IP Core Protection of Image Processing Filters With Multi-Level Encryption and Covert Steganographic Security Constraints

Paper ID: 1570843785

Track: Hardware for Secure Information Processing (SIP) - 1

Authors: Aditya Anshul, Anirban Sengupta

Speaker Name: Aditya Anshul

Introduction

Image processing filters

- Image processing filters are mainly used to suppress either the high frequencies in the image, i.e. smoothing the image, or the low frequencies, i.e. enhancing or detecting edges in the image.
- 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 a dedicated intellectual property (IP) core involves various hardware threats [1], [2].

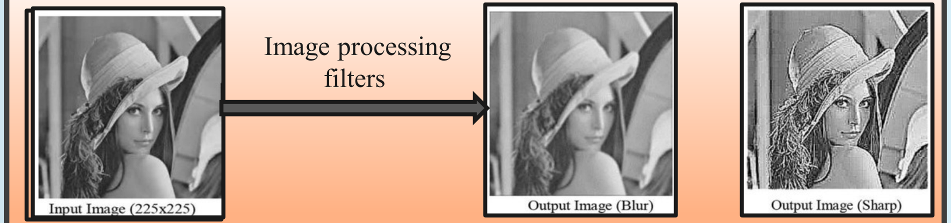
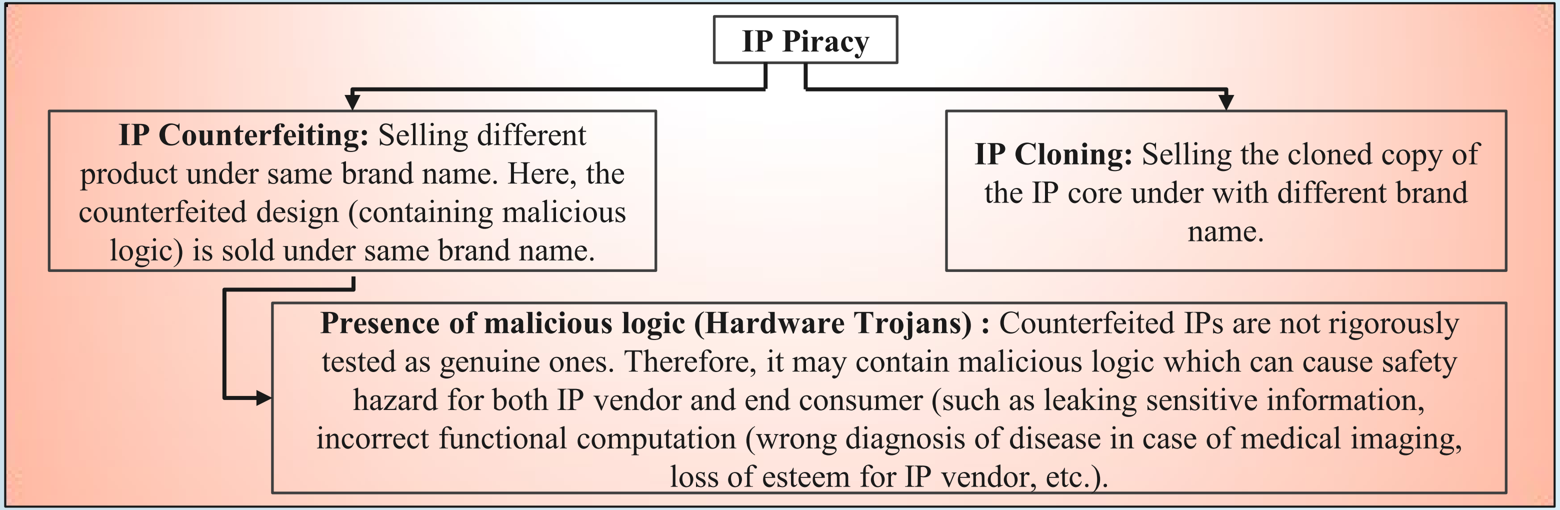


Figure 1: Image processing operations

Introduction : Hardware Threats

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



IP Piracy

IP Counterfeiting: Selling different product under same brand name. Here, the counterfeited design (containing malicious logic) is sold under same brand name.

