Low-Cost Hardware Security of Laplace Edge Detection (LED) and Embossment Filter Using HLS Based Encryption and PSO

Aditya Anshul, Anirban Sengupta "Low-Cost Hardware Security of Laplace Edge Detection and Embossment Filter Using HLS Based Encryption and PSO", Proceedings of 9th IEEE International Symposium on Smart Electronic Systems (iSES) , India, Dec 2023, pp. 135-140

Image processing filters

- Image processing filters are mainly used to suppress either the high frequencies in the image, *i.e.,* smoothing the image, or low frequencies, enhancing or detecting edges in the image, etc.
- The main objective of image processing is to extract some useful information from an image.
- From detection and recognition of license plates of vehicles on tolls (character recognition), advanced medical imagery (image analysis), biometric fingerprinting, robotics vision, and military operations to car driving automation, image processing plays a crucial role everywhere.
- Due to globalization of design supply chain, the design process of these image processing filters as dedicated intellectual property (IP) core involves various hardware threats [1], [2].

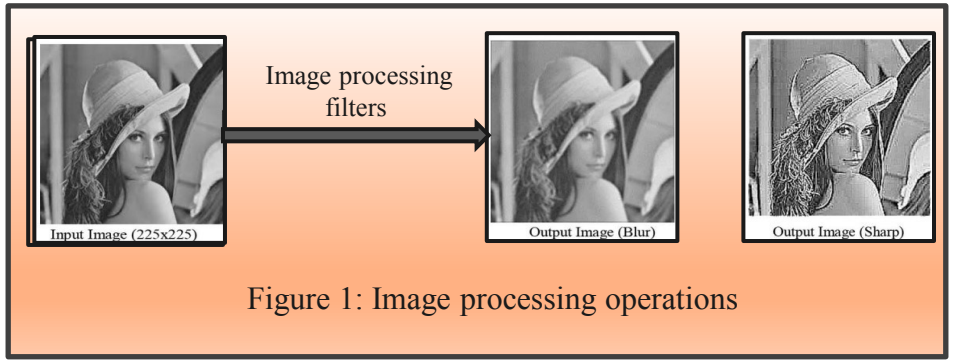
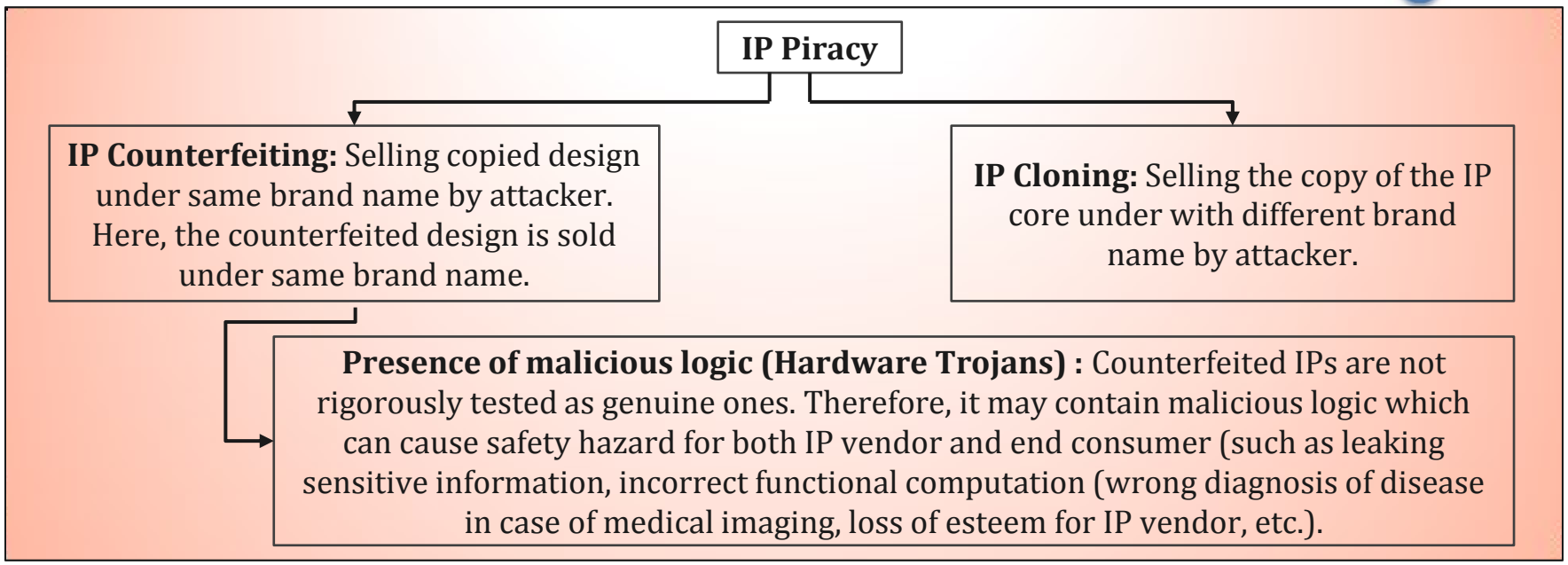


