Indian Institute of Technology Kharagpur

Open Soft Problem Statement

February 23, 2016

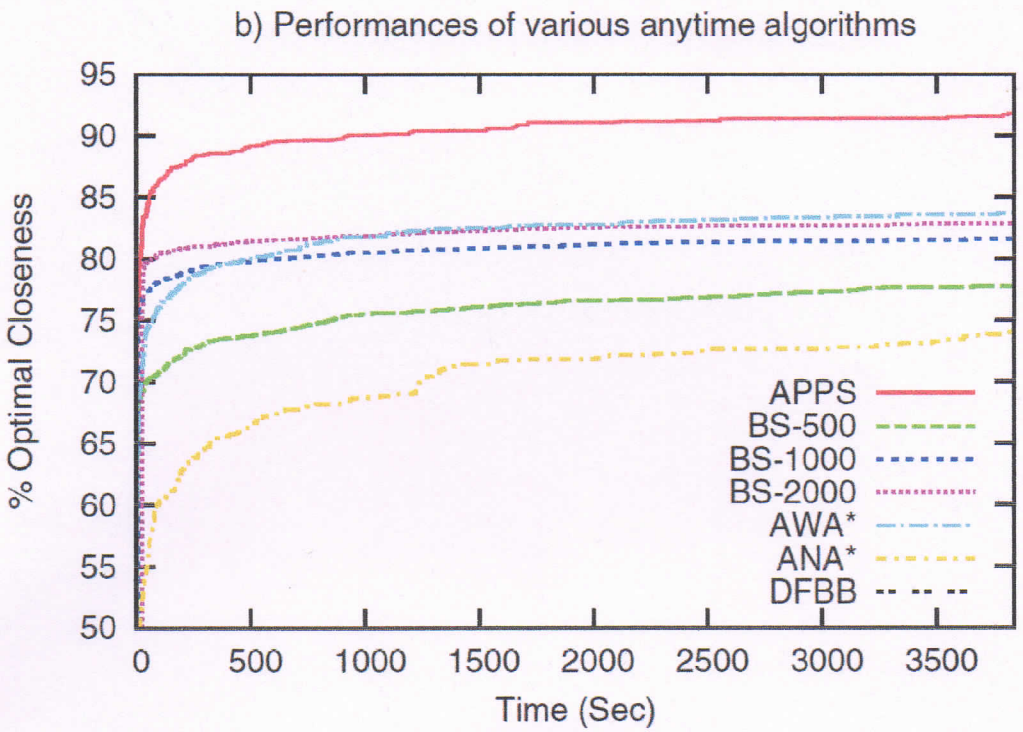
Plots to Tables

The input is a set of (multiple) scanned pdf pages, each containing (one or more) figures which are two-dimensional plots of experimental results. A scanned page can have a single or multiple plots which may or may not be embedded in text. Each plot has the x and y axis with their labels and unit measurements marked in the plot (linear scale). Inside each figure are one or more plots, each with a different colour depicting a certain plotted entity (E_i) and their labels given separately within the plot as a caption. (The example enclosed shows such an input set.)

You are required to read a set of scanned pages as input where each page has one or more plots embedded in text and convert them to a set of two-dimensional tables, one table per plot, where each row of the table has the following values - the x-axis value, y-axis value and values for each E_i or a dash (-) in case there is no value for that E_i. Each Table should have the first row as the name labels for the x-axis, y-axis and various E_i values. It should cover x and y axis values from the minimum to maximum range with one tenth of the minimum marked unit in the plot as granularity. The Table as a whole will have a caption as per the caption of the figure.

The output will be a set of pdf pages which contain the name of the participant as the first page followed by a sequence of results having the first input page followed by the set of tables corresponding to the figures in that page (one table per page) followed by the next input page and the tables of that page, etc.