IP Cloning: Selling the cloned copy of the IP core under with different brand name.

Presence of malicious logic (Hardware Trojans) : Counterfeited IPs are not rigorously tested as genuine ones. Therefore, it may contain malicious logic which can cause safety hazard for both IP vendor and end consumer (such as leaking sensitive information, incorrect functional computation (wrong diagnosis of disease in case of medical imaging, loss of esteem for IP vendor, etc.).

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.

Previous works

Related Work

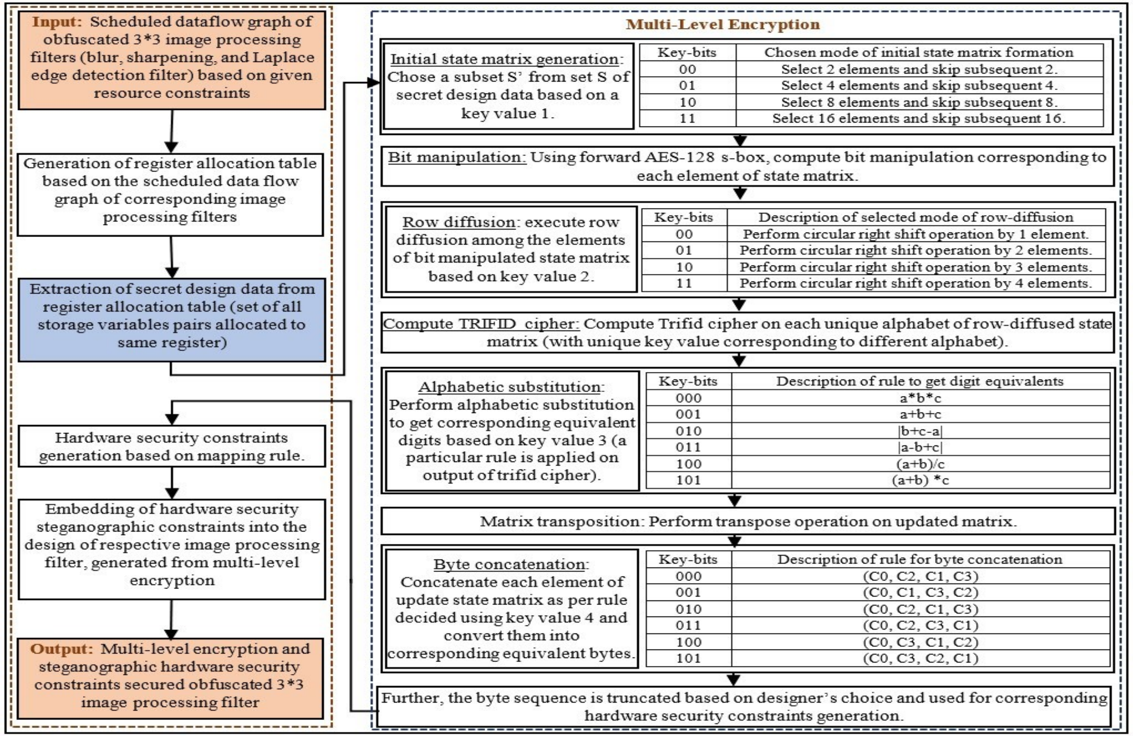
Sr. No.	Existing Work	Technique Used	Remarks
1.	D. Tsiktsiris, D. Ziouzios, and M.Dasygenis [5] (2018)	Authors discusses about the implementation of FPGA based image processing accelerators.	Does not focus on the security aspects of image processing filter hardware IPs.
2.	C. Shu, W. Pang, H. Liu, and 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 [7] (2005)	Hardware watermarking using two-variable (0, 1) signature encoding process.	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.

Proposed Work

- The proposed approach based on multi-level encryption technique used for securing image processing filters IP cores. This paper discusses signature-based hardware security methodology on image processing IP cores for the first time.
- The proposed approach uses the register allocation table of the image processing application to generate secret data, which is used for multi-level encryption to determine hardware security constraints .
- The generated hardware security constraints then embedded in the image processing filter IP Cores design to authenticate genuine IP Maker.
- The huge variation in the key selection at different levels of encryption and the use of secret steganographic data for signature generation increases the robustness of the proposed hardware security methodology for image filters.

