A Survey of High Level Synthesis Based Hardware Security Approaches for Reusable IP Cores

Anirban Sengupta, CSE, Indian Institute of Technology Indore

Aditya Anshul, Anirban Sengupta "A Survey of High Level Synthesis based Hardware Security Approaches for Reusable IP Cores", IEEE Circuits and Systems Magazine (CASM), Volume: 23, Issue: 4, 2023, pp. 44 - 62

Introduction

- This paper presents a novel survey of high-level synthesis (HLS) based hardware security approaches for reusable intellectual property (IP) cores used in the consumer electronics devices (such as mobile-phones, tablets, computers, digital cameras, digital music systems, smartwatches, and smart televisions etc.).

- The computing devices contain different types of application-specific integrated circuits (ASICs) that are intended to perform different crucial and data-intensive functions such as digital data filtering, compression and decompression of digital data, and different types of mathematical computations on digital data [20]-[24].

- These ASICs can also be designed as dedicated digital signal processing (DSP) intellectual property (IP) cores, and each DSP core is intended to perform a particular function on digital data.

- The paper presents a detailed design flow of hardware integrated circuits (ICs) along with vulnerability points where potential attacks/threats are possible.

- Trustworthy and untrustworthy regimes in the design flow have also been highlighted in the discussion.

Possible hardware threats and attacks in the design flow of hardware IC

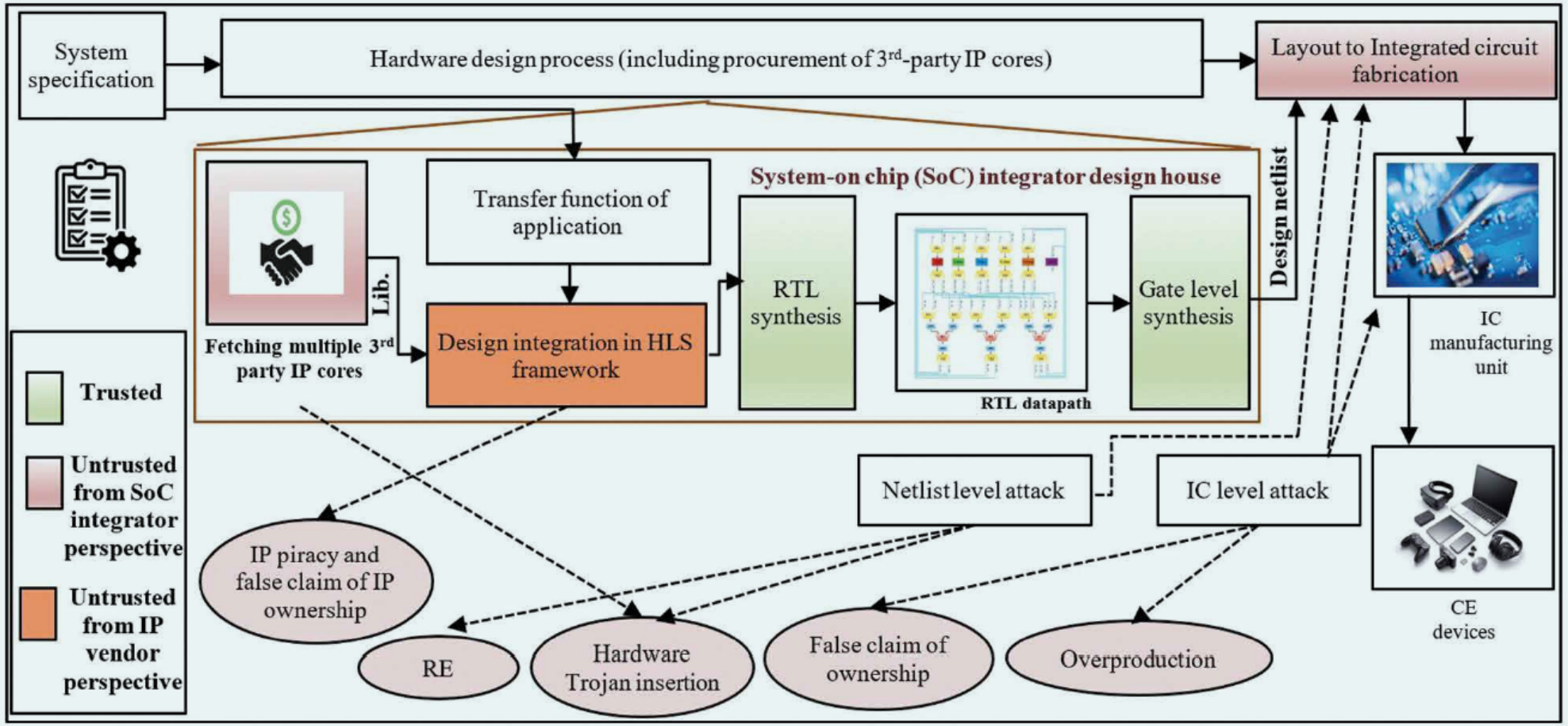

Figure 1. Possible hardware threats and attacks in the design flow of hardware IC.

Possible hardware threats and attacks in the design flow of hardware IC (Contd.)

