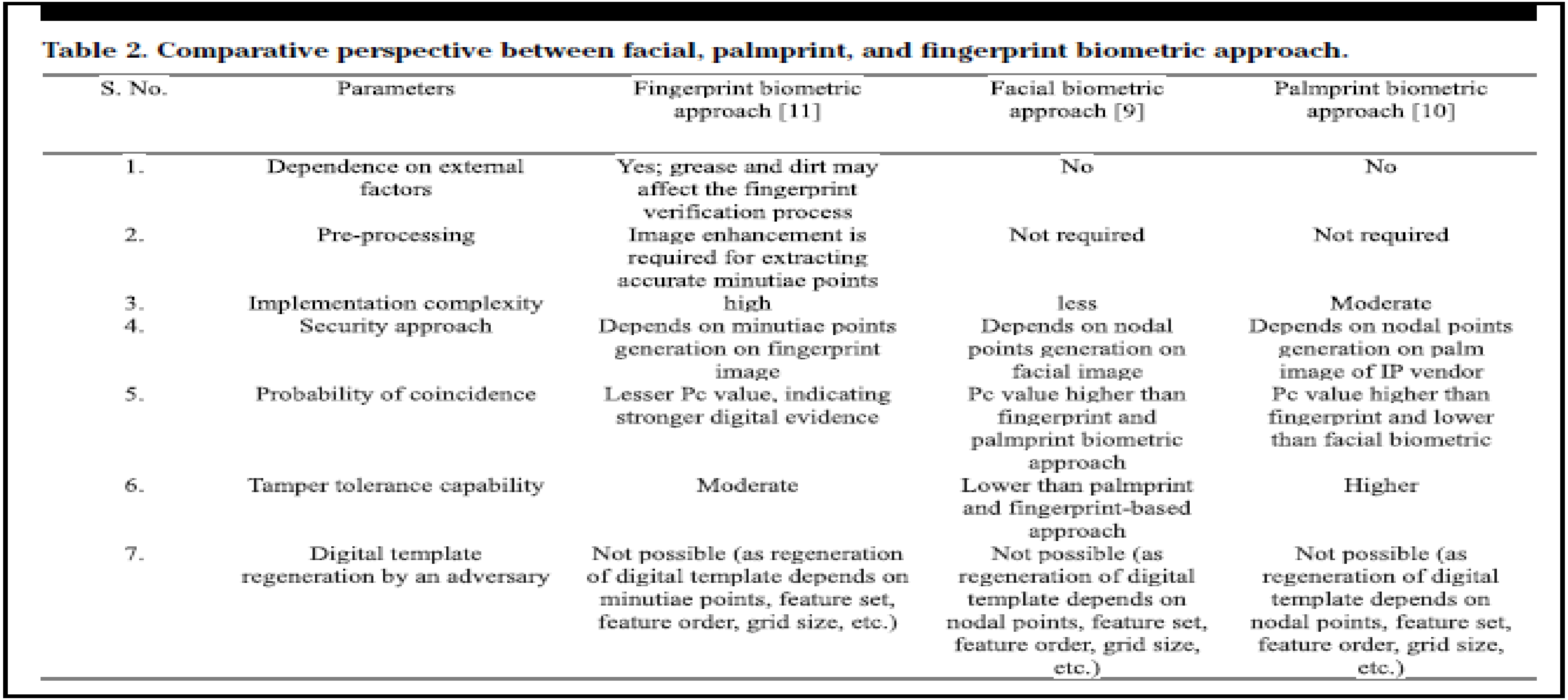


Table 2. Comparative perspective between facial, palmprint, and fingerprint biometric approach.

S. No.	Parameters	Fingerprint biometric approach [11]	Facial biometric approach [9]	Palmprint biometric approach [10]
1.	Dependence on external factors	Yes; grease and dirt may affect the fingerprint verification process	No	No
2.	Pre-processing	Image enhancement is required for extracting accurate minutiae points	Not required	Not required
3.	Implementation complexity	high	less	Moderate
4.	Security approach	Depends on minutiae points generation on fingerprint image	Depends on nodal points generation on facial image	Depends on nodal points generation on palm image of IP vendor
5.	Probability of coincidence	Lesser Pc value, indicating stronger digital evidence	Pc value higher than fingerprint and palmprint biometric approach	Pc value higher than fingerprint and lower than facial biometric
6.	Tamper tolerance capability	Moderate	Lower than palmprint and fingerprint-based approach	Higher
7.	Digital template regeneration by an adversary	Not possible (as regeneration of digital template depends on minutiae points, feature set, feature order, grid size, etc.)	Not possible (as regeneration of digital template depends on nodal points, feature set, feature order, grid size, etc.)	Not possible (as regeneration of digital template depends on nodal points, feature set, feature order, grid size, etc.)