Proposed Work : Flow Diagram

Detailed flow diagram of the proposed approach



Input: Scheduled dataflow graph of obfuscated 3*3 image processing filters (blur, sharpening, and Laplace edge detection filter) based on given resource constraints

Generation of register allocation table based on the scheduled data flow graph of corresponding image processing filters

Extraction of secret design data from register allocation table (set of all storage variables pairs allocated to same register)

Hardware security constraints generation based on mapping rule.

Embedding of hardware security steganographic constraints into the design of respective image processing filter, generated from multi-level encryption

Output: Multi-level encryption and steganographic hardware security constraints secured obfuscated 3*3 image processing filter

Multi-Level Encryption

Initial state matrix generation: Chose a subset S' from set S of secret design data based on a key value 1.

Key-bits	Chosen mode of initial state matrix formation
00	Select 2 elements and skip subsequent 2.
01	Select 4 elements and skip subsequent 4.
10	Select 8 elements and skip subsequent 8.
11	Select 16 elements and skip subsequent 16.

Bit manipulation: Using forward AES-128 s-box, compute bit manipulation corresponding to each element of state matrix.

Row diffusion: execute row diffusion among the elements of bit manipulated state matrix based on key value 2.

Key-bits	Description of selected mode of row-diffusion
00	Perform circular right shift operation by 1 element.
01	Perform circular right shift operation by 2 elements.
10	Perform circular right shift operation by 3 elements.
11	Perform circular right shift operation by 4 elements.

Compute TRIFID cipher: Compute Trifid cipher on each unique alphabet of row-diffused state matrix (with unique key value corresponding to different alphabet).

Alphabetic substitution: Perform alphabetic substitution to get corresponding equivalent digits based on key value 3 (a particular rule is applied on output of trifid cipher).

Key-bits	Description of rule to get digit equivalents
000	a*b*c
001	a+b+c
010	|b+c-a|
011	|a-b+c|
100	(a+b)/c
101	(a+b) *c

Matrix transposition: Perform transpose operation on updated matrix.

Byte concatenation: Concatenate each element of update state matrix as per rule decided using key value 4 and convert them into corresponding equivalent bytes.

Key-bits	Description of rule for byte concatenation
000	(C0, C2, C1, C3)
001	(C0, C1, C3, C2)
010	(C0, C2, C1, C3)
011	(C0, C2, C3, C1)
100	(C0, C3, C1, C2)
101	(C0, C3, C2, C1)

Further, the byte sequence is truncated based on designer's choice and used for corresponding hardware security constraints generation.

Fig.2. Flow-chart of proposed methodology for securing image processing filters using multi-level encryption and steganographic hardware security constraints.

Proposed Work : Extraction of secret design data from SDFG of image processing application

Determination of secret design data based on scheduled data flow graph of image processing application