Figure 1: Image processing operations

Security Issues associated with image processing filter IP Cores [3], [4]



Fraudulent claim of IP ownership: An adversary tries to fraudulently claim the ownership of the IP.

Therefore, it is essential to secure these image processing filter IP cores from these hardware threats.

Related Work

Sr. No.	Existing Work	Technique Used	Remarks
1.	D. Tsiktsiris, D. Ziouzios, and M.Dasygenis [5] (2018)	Authors discusses about the implementation of FPGA based image processing accelerators.	Does not focus on the security aspects of image processing filter hardware IPs.
2.	C. Shu, W. Pang, H. Liu, and S. Lu [6] (2019)	The paper focuses on designing of hardware accelerators for performing convolutional neural network (CNN).	Does not provide a framework for designing image processing filter IP cores using high level synthesis. Further, it also does not includes the security aspects of image processing filter hardware IPs.
3.	F. Koushanfar, I. Hong, and M. Potkonjak [10] (2005) and Sengupta *et. al.,* [9] (2018)	Hardware watermarking using two-variable (0, 1) signature encoding process, and hardware steganography based security approaches.	Weak watermarking mechanism due to involvement of only two variable signature encoding process. Not robust and future proof. The watermark (original signature) inserted by watermarking technique becomes vulnerable if relevant information (like signature size, digit encoding, and digit combination) gets leaked. Further, hardware steganography becomes weak in case of a compromised threshold entropy value.

Proposed Work

- The proposed low-cost approach uses an encryption based security framework and PSO-driven design space exploration (PSO-DSE) for generating secure LED and embossment filters IP cores.
- The proposed approach uses the register allocation table of the image processing application (*i.e.,* LED and embossment filter) to generate secret data, which is fed as input to the proposed encryption based security framework to determine hardware security constraints .
- The generated hardware security constraints are then embedded in the image filter IP Core deigns to authenticate genuine IP vendor/maker using register allocation table (RAT) framework of HLS process.
- The huge variation in the key selection at different levels of encryption in the proposed encryption based security framework increases the robustness of the proposed hardware security methodology for image filters.