- As shown in Figure 1,
  - The *input consists of system specifications*, which are the behavioral descriptions of the intended hardware design.
  - These specifications are then progressed through the *hardware design process*.
  - This process entails *acquiring various IP cores or designs from multiple third-party IP vendors*, followed by the *integration of these imported cores into a single chip* carried out by System-on-Chip (SoC) integrator.
    - This is because some places can afford lower technical costs while some cheap labor, besides time to market factor.
  - After *integration*, a corresponding register transfer level *(RTL) file is generated*, which subsequently undergoes *synthesis to transform it into a gate-level design file*, also known as a *netlist file*.
  - The *netlist* file is subsequently *transmitted to fabrication and manufacturing* facilities.
    - If physical design process (routing, floor-planning, layout generation etc. in the form of graphic design system (GDS) file) is not possible in the fabrication house, then SoC integrator will do that process.

- Here,
  - The *green-colored* components signify the trusted sector of the hardware design supply chain process,
  - The *orange-colored* component belongs to untrustworthy sector from an IP vendor's perspective, and
  - The *red-colored* component belongs to the untrustworthy sector from an SoC integrators perspective.

Possible hardware threats and attacks

- As shown in Figure 1, hardware attacks are categorized into four main types:
  - *(a) IP piracy and false claim of IP ownership*: can be potentially performed illegally by an adversary in the SoC integrator house
    - IP piracy (counterfeiting and cloning)
    - False claim of IP ownership,
  - *(b) Hardware Trojan insertion*: insertion of malicious logic through 3rd party IP (3PIP) cores,
  - *(c) The netlist level attacks*: can be potentially performed by an adversary in the foundry (fabrication house)
    - Reverse engineering (RE)
    - Hardware Trojan insertion, and
  - *(d) IC level attacks*: that can be potentially performed by an adversary in the foundry or open market
    - False claim of ownership
    - IC overbuilding/overproduction.

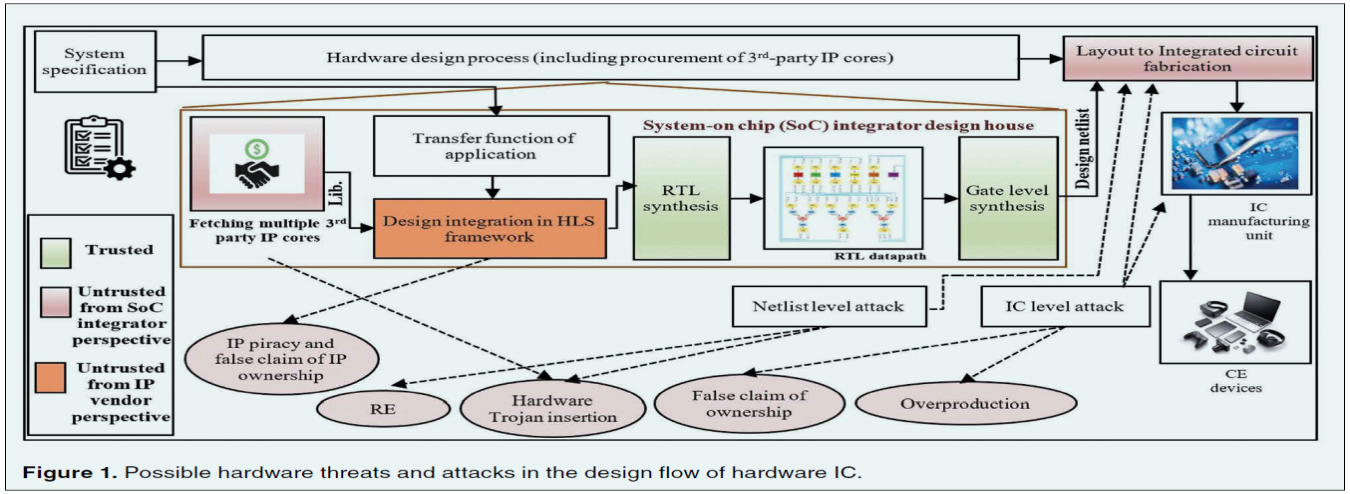

Figure 1. Possible hardware threats and attacks in the design flow of hardware IC.

Tree structure of the different hardware security techniques

Figure 2 illustrates the tree structure of the different hardware security techniques based on the different types of possible hardware threats.

