



**PADERBORN UNIVERSITY**

*The University for the Information Society*

Faculty for Computer Science, Electrical Engineering and  
Mathematics

Department of Computer Science

Research Group System Security

## Master's Thesis

Submitted to the System Security Research Group  
in Partial Fulfillment of the Requirements for the Degree of

## Master of Science

**Insert title here**

Insert your name here

Supervisors: Prof. Dr.-Ing. Juraj Somorovsky  
(insert further supervisors here)  
(and here)

Paderborn, December 8, 2020



## **Abstract**

How can I improve my scientific writing skills?

- <https://www.nds.ruhr-uni-bochum.de/teaching/theses/writing/>
- <https://www.uni-paderborn.de/universitaet/kompetenzzentrum-schreiben/>



## Official Declaration

I hereby declare that I prepared this thesis entirely on my own and have not used outside sources without declaration in the text. Any concepts or quotations applicable to these sources are clearly attributed to them. This thesis has not been submitted in the same or substantially similar version, not even in part, to any other authority for grading and has not been published elsewhere.

## Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen worden ist. Alle Ausführungen, die wörtlich oder sinngemäß übernommen worden sind, sind als solche gekennzeichnet.

---

DATE

---

INSERT YOUR NAME HERE



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Structure details . . . . .	1
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Duties and Agreements . . . . .	3
2.1.1	Rules for Students . . . . .	3
2.1.2	Rules for Supervisors . . . . .	4
2.2	Hints on Typesetting . . . . .	4
2.2.1	Citations . . . . .	5
2.2.2	Structuring Text, English Hints . . . . .	5
2.2.2.1	Text . . . . .	5
2.2.2.2	English Hints . . . . .	6
2.2.2.3	General Hints . . . . .	7
2.2.3	Formulas, Figures, Tables, Definitions . . . . .	8
2.2.3.1	Formulas . . . . .	8
2.2.3.2	Figures . . . . .	9
2.2.3.3	Tables . . . . .	10
2.2.3.4	Definitions . . . . .	10
2.2.4	Listings, Algorithms . . . . .	10
2.2.4.1	Listings . . . . .	10
2.2.4.2	Algorithms . . . . .	11
2.2.5	Protocols . . . . .	13
2.2.5.1	2-Party Protocol Sessions . . . . .	13
2.2.5.2	Protocol Headers . . . . .	13
<b>3</b>	<b>Design</b>	<b>15</b>
<b>4</b>	<b>Implementation</b>	<b>17</b>
<b>5</b>	<b>Evaluation</b>	<b>19</b>
<b>6</b>	<b>Conclusions and Future Work</b>	<b>21</b>
	<b>Bibliography</b>	<b>23</b>
	<b>List of Figures</b>	<b>25</b>

<b>List of Tables</b>	<b>26</b>
<b>List of Algorithms</b>	<b>27</b>
<b>List of Listings</b>	<b>28</b>
<b>A Java Code</b>	<b>29</b>



# 1 Introduction

This chapter should contain the following parts:

- Introduction to the topic. Motivation
- Information about the state-of-the-art research and techniques.
- Your contribution and results.
- Organization of the thesis

Basically, this chapter contains the first three chapters from the thesis exposé. It depends on you, whether you want to use subsections or small paragraphs to divide the above parts.

If you want to have **examples** of how good introductions look like, you can take a look at papers published at top-tier conferences, for example, at USENIX Security.<sup>1</sup>

## 1.1 Structure details

**Motivation.** Introduce and motivate the topic:

- Write some words in general about the topic you are going to tackle in your thesis. Motivate why is this topic interesting in general (e.g., where is it used, who implements it, what problem does it solve)?
- Explain some more background for the topic, help the reader (who may have never heard of the topic) understand what you are talking about.
- What is your personal motivation to deal with this topic?
- Which interesting problems do you expect?

---

<sup>1</sup><https://www.usenix.org/conference/usenixsecurity19/technical-sessions>

**Current state of research.** Explain what the current state of the research is and summarize related work:

- What is a trivial (not suboptimal) way to solve  $x$ ? How does attack  $y$  work?
- How is  $x$  usually solved in practice? What are the state-of-the-art counter-measures against attack  $y$ ?
- What is the relevance of this work concerning your thesis?
- What are typical ingredients/techniques to construct the systems we are interested in?
- How do Xie et al. [5] solve the problem? How does their work improve the algorithm?
- What does the algorithm in [5] do (roughly)?
- ...