Security analysis and discussion

- The probability of coincidence (PC shown in Fig. a) metric evaluates the likelihood of coincidently identifying the same watermarking constraints in an unsecured design, serving as an indicator of false positives.
- The tamper tolerance (TT value shown in Fig. b) metric assesses the watermarking method’s robustness against brute-force and tampering attacks.

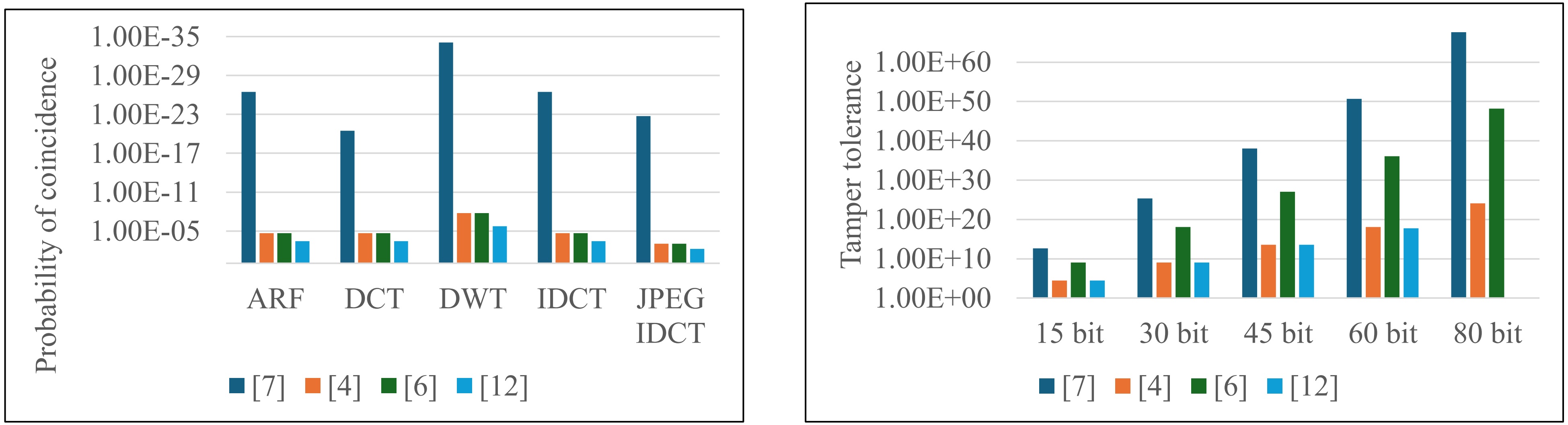
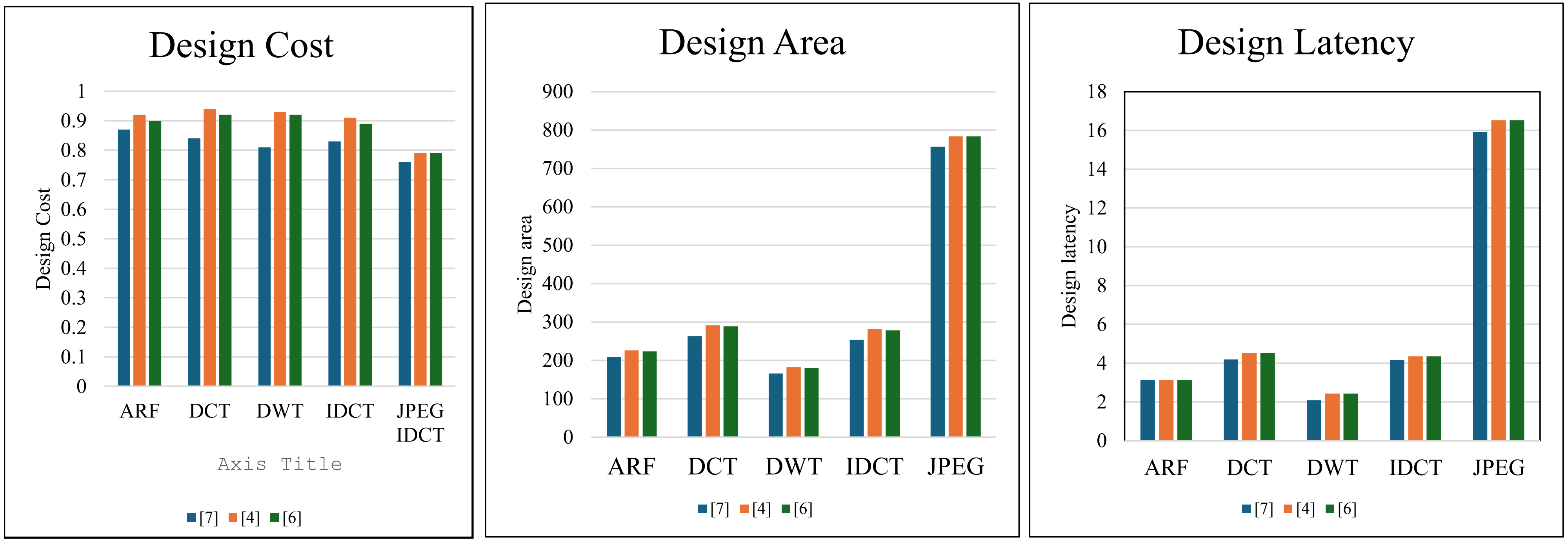