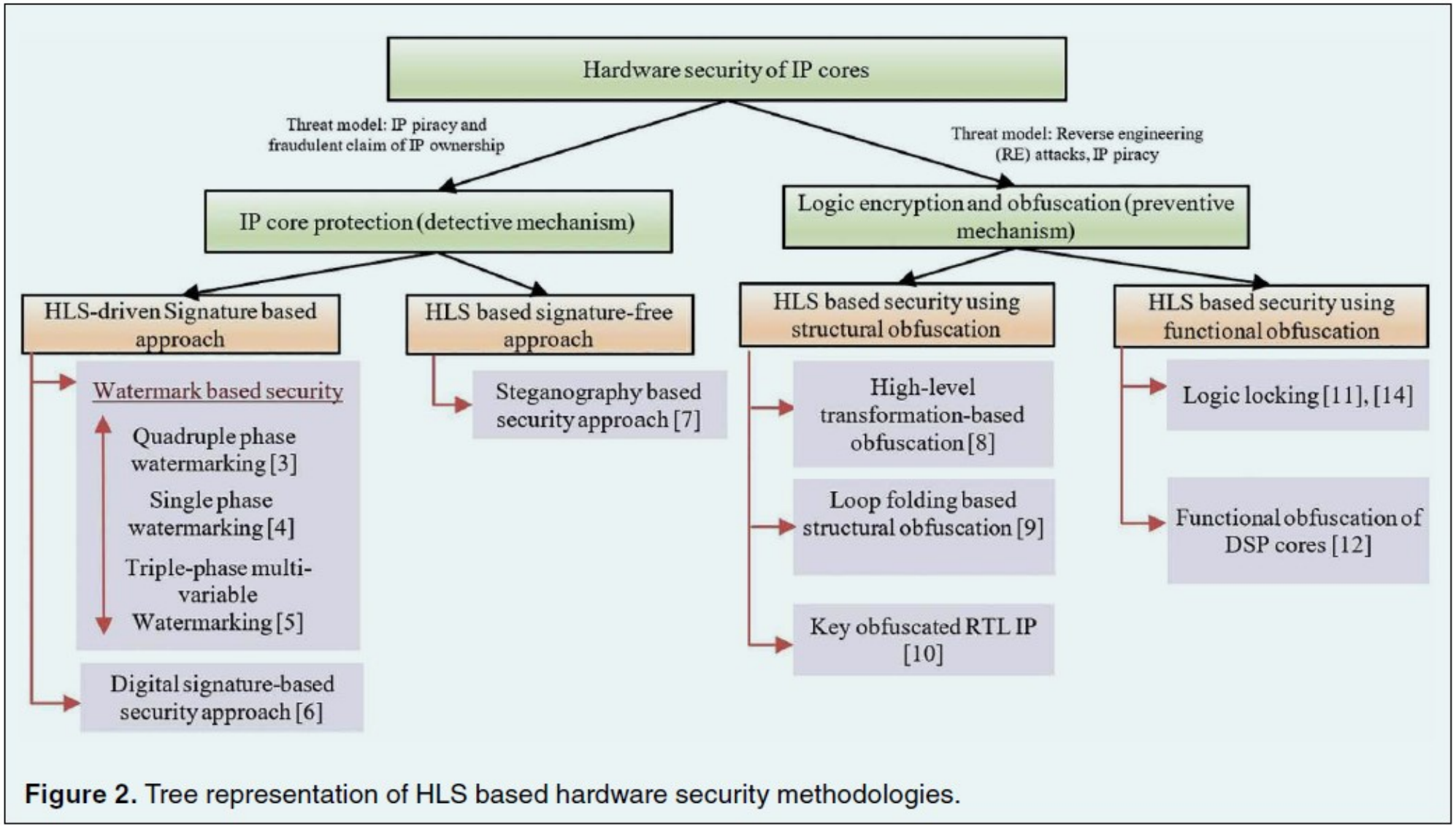

Figure 2. Tree representation of HLS based hardware security methodologies.

HLS-driven watermarking-based hardware security

- Figure 3 illustrates the details of a triple-phase watermarking technique [5] in the HLS framework.

- The IP vendor selected seven different signature digits and their corresponding encoding rules are as follows:

  (i) '**α**'= operations with odd numbers are allocated to hardware of vendor type 1 and operations with even numbers are allocated to hardware of vendor type 2 in odd control step (CS),

  (ii) '**β**'= operations with odd numbers are allocated to hardware of vendor type 2 and operations with even numbers are allocated to hardware of vendor type 1 in even control step,

  (iii) '**γ**'= a noncritical path operation with highest mobility is moved to immediate next control step,

  (iv) '**i**'= embed an artificial edge between <prime, prime> node pairs (storage variables) in colored interval graph (CIG) of DSP application,

  (v) '**I**'= embed an artificial edge between <even, even> node pairs (storage variables) in CIG of DSP application,

  (vi) '**T**'= embed an artificial edge between <odd, even> node pairs (storage variables) in CIG of DSP application, and

  (vii) '**!**' = embed an artificial edge between <0, any integer> node pairs (storage variables) in CIG of DSP application.