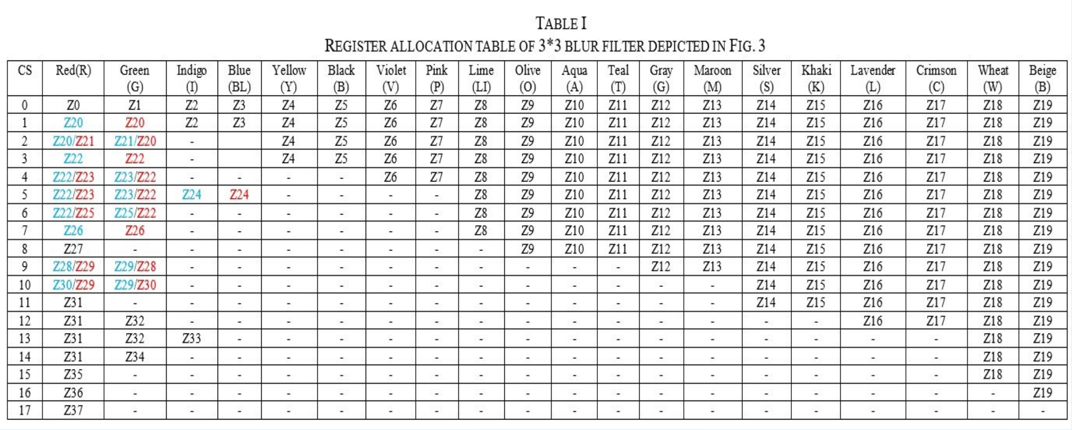


TABLE I

REGISTER ALLOCATION TABLE OF 3*3 BLUR FILTER DEPICTED IN FIG. 3

CS	Red(R)	Green (G)	Indigo (I)	Blue (BL)	Yellow (Y)	Black (B)	Violet (V)	Pink (P)	Lime (LI)	Olive (O)	Aqua (A)	Teal (T)	Gray (G)	Maroon (M)	Silver (S)	Khaki (K)	Lavender (L)	Crimson (C)	Wheat (W)	Beige (B)
0	Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
1	Z20	Z20	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
2	Z20/Z21	Z21/Z20	-		Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
3	Z22	Z22	-		Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
4	Z22/Z23	Z23/Z22	-	-	-	-	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
5	Z22/Z23	Z23/Z22	Z24	Z24	-	-	-	-	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
6	Z22/Z25	Z25/Z22	-	-	-	-	-	-	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
7	Z26	Z26	-	-	-	-	-	-	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
8	Z27	-	-	-	-	-	-	-	-	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
9	Z28/Z29	Z29/Z28	-	-	-	-	-	-	-	-	-	-	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
10	Z30/Z29	Z29/Z30	-	-	-	-	-	-	-	-	-	-	-	-	Z14	Z15	Z16	Z17	Z18	Z19
11	Z31	-	-	-	-	-	-	-	-	-	-	-	-	-	Z14	Z15	Z16	Z17	Z18	Z19
12	Z31	Z32	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Z16	Z17	Z18	Z19
13	Z31	Z32	Z33	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Z18	Z19
14	Z31	Z34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Z18	Z19
15	Z35	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Z18	Z19
16	Z36	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Z19
17	Z37	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

S = {(0,20), (0,22), (0,26), (0,27), (0,28), (0,30), (0,31), (0,35), (0,36), (0,37), (20,22), (20,26), (20,27), (20,28), (20,30), (20,31), (20,35), (20,36), (20,37), (22,26), (22,27), (22,28), (22,30), (22,31), (22,35), (22,36), (22,37), (26,27), (26,28), (26,30), (26,31), (26,35), (26,36), (26,37), (27,28), (27,30), (27,31), (27,35), (27,36), (27,37), (28,30), (28,31), (28,35), (28,36), (28,37), (30,31), (30,35), (30,36), (30,37), (31,35), (31,36), (31,37), (34,36), (35,37), (36,37), (1,21), (1,23), (1,25), (1,29), (1,32), (1,34), (21,23), (21,25), (21,29), (21,32), (21,34), (23,25), (23,29), (23,32), (23,34), (25,29), (25,32), (25,34), (29,32), (29,34), (32,34), (2,24), (2,33), (33,24)}.

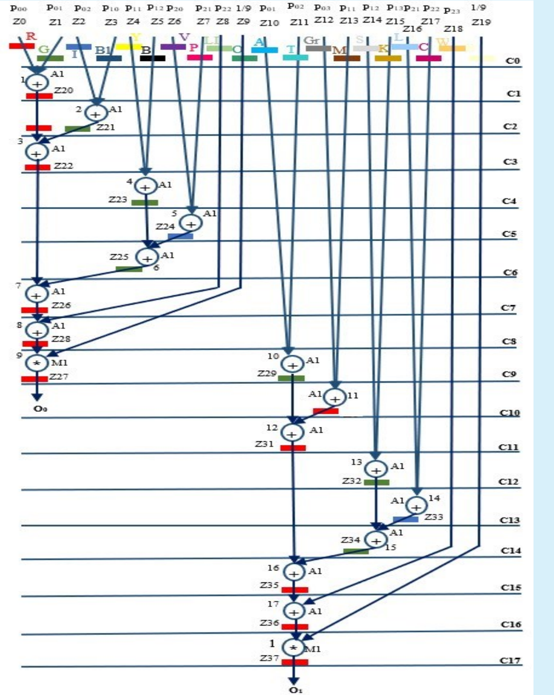


Fig.3. Scheduled data flow graph of 3*3 blur filter with 1(+), 1(-) and 2(*) as resource constraints