Fig. 6. (a). Comparison of the probability of coincidence among [4], [6], [7], and [12], (b). Comparison of the tamper tolerance among [4], [6], [7], and [12],

• Design Cost (Area and Latency)



(C) (d) (e)

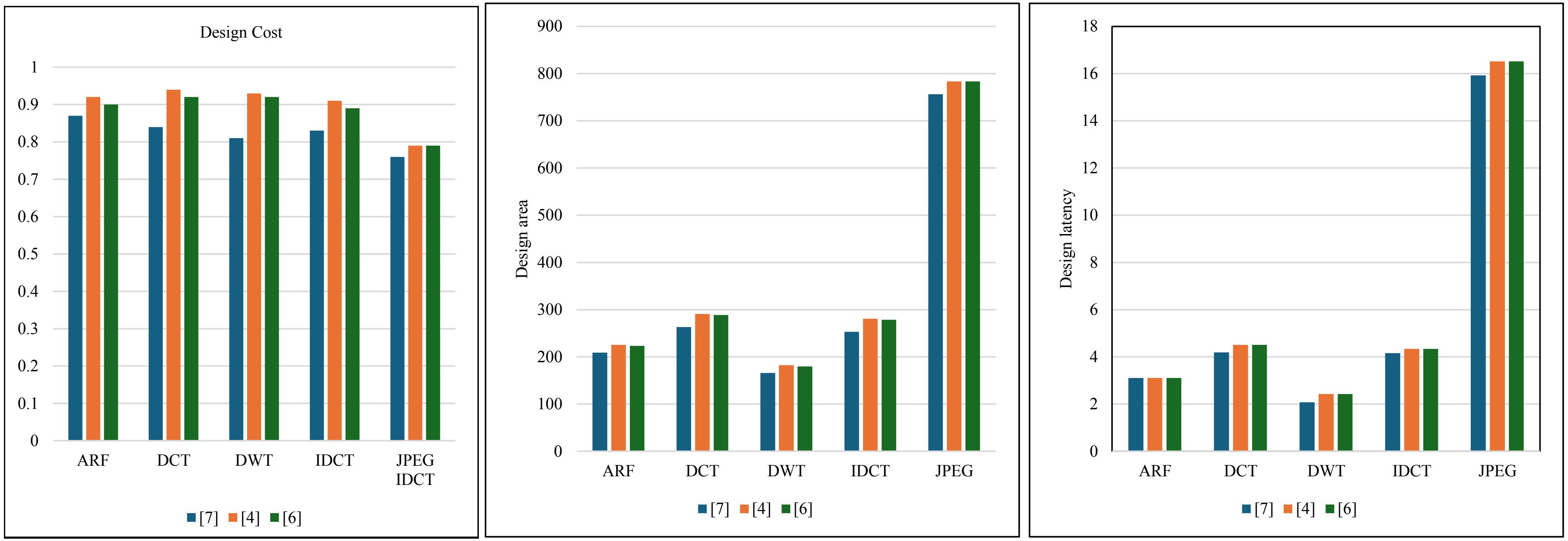
Fig. 6. (c). Comparison of design cost among [4], [6], and [7], (d). Comparison of design area among [4], [6], and [7], and (e). Comparison of design latency among [4], [6], and [7]