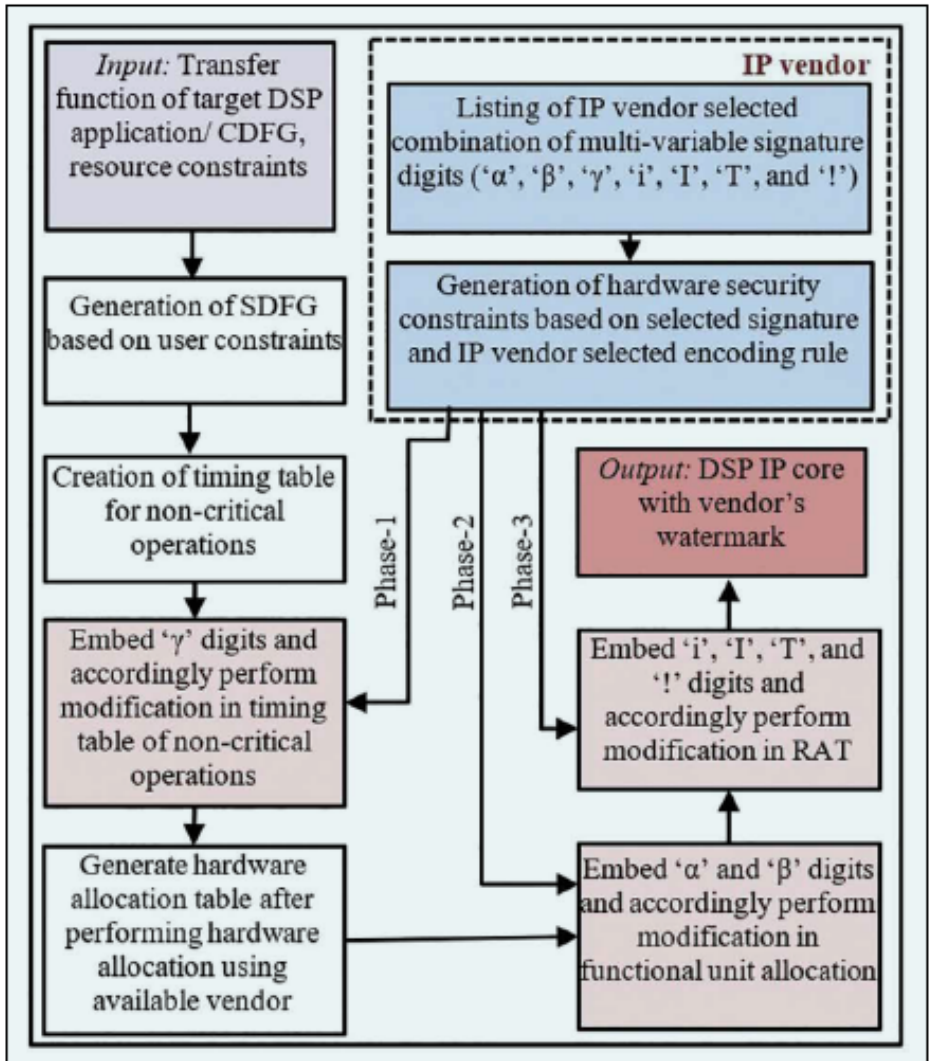

Figure 3. Secure HLS flow of triple phase watermarking mechanism for hardware IP cores [5].

[5] 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, Apr. 2018.

HLS-driven watermarking-based hardware security

- Here, in the ***first phase of watermarking***, non-critical operations (starting from CS 1) are moved to the immediate next CS for each occurrence of **'γ'** (shifting must not violate the data dependency and hardware constraints), and a modified timing table for non-critical operations is generated. A hardware allocation table is generated corresponding to different used functional units (hardware).

- Further, in the ***second phase of watermarking***, FUs are re-allocated according to the IP vendor selected encoding rules **'α'** and **'β'**, and a modified hardware allocation table is generated. After this, allocation of storage variables in the SDFG (double phased watermarked) is performed, and a CIG is created to find the minimum number of required registers for storage variables. Next, a register allocation table (RAT) is created from SDFG (assigned with storage variables).

- Then, in the ***third phase of watermarking,*** the additional artificial edges (security constraints) are determined based on the IP vendor's selected **'I'**, **'I'**, **'T'**, and **'!'** digits. Further, these determined security constraints are embedded into the CIG of the design, followed by local alteration in register allocation if two adjacent register's colors are the same. To resolve this conflict, either colors of the register are swapped, or a new colored register is allocated.

- Finally, RAT of triple-phase watermarked hardware IP core is generated using HLS.

- The involvement of *7-digit multi-variable signature* and different *IP vendor-selected encoding mechanisms* for different phases (scheduling, hardware allocation, and register allocation) of watermarking makes the proposed approach highly robust.

HLS-driven hardware steganography-based security approach

- Figure 5 illustrates the details of IP core steganography-based security approach [7].
- Introduces a hardware steganography method based on key-driven hash-chaining.
- The presented IP core steganography includes switch-controlled hash blocks (using stego-keys), standard hash blocks, and diverse encoding algorithms within the hash-chain process.