IEEE (®)computer society

The IEEE Computer Society Technical Committee on VLSI

Proposed Work : Implementation of multi-level encryption

Generation of initial state matrix and implementation of multi-level encryption on generated state matrix

S = {(0,5), (0,7), (0,B), (0,C), (0,D), (0,1), (0,5), (0,6), (5,7), (5,B), (5,C), (5,D), (5,1), (5,6), (5,7), (7,B), (7,C), (7,D), (7,1), (7,6), (B,C), (B,D), (B,1), (B,6), (C,D), (C,1), (C,6), (D,1), (D,6), (1,6), (1,8), (1,A), (1,E), (1,2), (1,4), (6,8), (6,A), (6,E), (6,2), (6,4), (8,A), (8,E), (8,2), (8,4), (A,E), (A,2), (A,4), (E,2), (E,4), (2,4), (2,3), (3,9)}.

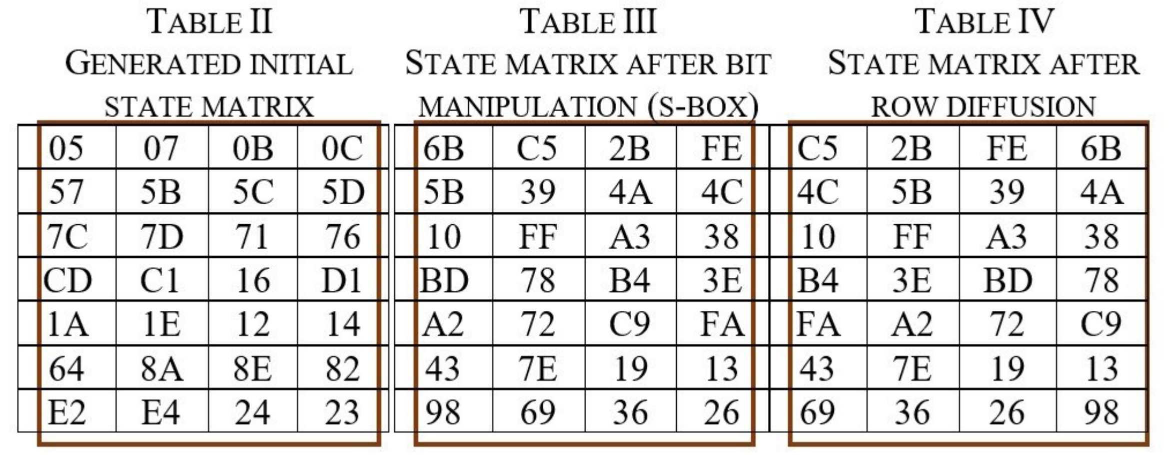


TABLE II
GENERATED INITIAL STATE MATRIX

05	07	0B	0C
57	5B	5C	5D
7C	7D	71	76
CD	C1	16	D1
1A	1E	12	14
64	8A	8E	82
E2	E4	24	23

TABLE III
STATE MATRIX AFTER BIT MANIPULATION (S-BOX)

6B	C5	2B	FE
5B	39	4A	4C
10	FF	A3	38
BD	78	B4	3E
A2	72	C9	FA
43	7E	19	13
98	69	36	26

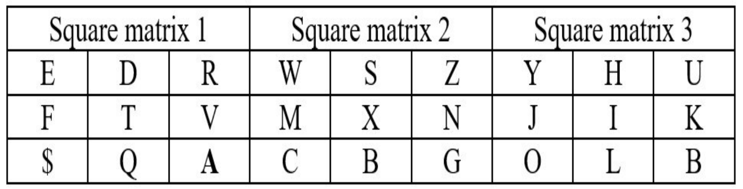
TABLE IV
STATE MATRIX AFTER ROW DIFFUSION