**Contributions.** You already know what are your exact results and contributions. So summarize them briefly here:

- What was the problem with existing solutions or existing attacks?
- What was the goal of the thesis in a nutshell? What problem could you solve? What algorithm/attack could you improve?
- Give an overview of your results.

**Organization of this Thesis.** This thesis is divided into  $x$  chapters. In Chapter 2, we summarize the background relevant to this thesis. ...

## 2 Background

Starting with this chapter, always start a chapter with a short but informative text about the following sections. Point out the relevance of the sections and create interconnections between them. Never ever just write a single sentence here. Furthermore, you are strongly advised to respect the hints given in this template.

In this specific chapter, describe the background relevant to the thesis, for example:

- Short cryptographic background
- Basic protocol description (e.g., TLS)
- Tools you are relying on (e.g., how TLS-Attacker works)

In the following, we give an overview of duties and responsibilities relevant for students as well as for their supervisors.

### 2.1 Duties and Agreements

To successfully write your thesis, you should definitely respect some rules. They are explained in the following sections.

#### 2.1.1 Rules for Students

At the beginning of your thesis, estimate the complexity of the work you are going to have. Take into account that you will have problems with certain aspects of your thesis that will consume a lot of time. Consider times for recreation and delays you can not influence, for instance, asking your supervisor, waiting for orders to be shipped, complex problems during the implementation phase, and so on . . .

For some students it is a good idea to agree upon a rough plan (with their supervisor) on how to make progress on their thesis and what goals to achieve. Milestones might help to control your progress. If you fail to meet a milestone in time, contact your supervisor on why this happened or when to expect it to be fulfilled.

If you have a problem, try to solve it on your own twice over. Some things just take time. In case you fail to solve it on your own, write an email to your supervisor and tell him about your problem and what you did to solve it. Make an appointment if necessary. Please do not jump right into his office, supervisors have other stuff to do, too.

For quotations, either use “quotation” or “quotation”. For some words, you should use a tilde to link them, for example, when referring to chapter 6 you should use it. Or use Chapter 6. This prevents words from being separated by a line break or some other rare circumstances. Use BibTeX within your thesis and learn the different citation options [5, 3].

One last piece of advice. Do not try to attend courses in parallel to your thesis. You should take this seriously and not think that writing a thesis is done quickly.

### 2.1.2 Rules for Supervisors

“With great power comes great responsibility” :-)

- It is very important that if you want specific things to be done that you send these important instructions by mail. Your student might be in a moment of confusion when telling him.
- Offer your students the opportunity to talk to you. While discussing things, tell your student to write down the results of this discussion and tell him to send you this summary by mail to ensure (if necessary), you did not talk at cross purposes.
- Last but not least: Please be gentle to your students :-)

## 2.2 Hints on Typesetting

To get this template running, you need at least

- either TeXLive (use update utility and install the most recent packages!)
- or MikTeX (**IMPORTANT**: install the cm-super font package manually!)

This template is confirmed to work in both situations. In case it does not work for you, there is something wrong with your L<sup>A</sup>T<sub>E</sub>X environment :-)

You can use `pdflatex` or `latex` to typeset this template.

### 2.2.1 Citations

Please use citations (referencing previous work as well as relevant standards and specifications):

- Some examples are given in `literature.bib` and provide you ways to cite scientific articles [5] or online texts [4]. You can extend the file with your references.
- For well-established conferences, we recommend the usage of `cryptobib`: <https://cryptobib.di.ens.fr/>. Download the file or include it in your repository as a submodule as written in the the README. You can then cite many papers, for example [1].
- For RFCs, there is also a useful bib file: <https://git.scc.kit.edu/TM/rfcbib>. This allows you to cite RFCs, for example with: `\cite{rfc5246}`. The result is [2].
- You can generate W3C report citations using: <https://w2.syronex.com/jmr/w3c-biblio>

### 2.2.2 Structuring Text, English Hints

Another text about the following sections ...