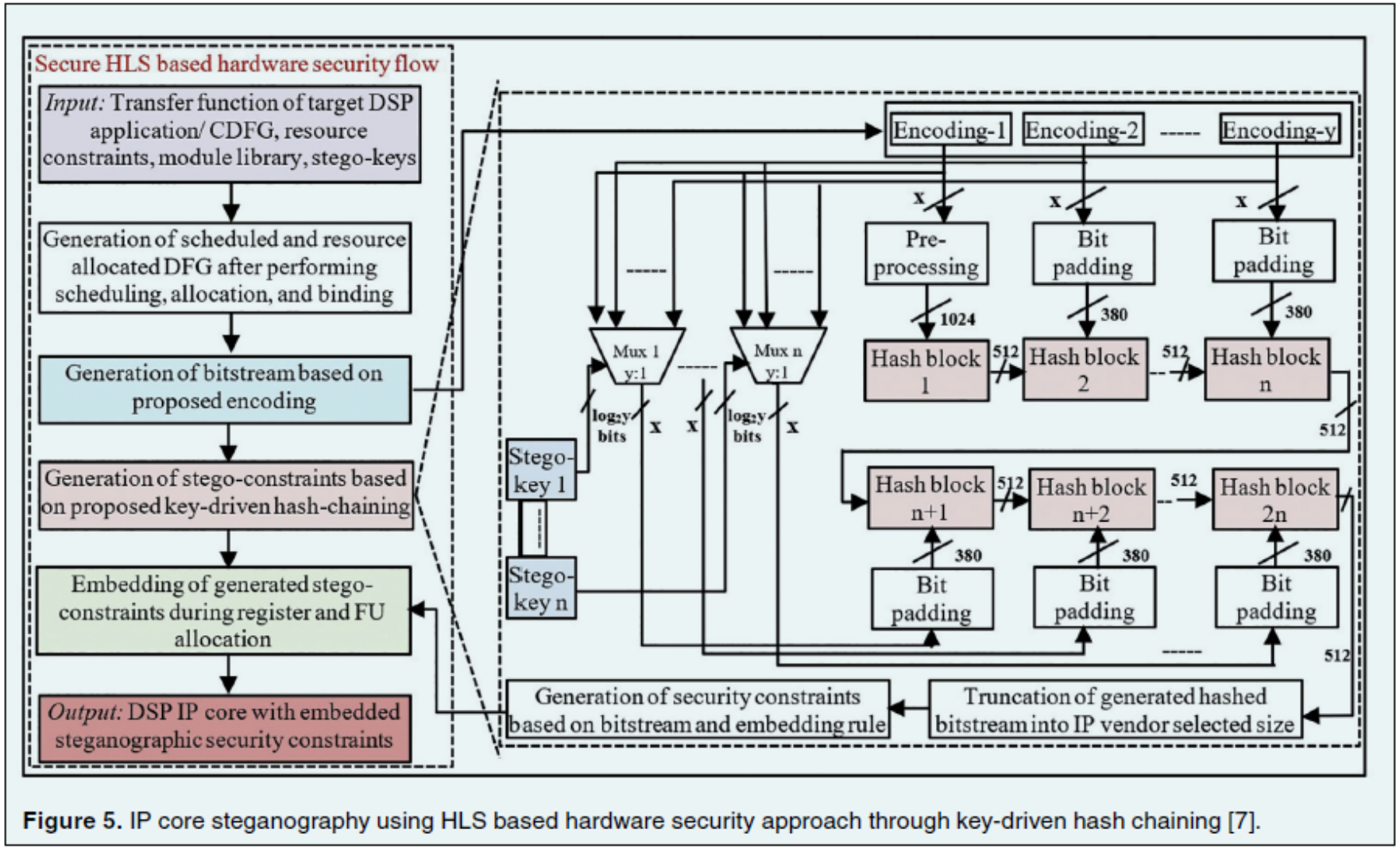

Figure 5. IP core steganography using HLS based hardware security approach through key-driven hash chaining [7].

[7] M. Rathor and A. Sengupta, "IP core steganography using switch based key-driven hash-chaining and encoding for securing DSP kernels used in CE systems," IEEE Trans. Consum. Electron., vol. 66, no. 3, pp. 251–260, Aug. 2020.

HLS-driven hardware steganography-based security approach

- In the presented approach steganography-based security approach [7]:
  - First, a bitstream is generated based on the IP vendor selected encoding rules extracted from the scheduled design itself (E1-E9).
  - Subsequently, a hash digest of the bitstream is generated based on IP vendor selected stego-keys.
  - Finally, the generated hashed bitstream (in binary form) is converted into security constraints based on the IP vendor's selected embedding rule to yield a secured DSP IP core containing the IP vendor's secret stego mark

- The nine different encoding rules corresponding to encoding block to generate initial bitstream are as follow:

  'E1' = the output bit is '0' if the CS number and operation number are even otherwise '1',

  'E2' = the output bit is '0' if the CS number and operation number are of same parity otherwise '1',

  'E3' = the output bit is '0' if the CS number and operation number are odd otherwise '1',

  'E4' = the output bit is '0' if the CS number and operation number are of different parity otherwise '1',

  'E5' = the output bit is '0' if the CS number and operation number are prime otherwise '1',

  'E6' = the output bit is '0' if the CS number and operation number are prime otherwise '1',

  'E7' = the output bit is '0' if the GCD of CS number and operation number is one otherwise '1',

  'E8' = the output bit is '0' if (operation number) mod (corresponding CS number) is zero otherwise '1', and

  'E9' = the output bit is '0' if CS number is equal to second odd sequence of operation number otherwise '1'.

[7] M. Rathor and A. Sengupta, "IP core steganography using switch based key-driven hash-chaining and encoding for securing DSP kernels used in CE systems," IEEE Trans. Consum. Electron., vol. 66, no. 3, pp. 251–260, Aug. 2020.

HLS-driven hardware steganography-based security approach

- If the total number of operations in the SDFG is ‘r’, then a r-bit bitstream output is generated based on any of the IP vendor's selected encoding rules.
- Moreover, the generated “r-bit” is appended with ‘1’ and then with ‘0’ to generate 896-bits (bitstream) as output, which is further again appended with the 128-bit representation of the length ‘r-bit’ of the encoded bitstream to generate 1024 bit as output.
- Now, the generated 1024-bit bitstream is fed as input to the hash chaining block.