An example set of input pages is enclosed.


b) Performances of various anytime algorithms
% Optimal Closeness
95
90
85
80
75
70
65
60
55
50
0
500
1000
1500
2000
2500
3000
3500
Time (Sec)
APPS
BS-500
BS-1000
BS-2000
AWA*
ANA*
DFBB

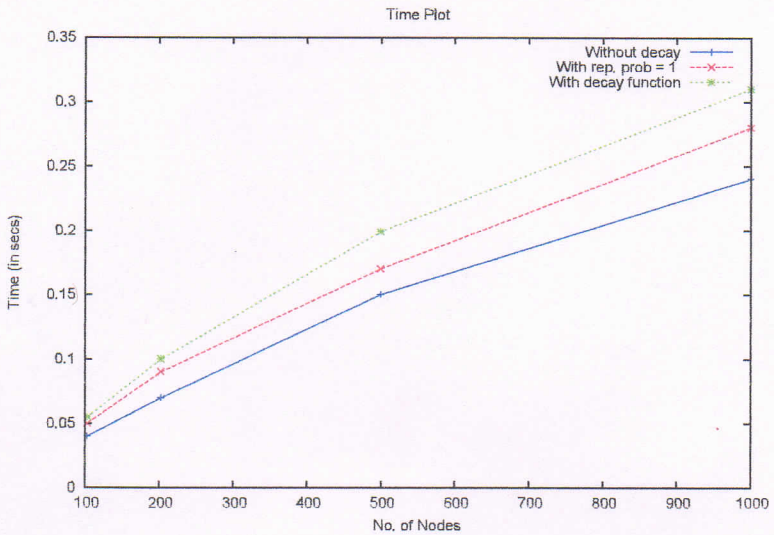
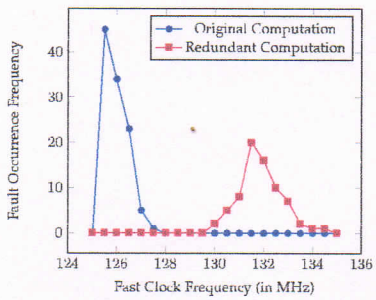


Figure 3.3: Variation of metric computation time (in sec) with size of attack graphs. Memoryless, partial memory and full memory of attacker for repeated vulnerabilities are considered. Time is reported in seconds.

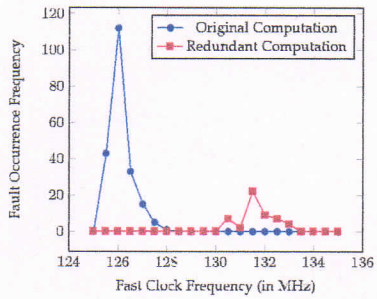
Workstation (8 GB memory). It can be seen that the the growth of computational time is sub-exponential. The computation time is highest for the case of partial memory attacker scenario.

3.5 Conclusion

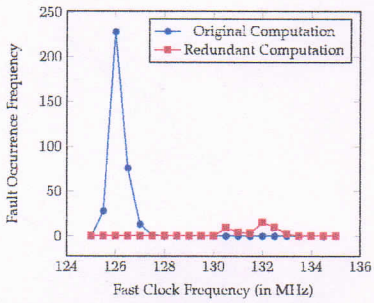
Security analysis is a challenging problem due to inherent complexities of attack modalities, scale and computational cost. We present a structured framework for probabilistic security metric computation using an multiplicative idempotency operation that can handle repeated vulnerabilities in an attack path. Proof of correctness and complexity analysis of security metric computation are provided. The metric is then extended to model the scenario where attackers have (i) full memory of previous exploits, (ii) partial memory of repeated vulnerabilities as characterized by a decay function, and (iii) no memory of past exploits. The metrics are then used for computing vulnerabilities of large attack graphs having cycles and repeated vulnerabilities. Scalability of the propose method with increasing network size is studied.



