Department of Computer Science 12
(Hardware-Software-Co-Design)
Friedrich-Alexander University Erlangen-Nürnberg
Prof. Dr.-Ing. Jürgen Teich

Examination
Security in Embedded Hardware

12.10.2018

Name	
Matriculation Number	
Subject of Study	

Problem	1	2	3	Σ
Max. Points	22	24	22	68
Achieved Points				
Grade				

Organisatorische Hinweise

Bitte sorgfältig lesen und die Kenntnisnahme durch Unterschrift bestätigen

---

1. Bitte legen Sie Ihren Studentenausweis bereit.
2. Als Hilfsmittel sind nur Schreibmaterialien zugelassen.
3. Schmierpapier wird nicht abgegeben und nicht korrigiert.
4. Sie können bei der Aufsicht zusätzliche Bearbeitungsblätter anfordern. Diese müssen Ihrer Arbeit angeheftet werden.
5. Unleserliches wird nicht bewertet.

ERKLÄRUNG

1. Im Falle einer während der Prüfung auftretenden Prüfungsunfähigkeit zeige ich dies sofort der Aufsicht an und befolge deren Anweisungen. Ich weiß, dass ich die volle Beweislast trage. Ich lasse mir das Formular des Prüfungsamts, das für diese Fälle vorgesehen ist, aushändigen und verfahre nach den dort niedergelegten Richtlinien.
2. Ich weiß, dass im Falle des Täuschungsversuchs oder der Benutzung unerlaubter Hilfsmittel („Unterschleif") der Prüfungsausschuss die Entscheidung treffen kann, die betroffene Prüfungsleistung als mit „nicht ausreichend" bewertet gelten zu lassen.
3. Ich habe die obigen Hinweise zur Kenntnis genommen.

Erlangen, den 12.10.2018 .....

Unterschrift

Problem 1 (Introduction) (22 Points)

1. Explain the two *attributes of dependability* availability and reliability. Which measures or metrics exist for evaluating them? (4 Points)

2. Name the other attributes of dependability. (4 Points)

3. What are the means to attain dependability? (4 Points)

4. Is *security* also an attribute of dependability? What is the relation between *security* and the attributes of dependability. (3 Points)

5. Explain the error chain (*fault, error, failure*) with the example of a bit being flipped due to cosmic rays. (3 Points)

6. Cryptographic algorithms can be categorized by the way input data is grouped and encoded. Name the corresponding categories. (2 Points)

7. Decrypt the following text that was encrypted by the Caesar cipher: (2 Points)

   f d h v d u

Problem 2 (Control Flow Checking) (24 Points)

1. Discuss the purpose and functionality of *canary words* in the context of code injection attacks. (4 Points)

2. The following code is to be analyzed:

```
1  double sehFunc (double f, double g) {
2    double a,b,c;
3    double d = 3.0;
4
5     if (f == c)
6          a = b;
7
8
9    if (f == g) {
10     a = 2.0 + g;
11     b = 5*a - 2;
12     if (g == 9.0)
13       c = a;
14     if (b == 3)
15       a = 42.0;
16   } else {
17       a = f;
18       c =
19       d =  .0;
20   }
21
22   return  c;
23 }
24
```

a) How many different program paths do exist? (4 Points)

b) Name all, if any, variables that might remain uninitialized. (2 Points)

3. What is the difference between Control Flow Graphs (CFG) and Control Flow Instruction Graphs (CFIG)? (2 Points)

4. Explain how these graphs can be used to improve the security of program execution. (2 Points)

5. Draw both the CFG and CFIG for the following assembler code:

```
1  40001270 <incr>:
2  40001270:  save %sp, -104, %sp
3  40001274:  inc  %g3
4  40001278:  ret
5  4000127c:  restore
6
7  40001280 <main>:
8  40001284:  save %sp, -104, %sp
9  40001288:  cmp  %g3, 4
10 4000128c:  bg   400012a0
11 40001290:  nop
12 40001294:  call 40001270 <incr>
13 40001298:  nop
14 4000129c:  b    40001284
15 400012a0:  nop
16 400012a4:  ret
17 400012a8:  restore
18
```

a) Control Flow Graph (CFG) (6 Points)

b) Control Flow Instruction Graph (CFIG) (4 Points)

Problem 3 (Non-Invasive Physical Attacks) (22 Points)

1. What are side channels in the context of non-invasive logical attacks? (2 Points)

2. What are the goals of a side channel attack? (2 Points)

3. Explain why the following implementation of a string compare function is vulnerable to side-channel attacks. (3 Points)

```
1  int strcmp (const char *str1, const char *str2) {
2          const unsigned char *s1 = (const unsigned char *) str1;
3          const unsigned char *s2 = (const unsigned char *) str2;
4          unsigned char c1, c2;
5
6          do {
7                  c1 = (unsigned char) *s1++;
8                  c2 = (unsigned char) *s2++;
9                  if (c1 == '\0')
10                         return c1 - c2;
11         } while (c1 == c2);
12
13         return c1 - c2;
14 }
```


4. Explain both the approaches of Simple Power Analysis (SPA) and Differential Power Analysis (DPA) and outline their differences. (5 Points)

5. Exploit the vulnerability of an AES algorithm that employs an unsafe implementation of the *xtime* function and find out the secret key. You know the first byte of the key to be either 0xAB, 0x29, or 0xC2, and have the following (empirically found) time measurements for random input ciphers with four different first bytes:

input	time measurements
0x7A	1158 μs
0xC2	1161 μs
0x49	1175 μs
0xA5	1108 μs

Furthermore, you know the employed S-Box of the algorithm:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Also, given is an XOR Table, with both operands being show on the axis. E.g. A XOR 2 = 8

XOR	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
4	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
5	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

a) Explain how the given information can be used to find the first byte of the secret AES key. (5 Points)

b) Perform the previously explained steps to find the first byte of the secret AES key. (5 Points)

Additional Page 1

Additional Page 2