- Further, the output 512-bit of the first hash block concatenated with “1000”, 380-bit padding block, and 128-bit representation of the previous 512-bit hash together is fed as the input to the second hash block.
- This process continues till n-hash blocks of the security algorithm.

Example:

896-bits (*i.e.*, r-bit ╫ 100…0) ╫ **128**-bit representation of the length r-bit => **1024** bits => *Hash Block 1* => **512** bits

512 bits o/p ╫ **4**-bits (*i.e.*, 1000) ╫ **380**-bits (*i.e.*, 000…0 ╫ r-bits) + **128**-bit representation of previous 512-bit hash => 1024 bits => *Hash Block 2* => **512** bits

……

The final output of the “2n” hash block is a **512**-bit bitstream digest which is further truncated based on the IP vendor’s selected truncation length.

HLS-driven hardware steganography-based security approach

- The truncated bitstream is converted into hardware security constraints (stego-constraints) based on IP vendor selected embedding rules:
  - If the bit is ‘0’, then an artificial edge between <even, even> storage variable (node) pairs of the CIG is added
  - If the bit is ‘1’, then all odd operations in SDFG are assigned with vendor 1 type FUs and all even operations with vendor 2 type FUs.

- Finally, all the generated covert security constraints are embedded into the register allocation design of the DSP application during HLS.

- In the end, a secured DSP IP core at RTL is produced as the output, containing stego-security constraints as digital evidence, which can be used to provide robust detection against pirated IP cores.

HLS-driven structural obfuscation-based hardware security

- Obfuscation is the art of making any design of interest unobvious to an adversary without affecting its original functionality.
  - *Structural obfuscation* hides the functionality and implementation of a design by altering the structure intentionally.
  - *Functional obfuscation* hides the functionality and implementation of a design by inserting additional key gates (logic lock) into it.

- Figure 6 presents high-level transformation-based approach to implement RTL structural transformations along with the integration of PSO-DSE to obtain low-cost obfuscated DSP RTL design [8].

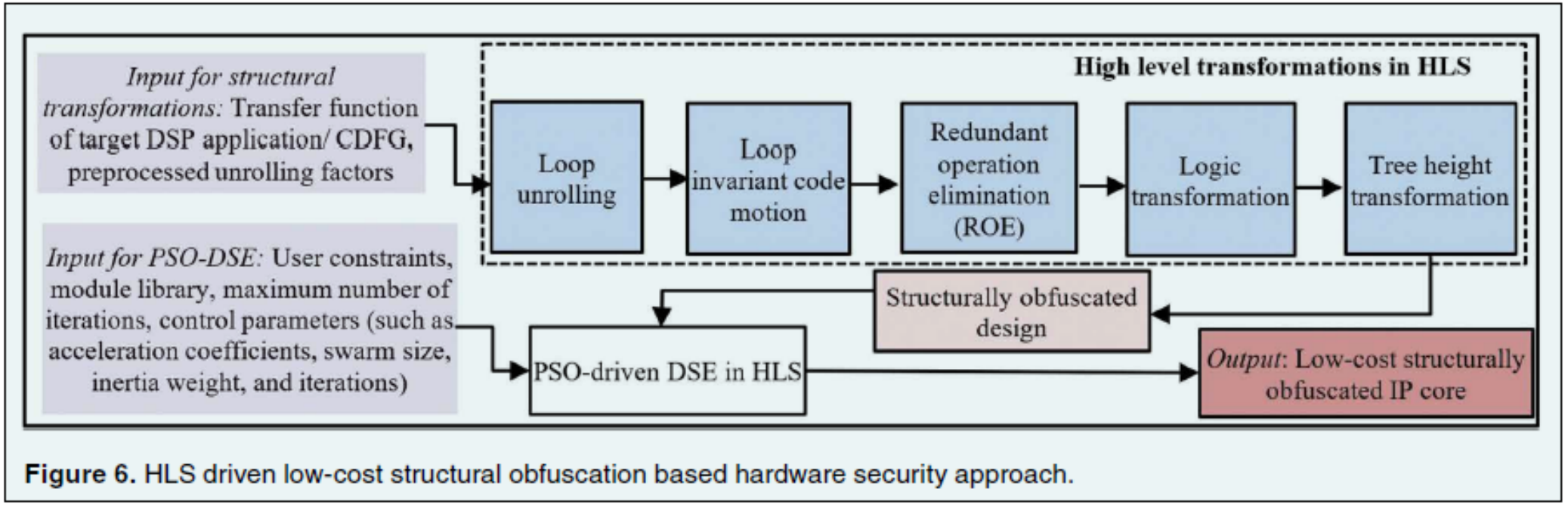

Figure 6. HLS driven low-cost structural obfuscation based hardware security approach.

[8] A. Sengupta et al., "DSP design protection in CE through algorithmic transformation based structural obfuscation," IEEE Trans. Consum. Electron., vol. 63, no. 4, pp. 467–476, Nov. 2017.

HLS-driven structural obfuscation-based hardware security

- Five algorithmic transformation techniques are [8]:
  - Loop unrolling (LU),
  - Loop invariant code motion (LICM),
  - Redundant operation elimination (ROE),
  - Logic transformation (LT), and
  - Tree height transformation (THT).