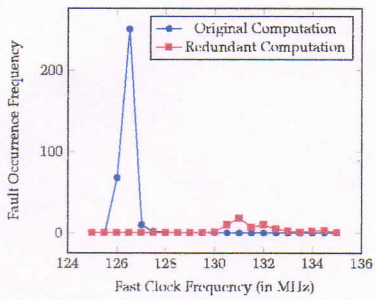
(a) Fault Space Transformation : SBU



(b) Fault Space Transformation : SBDBU

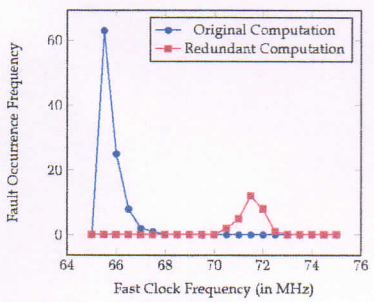


(c) Fault Space Transformation : SBTBU

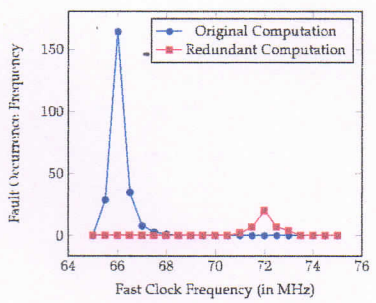


(d) Fault Space Transformation : SBQBU

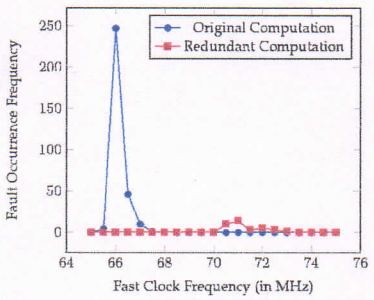
Fig. 9: Effect of Fault Space Transformation on the Time Redundancy Countermeasure



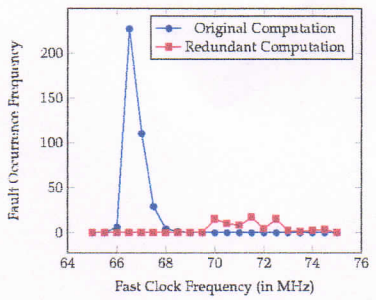
(a) Fault Space Transformation : SBU



(b) Fault Space Transformation : SBDBU



(c) Fault Space Transformation : SBTBU



(d) Fault Space Transformation : SBQBU

Fig. 10: Effect of Fault Space Transformation on the Hardware Redundancy Countermeasure

TABLE 3: Fault Distribution

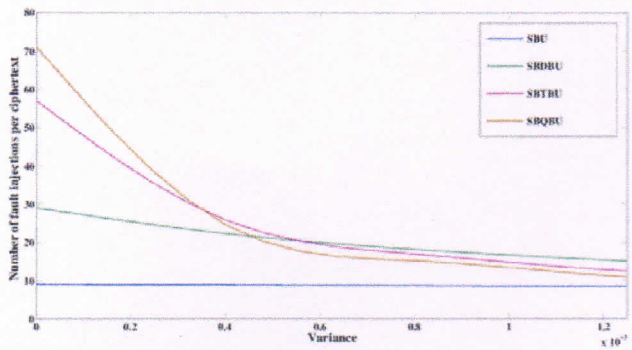
(a) Fault Distribution - Time Redundancy

Fast Clock Frequency (MHz)	FF	SBU	SBDBU	SBTBU	SBQBU	OSB	MB
125.0	512	0	0	0	0	0	0
125.1	503	9	0	0	0	0	0
125.2	489	22	1	0	0	0	0
125.3	456	50	6	0	0	0	0
125.4	425	59	22	6	0	0	0
125.5	396	45	43	28	0	0	0
125.6	351	34	112	32	0	0	0
125.7	303	23	101	85	0	0	0
125.8	260	11	55	86	0	0	0
125.9	208	5	46	147	6	0	0
126.0	176	1	39	228	68	0	0
126.1	143	0	18	211	136	4	0
126.2	115	0	10	94	178	15	0
126.3	101	0	8	95	251	49	8
126.4	65	0	9	45	232	141	20
126.5	32	0	5	16	131	187	141
126.6	13	0	3	8	98	101	289
126.7	5	0	1	4	32	112	358
126.8	0	0	1	2	5	105	399
126.9	0	0	1	2	3	88	421
127.0	0	0	0	1	2	33	476
127.1	0	0	0	0	1	12	499
127.2	0	0	0	0	0	0	512
127.3	0	0	0	0	0	0	512
127.4	0	0	0	0	0	0	512
127.5	0	0	0	0	0	0	512