#### 2.2.2.1 Text

Always try to structure your text in a manner that makes sense. Either use indentations, itemize or enumeration environments.

This sentence will have an indentation at the beginning. Now an enumeration starts:

1. One.
2. Two.
3. Three.

Sometimes you do not want an indentation. Use the `noindent` command in such a case.

**One** Is the first number.

**Two** Is the second number.

**Glossary** Use the glossary package for acronyms. In addition, the glossay package can help you to avoid typing the same word in different ways. For example students tend to mix-up the writing of the word *User-Agent* in different ways: user-agent, User-Agent, user agent, User agent. This inconsistency can be avoided by just using the glossary entry: User-Agent (UA).

#### 2.2.2.2 English Hints

- Use an active voice and avoid using passive wherever possible.
- *Always* use the present tense (especially when you refer to content that occurs later in your text). For example:
  - *wrong*: The next chapter *will* explain ...
  - *correct*: The next chapter explains ...
- Either use American English or British English, but do not mix (e. g. summarize vs. summarise, analyze vs. analyse, ...). American English is preferred.
- Do not use filler words.
  - omit: “some kind of” and others ...
- Never use a comma before “that”.
- For enumerations, always use a comma before “and”: “... module 1, module 2, and module 3.”.
- The title of your thesis is capitalized except for words like and, or, with, the, a ...
- *Always* address the reader using the third person: “As one can see from ...” and not “As you can see ...”.
- All tables, figures have to be explained very briefly in the text itself.
- Always use correct quantifications:
  - *wrong*: ... a small amount of runs ...
  - *correct*: ... at most three runs ...
- Never use “I”. Depersonalize your sentences or use “we” if necessary.
- Read *The Elements of Style* by William Strunk, Jr., which is for example available at <http://www.crockford.com/wrrrld/style.html>. The (short) book provides an overview of typical errors and helps you to significantly improve your English.

- Do not abbreviate “e. g.” within a sentence, always write “for example”. However, within in parentheses you are allowed to abbreviate and use, e. g., and, i. e., as shown here: with a comma right before and after it.
- Use a comma after *However*, *Moreover* etc.

### 2.2.2.3 General Hints

- Use non-breaking small space for some abbreviation
  - z. B.
  - u. a.
  - e. g.
- Use a non-breaking space just before references, parentheses and so which shall not begin at the beginning of a new line. This sentence will not break here (and here).
- Did you notice the overfull horizontal box (hbox)? You should avoid these! Underfull boxes are not that bad. But only fix them when most of the section, paragraph etc is ready. Otherwise you have to fix them more than once. You can tell L<sup>A</sup>T<sub>E</sub>X when to break a word if it does not do it correctly. Just put a \- at the corresponding position in the word. Vertical overfull boxes (vbox) occur if the document uses \flushbottom instead of \raggedbottom. That way, L<sup>A</sup>T<sub>E</sub>X ensures that each page ends with the last sentence in the last line (except for the final line in a section). To enforce this, L<sup>A</sup>T<sub>E</sub>X sometimes has to add extra vertical space between, e. g., paragraphs. Overfull vertical boxes are hard to fix, as additional content needs to be added or even has to be removed sometimes. Keep in mind that any changes to the type area (Satzspiegel) might produce many additional over- or underfull boxes (and of course it will fix other boxes).
- Read <ftp://ftp.dante.de/tex-archive/info/german/l2tabu/l2tabu.pdf>. Really, read it.
- You can find many more good information at <http://www.dante.de/CTAN/info/lshort/german/l2kurz.pdf>
- The KomaScript guide is very useful: <ftp://ftp.dante.de/pub/tex/macros/latex/contrib/koma-script/scrguide.pdf>

### 2.2.3 Formulas, Figures, Tables, Definitions

#### 2.2.3.1 Formulas

Define abbreviations with the `\acro{...}` command, use them in the text mostly with `\ac{...}`. (Yes, in this example there are still a lot of wrong abbreviations. Make it better :)

So, testing abbreviations the Advanced Encryption Standard (AES) is written in different form. Lets see, when using the AES again, what will happen :D .