- After implementing all the algorithmic transformations, the generated structurally obfuscated CDFG is fed to the PSO-based DSE block to generate a low-cost structurally obfuscated RTL datapath corresponding to the DSP IP core using inputs for the PSO-DSE block.

- PSO performs pruning in the design search space to obtain a low-cost optimized design solution corresponding to obfuscated DSP application [8].

- Finally, a low-cost structurally obfuscated RTL IP core is obtained with the capability to hinder RE attacks.

[8] A. Sengupta et al., "DSP design protection in CE through algorithmic transformation based structural obfuscation," IEEE Trans. Consum. Electron., vol. 63, no. 4, pp. 467–476, Nov. 2017.

HLS-driven functional obfuscation-based hardware security

- The art of performing logic encryption (with the help of key gate logic) to lock the original functionality of the IP design is known as *functional obfuscation*.
  - An adversary will need the correct set of keys to unlock the design, prohibiting the illegal use of the IP core.

- Figure 8 describes a robust logic locking technique to perform functional obfuscation [12].
- The functional obfuscation methodology is implemented in two blocks: (a) *PSO-based DSE block* and (b) *logic encryption block*, respectively.

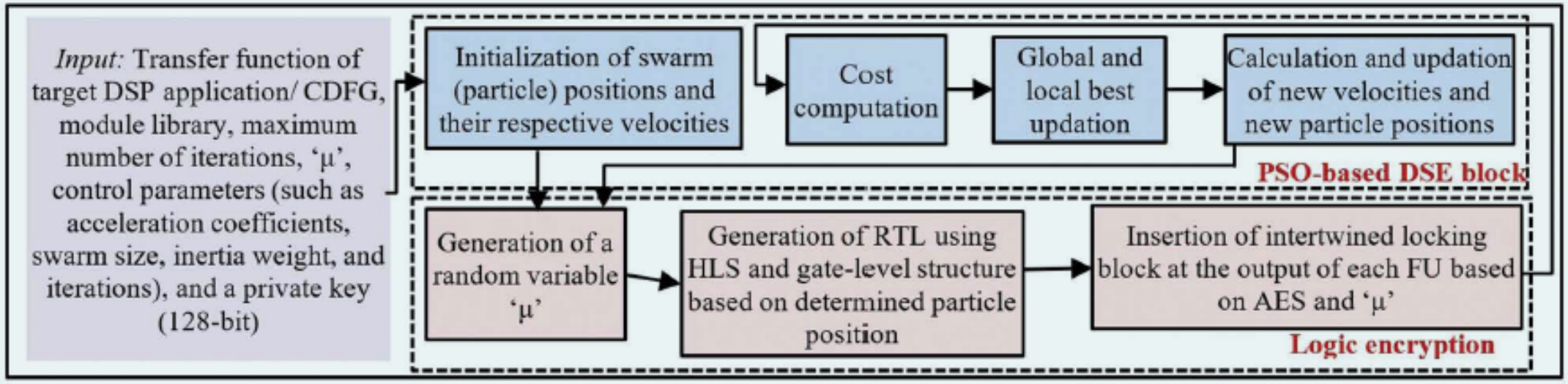

Figure 8. PSO-HLS based logic locking technique for performing functional obfuscation.

[12] A. Sengupta, D. Kachave, and D. Roy, "Low cost functional obfuscation of reusable IP cores used in CE hardware through robust locking," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 38, no. 4, pp. 604–616, Apr. 2019.

HLS-driven functional obfuscation-based hardware security

- IP locking blocks (ILBs) are composed of a combination of AND, NAND, XOR, XNOR, NOT, and OR gates. The key benefits of ILBs are as follows:

  (a) *Multi-pair wise security*: To sensitize one-bit input to the output decoding of eight key bits is required. The inserted ILB blocks depend on multiple key-bits,

  (b) *Prohibiting key gate isolation*: inserted ILBs are free from isolated gates, as key i/ps are dependent on each other (*i.e.*, there is a path between the keys),

  (c) *Resilient to run of key gates*: In the design of the ILB, no sequence of key gates can be substituted with a single key gate, and

  (d) *Resilient to the muting of key gates*: ILBs are designed in a way that all keys impart some value towards output generation. Muting any particular key gate by an adversary is never possible.

- The used random variable ‘μ’ lies between 1 and IP designer selected repeated patterns of ILBs used for insertion into the gate level design datapath.

  $1 \leq \mu \leq T_{ILB}$; where, T_{ILB} = IP designer selected repeated patterns of ILBs = 4 (in [12])

- The integration of PSO-based DSE helps to determine a final low-cost optimized functionally obfuscated design (with minimal design cost overhead) among potential designs possible in the design search space.

[12] A. Sengupta, D. Kachave, and D. Roy, “Low cost functional obfuscation of reusable IP cores used in CE hardware through robust locking,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 38, no. 4, pp. 604–616, Apr. 2019.

HLS-driven functional obfuscation-based hardware security

- To protect the functionally obfuscated design (using ILBs) against SAT attacks, a customized lightweight advanced encryption standard (AES) block (with an IP designer selected private key) has been used, whose output will be concatenated with the input of inserted ILBs.