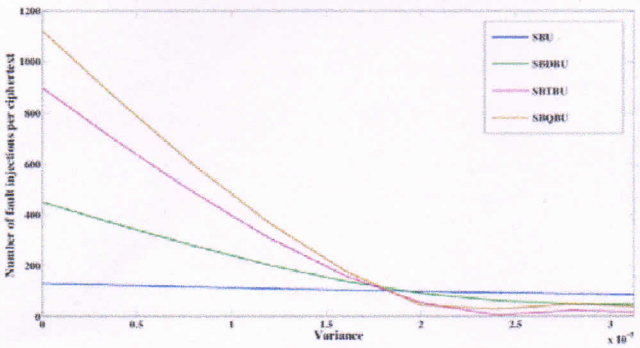
(b) Fault Distribution - Hardware Redundancy

Fast Clock Frequency (MHz)	FF	SBU	SBDBU	SBTBU	SBQBU	OSB	MB
70.0	512	0	0	0	0	0	0
70.1	512	0	0	0	0	0	0
70.2	504	8	0	0	0	0	0
70.3	475	34	3	0	0	0	0
70.4	460	47	5	0	0	0	0
70.5	416	63	29	4	0	0	0
70.6	378	38	71	25	0	0	0
70.7	345	29	120	32	0	0	0
70.8	293	21	164	28	0	0	0
70.9	234	14	120	144	2	0	0
71.0	216	4	39	247	6	0	0
71.1	189	2	35	220	66	0	0
71.2	130	0	15	180	176	11	0
71.3	105	0	10	104	278	15	0
71.4	83	0	10	66	227	100	26
71.5	50	0	8	46	157	162	90
71.6	27	0	5	16	113	125	226
71.7	21	0	4	10	98	118	261
71.8	13	0	3	6	50	103	337
71.9	7	0	3	5	21	107	369
72.0	5	0	3	2	10	99	393
72.1	2	0	1	1	8	44	456
72.2	1	0	0	1	6	19	485
72.3	1	0	0	0	2	8	501
72.4	0	0	0	0	1	5	506
72.5	0	0	0	0	0	0	512

Fig. 4: Number of Fault Attacks per Faulty Ciphertext vs Variance of Fault Probability Distribution



(a) Adversary has perfect control over target byte



(b) Adversary has no control over target byte

recover the full key under different fault models. In the second half, we vary the probability distribution for each fault model to confirm the correlation of the bias with the fault collision probability, as described by Equation 2. We quantify the bias of the fault model using the variance of the fault probability distribution, and the fault collision probability by the number of fault injections required per faulty ciphertext.

TABLE 6: Number Of Faulty Ciphertexts Required To Guess the Entire Key With 99% Probability

Round	Fault Model	N_C
8	SBU	320-340
	SBDBU	580-600
	SBTBU	1000-1040
	SBQBU	1900-2000
9	SBU	288-320
	SBDBU	608-640
	SBTBU	832-880
	SBQBU	1360-1440

5.3.1 Simulation: Part-1

In this part of the simulation, we assume identical faults in both the original and redundant computation rounds and aim to estimate the average number of faulty ciphertexts required to recover the entire key. Note that since the actual attack procedure is independent of the countermeasure scheme being targeted (time or hardware redundancy), the simulation results are presented for a general attack on either countermeasure scheme.

In the simulation, a byte of the state at the desired attack point is chosen at random and then fault is introduced into a certain number of bits belonging to that byte, varying from 1 to 4. Note that these bits are also chosen at random. We simulate the attacks in rounds 8 and 9 respectively. In each case, the appropriate distinguisher function is used to choose the key hypothesis. Table 6 summarizes the number of faulty ciphertexts required for each fault model to guess the entire 128-bit key with 99% accuracy for the attacks on rounds 8 and 9.

5.3.2 Simulation: Part-2

In the second half of the simulation, we varied the degree of bias for each fault model by controlling the variance of the