C5	2B	FE	6B
4C	5B	39	4A
10	FF	A3	38
B4	3E	BD	78
FA	A2	72	C9
43	7E	19	13
69	36	26	98

Proposed Work : Multi-level encryption

Trifid cipher computation and alphabetic substitution

- Computing TRIFID cipher on "A":
- Let IP vendor selected key: EDRFTV$QAWSZMXNCBGYHUJIKOLP
- Here, row number (a) is 3, column number (b) is 3, and square matrix (c) number is 1. The state corresponding to "A" is 331. Similarly, the state corresponding to the remaining alphabets is computed based on chosen key.



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

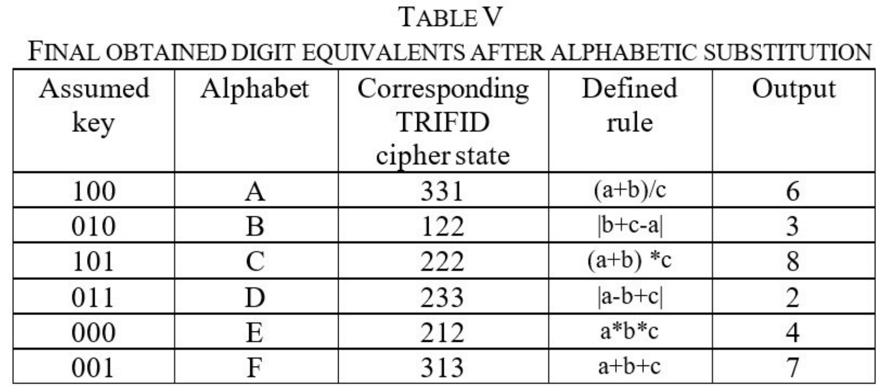


TABLE V

FINAL OBTAINED DIGIT EQUIVALENTS AFTER ALPHABETIC SUBSTITUTION

Assumed key	Alphabet	Corresponding TRIFID cipher state	Defined rule	Output
100	A	331	(a+b)/c	6
010	B	122	|b+c-a|	3
101	C	222	(a+b) *c	8
011	D	233	|a-b+c|	2
000	E	212	a*b*c	4
001	F	313	a+b+c	7

Proposed Work : Signature generation

Generation of multi-level encryption based signature

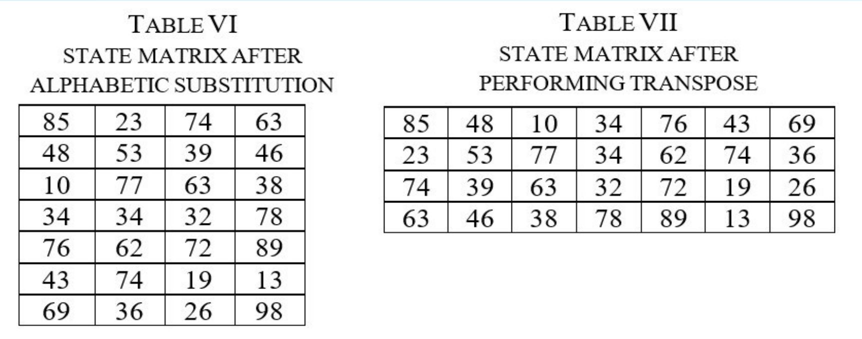


TABLE VI
STATE MATRIX AFTER
ALPHABETIC SUBSTITUTION

85	23	74	63
48	53	39	46
10	77	63	38
34	34	32	78
76	62	72	89
43	74	19	13
69	36	26	98

TABLE VII
STATE MATRIX AFTER
PERFORMING TRANSPOSE

85	48	10	34	76	43	69
23	53	77	34	62	74	36
74	39	63	32	72	19	26
63	46	38	78	89	13	98

- The generated final sequence after byte concatenation is: "857423634846395310776338347834327662897243191374369 362698".
- The generated final signature through the proposed approach is:"10001011111001011110111001000100110111001101 1110111111101111100011100111100011100111011111011010100010011111010011110011111110011 0100111110101101001100 0" .
- 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 <Z0,Z2>, <Z0,Z4>, <Z0,Z6>,-------,<Z12,Z16>, <Z12,Z18>, <Z1,Z3>, <Z1,Z5>,-------- ,<Z7,Z19>, <Z7,Z21>.