  SAT attack model:

  - The attacker is an untrusted foundry whose objective is to unlock the locked netlist.
  - The attacker has access to the GDSII layout file. RE to locked netlist from GDS file is the target for him/her. This is because fabrication is performed at the foundry containing layout details provided by the designer.

- Further, the combined synthesis of custom lightweight AES block (not publicly available) with the intertwined logic blocks helps to generate an indistinguishable circuit design, making it challenging for an adversary to identify and remove the logic encryption units.

[12] A. Sengupta, D. Kachave, and D. Roy, "Low cost functional obfuscation of reusable IP cores used in CE hardware through robust locking," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 38, no. 4, pp. 604–616, Apr. 2019.

References

[1] C. Pilato et al., “Securing hardware accelerators: A new challenge for high-level synthesis,” IEEE Embedded Syst. Lett., vol. 10, no. 3, pp. 77–80, Sep. 2018.

[2] G. T. Becker, M. Fyrbiak, and C. Kison, “Hardware obfuscation: Techniques and open challenges,” in Foundations of Hardware IP Protection, L. Bossuet and L. Torres, Eds. Cham, Switzerland: Springer, 2017.

[3] M. Rathor et al., “Quadruple phase watermarking during high level synthesis for securing reusable hardware intellectual property cores,” Comput. Elect. Eng., vol. 105, Jan. 2023, Art. no. 108476.

[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, 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, Apr. 2018.

[6] 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.

[7] M. Rathor and A. Sengupta, “IP core steganography using switch based key-driven hash-chaining and encoding for securing DSP kernels used in CE systems,” IEEE Trans. Consum. Electron., vol. 66, no. 3, pp. 251–260, Aug. 2020.

[8] A. Sengupta et al., “DSP design protection in CE through algorithmic transformation based structural obfuscation,” IEEE Trans. Consum. Electron., vol. 63, no. 4, pp. 467–476, Nov. 2017.

[9] Y. Lao and K. K. Parhi, “Obfuscating DSP circuits via high-level transformations,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 5, pp. 819–830, May 2015.

[10] S. A. Islam, L. K. Sah, and S. Katkoori, “High-level synthesis of keyobfuscated RTL IP with design lockout and camouflaging,” ACM Trans. Des. Autom. Electron. Syst., vol. 26, no. 1, pp. 1–35, 2020.

[11] M. Yasin et al., “On improving the security of logic locking,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 35, no. 9, pp. 1411– 1424, Sep. 2016.

[12] A. Sengupta, D. Kachave, and D. Roy, “Low cost functional obfuscation of reusable IP cores used in CE hardware through robust locking,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 38, no. 4, pp. 604–616, Apr. 2019.

[13] J. J. V. Rajendran, “An overview of hardware intellectual property protection,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2017, pp. 1–4.

[14] C. Pilato et al., “ASSURE: RTL locking against an untrusted foundry,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 29, no. 7, pp. 1306–1318, Jul. 2021.

[15] W. Hu et al., “An overview of hardware security and trust: Threats, countermeasures, and design tools,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 40, no. 6, pp. 1010–1038, Jun. 2021.

[16] M. Potkonjak, “Methods and systems for the identification of circuits and circuit designs,” U.S. Patent 7 017 043 B1, Mar. 21, 2006.

[17] A. Sengupta, S. Bhadauria, and S. P. Mohanty, “Embedding low cost optimal watermark during high level synthesis for reusable IP core protection,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Montreal, QC, Canada, May 2016, pp. 974–977.

[18] NIST Computer Security Resource Center. Glossary. USA. Accessed: Feb. 2022. [Online]. Available: https://csrc.nist.gov/glossary/term/ entropy#:~:text=A%20measure%20of%20the%20amount,is%20usually% 20stated%20in%20bits

[19] Express Benchmark Suite. Accessed: Jan. 2022. [Online]. Available: http://www.ece.ucsb.edu/EXPRESS/benchmark/.

[20] Anirban Sengupta, Saumya Bhadauria, "Exploring Low Cost Optimal Watermark for Reusable IP Cores during High Level Synthesis", IEEE Access, Volume:4, Issue: 99, pp. 2198 - 2215, May 2016

[21] Anirban Sengupta, Saumya Bhadauria, Saraju P Mohanty "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling with Optimal Loop Unrolling Factor during High Level Synthesis", IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems (TCAD), Volume: 36, Issue: 4, April 2017, pp. 655 – 668

[22] Anirban Sengupta, Dipanjan Roy, Saraju Mohanty, Peter Corcoran "DSP Design Protection in CE through Algorithmic Transformation Based Structural Obfuscation", IEEE Transactions on Consumer Electronics, Volume 63, Issue 4, November 2017, pp: 467 – 476

[23] Anirban Sengupta, Mahendra Rathor "IP Core Steganography for Protecting DSP Kernels used in CE Systems", IEEE Transactions on Consumer Electronics (TCE) , Volume: 65 , Issue: 4 , Nov. 2019, pp. 506 – 515

[24] Anirban Sengupta, Mahendra Rathor "Securing Hardware Accelerators for CE Systems using Biometric Fingerprinting", IEEE Transactions on Very Large Scale Integration Systems , Vol 28, Issue: 9, 2020, pp. 1979-1992

Thank You !!!