Detailed flow diagram of the proposed approach

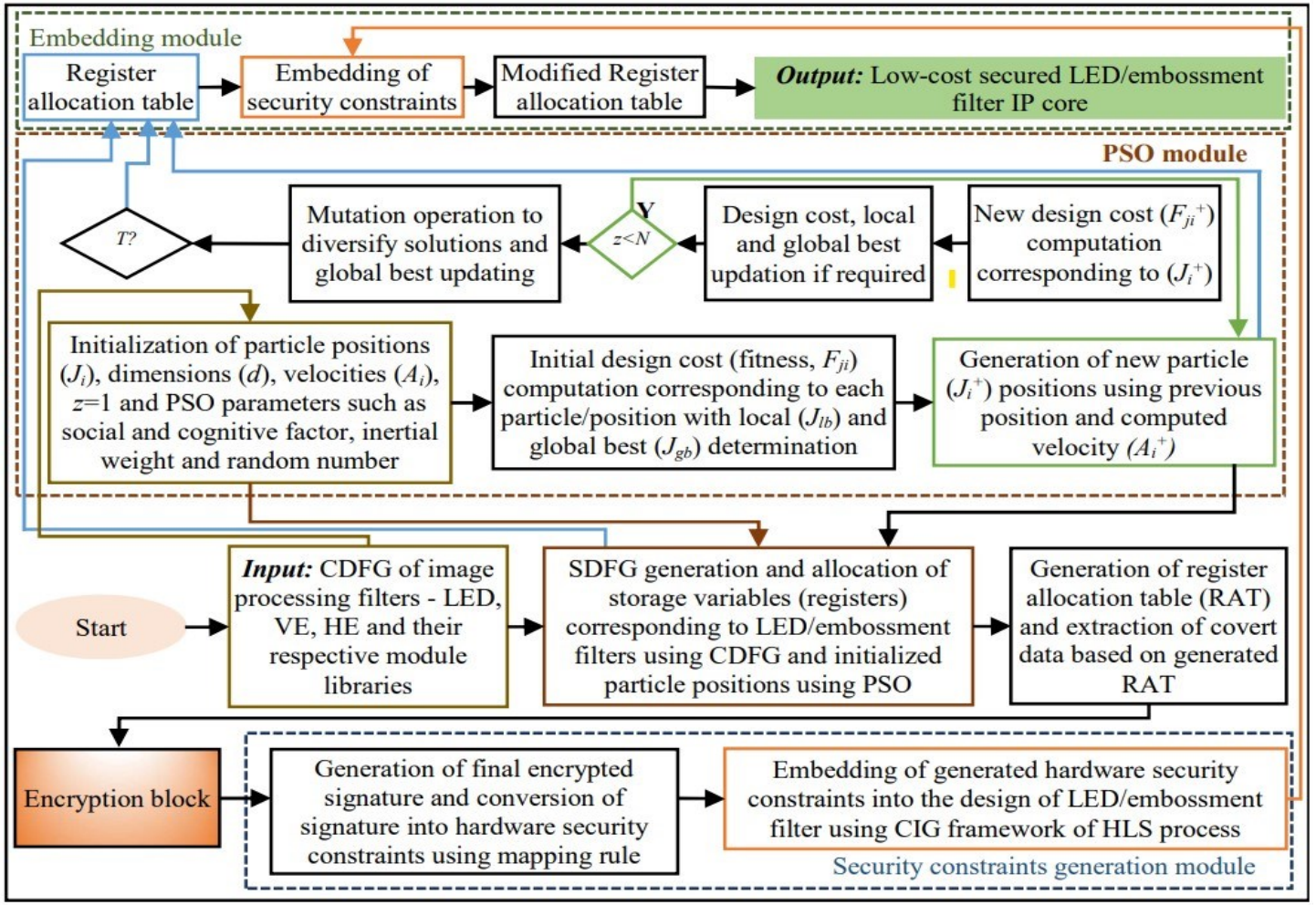


Figure 2: Flow diagram of the proposed methodology

Importance of particle swarm optimization based design space exploration (PSO-DSE) [8]

- The integration of the PSO-DSE block with the proposed security methodology serves the objective of determining an optimized architectural solution.
- PSO prunes the design search space based on IP vendor specified high level specification such as area, delay, energy, power, etc. corresponding to secured DSP design to generate an optimized low-cost design.

Advantage of PSO-DSE [8] over others such as genetic algorithm [9] and bacterial foraging [10] based DSE:

- PSO-DSE considers the magnitude of the previously computed velocity with the help of a parameter called inertia weight, while [7] and [11] do not consider the momentum of prior iterations, which increases the probability of getting stuck in the local minima during architecture exploration.
- PSO-DSE creates a balance between exploitation and exploration time with the help of linearly decreasing the value of inertia from 0.9 to 0.1. The algorithm takes more significant steps at the beginning and smaller steps on reaching higher fitness solutions, which is missing in [7]and [11]. This also enhances the chance of reaching global optimal solution.
- The inclusion of various other factors (hyperparameters), such as social and cognitive factors in PSO-DSE, helps achieve higher fitness solution within a very low exploration time

Details of the proposed encryption based security framework

Encryption process	Functions
Initial state grid generation	Generation of initial state grid using key 1 and covert data
Grid data manipulation	Substitution of state grid elements with their AES-128 bit s-box equivalents
Row transformation	Execution of circular shifts in each row of sate grid using key 2
TRIFID cipher calculation	TRIFID cipher is computed corresponding to each element of state grid
Alphabetic swapping	Each alphabet of grid is swapped with a numeric digit computed using key 3
Grid transposition	The final state grid generated is transformed
Grid data integration	Column wise concatenation is performed using key 4 on final state grid