[4] F. Koushanfar, I. Hong and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," ACM Trans. Des. Autom. Electron. Syst., vol. 10, no. 3, pp. 523-545, 2005.

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

[7] A. Sengupta, D. Roy and S. P. Mohanty, "Triple-Phase Watermarking for Reusable IP Core Protection During Architecture Synthesis," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 37, no. 4, pp. 742-755, 2018.

• Design Cost (Area and Latency)



(C) (d) (e)

Fig. 6. (c). Comparison of design cost among [4], [6], and [7], (d). Comparison of design area among [4], [6], and [7], and (e). Comparison of design latency among [4], [6], and [7]

• References

[1] M. Rostami, F. Koushanfar and R. Karri, “A Primer on Hardware Security: Models, Methods, and Metrics,” Proceedings of the *IEEE*, vol. 102, no. 8, pp. 1283-1295, Aug. 2014.

[2] S. Rizzo, F. Bertini, and D. Montesi. “Fine-grain watermarking for intellectual property protection.”, *EURASIP J. on Info. Security*, 2019, 10 (2019).

[3] S. Ariful Islam, L. Kumar Sah, and S. Katkoori. 2020. “High-Level Synthesis of Key-Obfuscated RTL IP with Design Lockout and Camouflaging”. *ACM Trans. Des. Autom. Electron. Syst*. 26.

[4] F. Koushanfar, I. Hong and M. Potkonjak, “Behavioral synthesis techniques for intellectual property protection,” *ACM Trans. Des. Autom. Electron. Syst*., vol. 10, no. 3, pp. 523-545, 2005.

[5] A. Sengupta, E. R. Kumar, and N. P. Chandra, “Embedding digital signature using encrypted hashing for protection of DSP Cores in CE,” *IEEE Trans. Consum. Electron*., vol. 65, no. 3, pp. 398–407,Aug. 2019.

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

[7] A. Sengupta, D. Roy and S. P. Mohanty, “Triple-Phase Watermarking for Reusable IP Core Protection During Architecture Synthesis,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 4, pp. 742-755, 2018.

[8] L. Gal, B. Bossuet, “Automatic low-cost IP watermarking technique based on output mark insertions”, *Des Autom Embed Syst* 16, 71–92 (2012).

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

• References

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

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

[12] J. Chen and B. C. Schafer, “Watermarking of Behavioral IPs: A Practical Approach,” 2021 *IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE*), Grenoble, France, 2021.

[13] H. Badier, C. Pilato, J. -C. L. Lann, P. Coussy and G. Gogniat, “Opportunistic IP Birthmarking using Side Effects of Code Transformations on High-Level Synthesis,” *2021 IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE),* Grenoble, France, 2021.

[14] M. Lewandowski and S. Katkoori, “A Darwinian Genetic Algorithm for State Encoding Based Finite State Machine Watermarking,” 20th *International Symposium on Quality Electronic Design, USA*, 2019, pp. 210-215.

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

[16] University of California Santa Barbara Express Group, accessed on May. 2024. [Online]. Available: http://express.ece.ucsb.edu/benchmark/.

[17] M. Potkonjak “Methods and systems for the identification of circuits and circuit designs,” USPTO, US7017043B1, 2006.

[18]Xiaolong Guo, R. G. Dutta, Yier Jin, F. Farahmandi and P. Mishra, “Pre-silicon security verification and validation: A formal perspective,” 52nd ACM/EDAC/IEEE Design Automation Conference, USA, 2015, pp. 1-6.

THANK YOU