Security metrics

Evaluation parameters

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

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

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

- **Tamper tolerance:**

$$\mathrm{TT} = q^{t}$$

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

- **Design cost:**

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

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.

Comparative Analysis

Comparison of Probability of coincidence (P) between proposed and [7] and design cost comparison before and after embedding security constraints

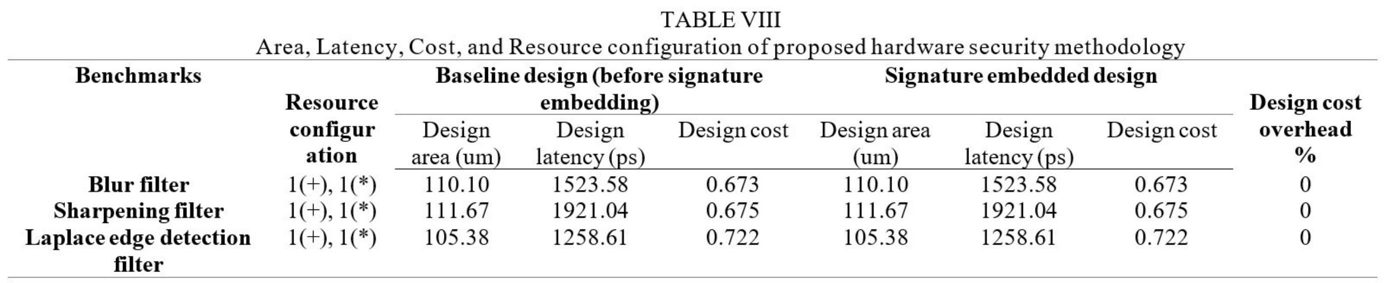


TABLE VIII
Area, Latency, Cost, and Resource configuration of proposed hardware security methodology

Benchmarks	Resource configuration	Baseline design (before signature embedding)			Signature embedded design			Design cost overhead %
		Design area (um)	Design latency (ps)	Design cost	Design area (um)	Design latency (ps)	Design cost	
Blur filter	1(+), 1(*)	110.10	1523.58	0.673	110.10	1523.58	0.673	0
Sharpening filter	1(+), 1(*)	111.67	1921.04	0.675	111.67	1921.04	0.675	0
Laplace edge detection filter	1(+), 1(*)	105.38	1258.61	0.722	105.38	1258.61	0.722	0

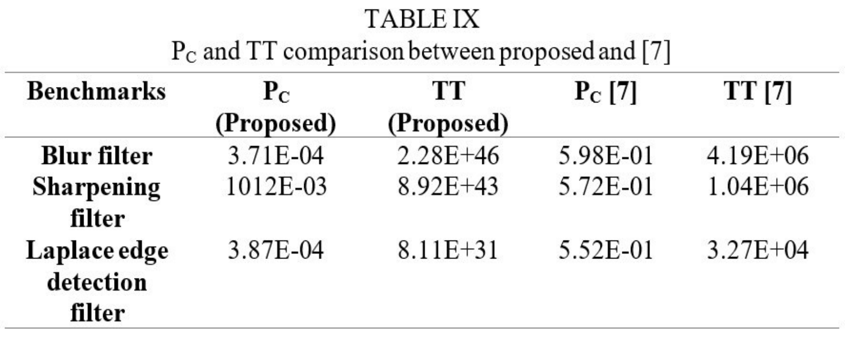


TABLE IX
P_C and TT comparison between proposed and [7]

Benchmarks	P_C (Proposed)	TT (Proposed)	P_C [7]	TT [7]
Blur filter	3.71E-04	2.28E+46	5.98E-01	4.19E+06
Sharpening filter	1012E-03	8.92E+43	5.72E-01	1.04E+06
Laplace edge detection filter	3.87E-04	8.11E+31	5.52E-01	3.27E+04

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. 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.

Thank You