Figure 3: Different stages of the proposed multi-stage encryption methodology

$$Kernel_{LED} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} Kernel_{HE} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} Kernel_{VE} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$H_0 = [(Q_{01}*(-1))] + [(Q_{10}*(-1)) + (Q_{11}*(4)) + (Q_{12}*(-1))] + [(Q_{21}*(-1))] \quad (1)$$

$$H_1 = [((Q_{02}*(-1))] + [(Q_{11}*(-1)) + (Q_{12}*(4)) + (Q_{13}*(-1))] + [(Q_{22}*(-1))] \quad (2)$$

Determination of secret design data based on scheduled data flow graph (SDFG) of LED filter

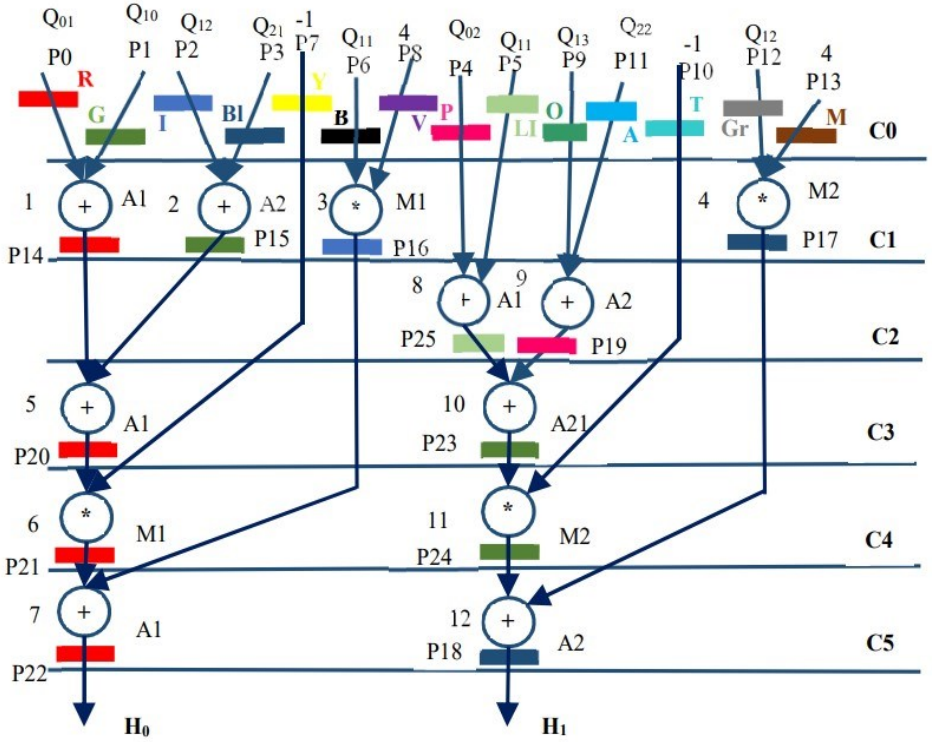


Figure 4: SDFG of 3*3 LED image filter using PSO-driven DSE explored two adders (+) and two multipliers (*)

	C0	C1	C2	C3	C4	C5
Red(R)	P0	P14/P15	P14/P15	P20/P23	P21	P22
Green (G)	P1	P15/P14	P15/P14	P23/P20	P24	P22
Indigo (I)	P2	P16/P17	P16/P17	P16/P17	P16/P17	-
Blue (BL)	P3	P17/P16	P17/P16	P17/P16	P17/P16	P18
Yellow (Y)	P7	P7	P7	P7	-	-
Black (B)	P6	-	-	-	-	-
Violet (V)	P8	-	-	-	-	-
Pink (P)	P4	P4	P19	-	-	-
Lime (LI)	P5	P5	P25	-	-	-
Olive (O)	P9	P9	-	-	-	-
Aqua (A)	P11	P11	-	-	-	-
Teal (T)	P10	P10	P10	P10	-	-
Gray (G)	P12	-	-	-	-	-
Maroon (M)	P13	-	-	-	-	-

Figure 5: Register allocation table (RAT) pre and post adding secret security constraints corresponding to LED filter

I = {(0,21),(17,18),(14,20),(1,24),(0,14),(0,22),(15,24),(0,20),(20,21),(21,22),(14,21),(1,15),(2,16),(20,22),(1,23),(14,22),(23,24),(15,26),(4,19),(3,18), (3,17),(5,25)}

Generation of initial state grid and implementation of proposed multi-stage encryption on generated state grid

I = {(0,6),(2,3),(E,5),(1,9),(0,E),(0,7),(0,9),(0,5),(5,6),(6,7),(E,6),(1,0),(2,1),(5,7),(1,8),(E,7),(8,9),(0,8)}.

Table 1
Initial state grid generation

06	23	E5	19
56	67	E6	10

Table 2
Data manipulated state grid

6F	26	D9	D4
B1	85	8E	CA

Table 3
Row transformed state grid

26	D9	D4	6F
CA	B1	85	8E

TRIFID Cipher Computation: Computing TRIFID cipher on "A":
Let IP vendor selected key: FTV$QEDRAYHUJIKOLPWSZMCBGXN.
Here, row number (p) is 3, column number (q) is 3, and square matrix (r) number is 1. The state corresponding to "A" is 331. Similarly, the state corresponding to the remaining alphabets is computed based on respective chosen key.

Square matrix 1			Square matrix 2			Square matrix 3		
F	T	V	Y	H	U	W	S	Z
$	Q	E	J	I	K	M	C	B
D	R	A	O	L	P	G	X	N

Generation of multi-stage encryption based signature

Table 4
State grid after alphabetic swapping

26	29	24	67
86	31	85	81

Table 5
State grid after grid transposition

26	86
29	31
24	85
67	81

- The generated final sequence after byte concatenation is: "266724298685813 1".
- The generated final signature through the proposed approach is:"10110110111101001010011000110100010110001111".
- The generated signature is mapped to its corresponding hardware security constraints as per the IP vendor selected mapping rule (if encoding bit of signature is '0' then embed an edge between (even, even) storage variable pair, otherwise embed an edge between (odd, odd) storage variable pair). The generated hardware security constraints are *(P0,P2),(P0,P4),---,(P0,P24),(P2,P4),---,(P2,P24),(P4,P6),(P1,P3),(P1,P5),---,(P1,P23),(P3,P5),(P3,P7),---,(P3,P23),(P5,P7).*

Evaluation (security) parameters:

- **Evaluation of Robustness Using Probability of Coincidence (Pc):**

$$Pc = \left(1 - \frac{1}{x}\right)^{z}$$

Where 'x' denotes the number of registers used in the CIG and 'z' denotes the number of hardware constraints added.

- **Tamper tolerance:**

$$TT = q^{t}$$

Where 'q' and 't' are types of encoding bits present in the mapping rule and strength (size) of generated security constraints respectively.

- **Design cost:**

$$Cost = t1 * \frac{Area\ - Ac}{Max\ area} + t2 * \frac{Latency\ - Lc}{Maximum\ latency}$$

Where 'area' and 'latency' represents the total area and latency (delay) of the proposed methodology-based secured IP core design; 'max area and max latency' depict the maximum area and latency of the proposed secured design of IP core using maximum resource constraints possible. 't1 and t2' are the weighing factors (weightage given to are and delay), which in the proposed approach is 0.5 each. *Ac* and *Lc* are IP designer specified design constraints value.

Comparison of Probability of coincidence and tamper tolerance between proposed, [9], and [10] along with design cost comparison before and after embedding security constraints

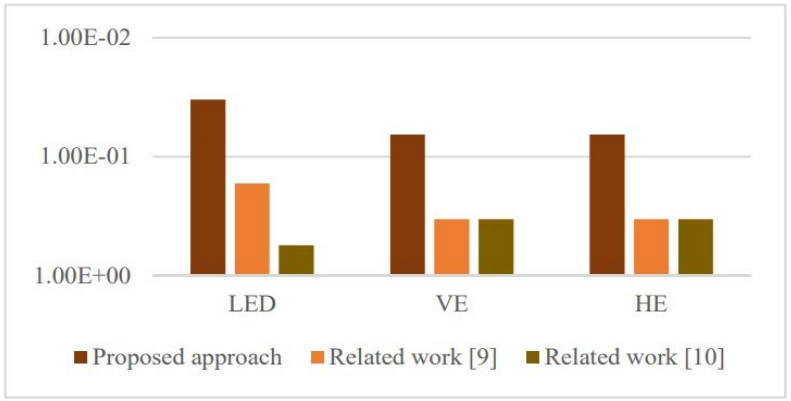


Figure 6: Probability of coincidence comparison between proposed, [9] and [10]

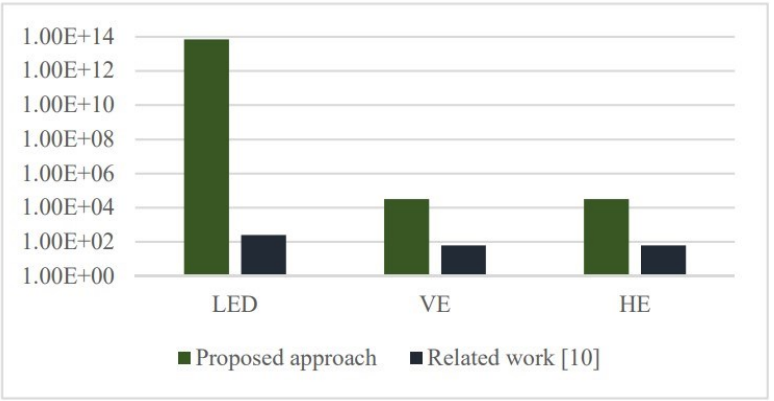


Figure 7: Tamper tolerance comparison between proposed and [10]

Table 6
Resource configuration, area, latency, cost, and of proposed low-cost hardware security methodology

Benchmarks	Explored Resource configuration	Baseline design (before signature embedding)			Signature embedded design			Design cost overhead %
		Design area (um^2)	Design latency (ps)	Design cost	Design area (um^2)	Design latency (ps)	Design cost	
Laplace edge detection filter	2(+), 2(*)	199.75	728.67	-0.108	199.75	728.67	-0.108	0
Horizontal embossment	1(+), 1(*)	99.09	596.18	-0.077	99.09	596.18	-0.077	0
Vertical embossment	1(+), 1(*)	99.09	596.18	-0.077	99.09	596.18	-0.077	0

References

1. R. Schneiderman, “DSPs evolving in consumer electronics applications,” *IEEE Signal Process. Mag*., vol. 27 (3), pp. 6–10, 2010.
2. F. Koushanfar et al., “Can EDA combat the rise of electronic counterfeiting?,” *IEEE DAC*, CA, 2012, pp. 133-138.
3. A. Sengupta, “Hardware security of CE devices,” *IEEE Consum. Electron. Mag*., vol. 6, no. 1, pp. 130–133, Jan. 2017.
4. A. Sengupta, D. Roy, S. P. Mohanty, and P. Corcoran, “Low-cost obfuscated JPEG CODEC IP core for secure CE hardware,” *IEEE Trans. Consum. Electron*., vol. 64, no. 3, pp. 365–374, Aug. 2018.
5. D. Tsiktsiris, D. Ziouzios, and M. Dasygenis, “A portable image processing accelerator using FPGA,” in Proc. *MOCAST*, 2018, pp. 1–4.
6. C. Shu, W. Pang, H. Liu, and S. Lu, “High energy efficiency FPGA based accelerator for convolutional neural networks using weight combination,” in Proc. *ICSIP*, Wuxi, China, 2019, pp. 578–582.
7. V. Krishnan and S. Katkoori, "A genetic algorithm for the design space exploration of datapaths during high-level synthesis," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 213-229, June 2006.
8. Vipul Kumar Mishra, Anirban Sengupta, MO-PSE: Adaptive multi objective particle swarm optimization based design space exploration in architectural synthesis for application specific processor design, *Advances in Engineering Software*, Vol 67,2014,pp. 111- 124.
9. A. Sengupta and M. Rathor, "IP Core Steganography for Protecting DSP Kernels Used in CE Systems," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 4, pp. 506-515, Nov. 2019.
10. F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," *ACM Trans. Design Autom. Electron. Syst*., vol. 10, no. 3, pp. 523–545, Jul. 2005.
11. A. Sengupta and S. Bhadauria, "Automated exploration of datapath in high level synthesis using temperature dependent bacterial foraging optimization algorithm," 2014 *IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE),* 2014, pp. 1-5.

Thank You!