Using the method shown in Table XX for all three functions yields.

$$f_a^4 = 0x2C79 = abc + ac + ad + bc + a + b + d + 1 \quad (2.1)$$

$$f_b^4 = 0x6671 = abd + acd + bcd + ab + ac + bc + a + b + d + 1 \quad (2.2)$$

$$f_c^5 = 0x7907287B = cde + abde + ade + de + abce + bce + ce + be + bcd + acd + bd + d + bc + ab + b + 1 \quad (2.3)$$

When typesetting formulas, pay special notice on constants, variables, and units:

$$\mathcal{F}_\omega\{x(t)\} = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \quad (\text{Fourier-Transformation})$$

The use of constants, variables and units is explained by “Rohde & Schwarz” in their famous document “Der korrekte Umgang mit Größen, Einheiten und Gleichungen” [4]. These rules are in compliance with ISO-31. Consequently, always typeset the following in italics:

- Variables like  $k$ ,  $x$ , ...
- Functions like  $f(x)$ , ...
- Physical constants like  $c_0$ , ...
- Indices that are variables or physical units, like  $a_{i,j}$  or  $c_V$ .

Always typeset the following upright:

- Functions with fixed name like  $\sin(x)$  or  $\Gamma(x)$ .
- Mathematical constants like  $\pi$ ,  $i$  or  $e$ .
- Units and their prefixes, like  $\lambda = 0.56 \mu\text{m}$ , alternatively  $\lambda = 0.56 \mu\text{m}$ .
- Indices that represent names or identifiers, like  $x_{\text{max}}$  or  $\mu_{\text{B}}$ .



In case it is necessary to make heavy use of user defined functions, one should use `\DeclareMathOperator` to define the corresponding function. Finally, a good example how it should *not* look like.

$$Throughput = 30mbit/s$$

In case you need some extra symbols: <http://mirror.ctan.org/info/symbols/comprehensive/symbols-a4.pdf>

### 2.2.3.2 Figures

Figures and tables are important to explain things. Here are some rules that apply, when using figures:

- Whenever possible use vector graphics (eps, pdf, svg, ...) instead of bitmap graphics (jpg, gif, ...).
- All figures should have the same font and size (do not scale them or the size will change) and “style” (line strength, arrow heads, ...).
- Some employees of the chair need all figures in `.eps`. However, do *not* convert your `.jpg` and `.png` to `.eps`, instead use a *wrapper* program to wrap these file types into the `.eps` format. As a consequence, you are forced to use `latex` to typeset your document instead of `pdflatex`. Appropriate wrapper programs can be found here:
  - Windows: click
  - Linux/Mac: click
- **Always** try to use your own figures, so you do not run into copyright problems and it is easier for us, to reuse these figures for papers. You might want to have a look at these tools to create your own figures:
  - Windows: MS Visio (available via MSDNAA), Graphviz, Gnuplot ...
  - Linux: xfig/jfig, IPE, Graphviz, Gnuplot ...
  - Mac: IPE, Graphviz, Gnuplot, OmniGraffle (commercial, academic licensing available) ...

There are many possibilities on how to include figures, here is just one example on how to do it. In case you need further assistance, please google for `l2picfaq`.

### 2.2.3.3 Tables

There are many possibilities on how to create and include tables. From a typographic point of view, *one should avoid any vertical lines*, cf. Table 2.1.

Table 2.1: Captions for tables are *always above* the table and give a short but informative description of the table. Always use full sentences here and end them with a full stop.

Amount <sup>a</sup>	Price	Component	
		Description	Role
23	1.234 \$	good stuff	important
multirow example the other row	x	y	XXX
42	43.123,13 <sup>b</sup>	good stuff	important

<sup>a</sup>This is a footnote inside a table, you need a minipage for this to work.

<sup>b</sup>This is another footnote inside a table.

### 2.2.3.4 Definitions

This is a definition. You can of course make a reference to it 2.2.

**Definition 2.2 (A name)** *A really good definition. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.*

## 2.2.4 Listings, Algorithms

### 2.2.4.1 Listings

For source code listings, three options are available:

- the `verbatim` environment,
- the `listings` package,
- and the `lgrind` package.

The `verbatim` environment is the most simple environment and not suited for large code listings (due to different limitations). Only use it for single

`$ important shell commands`

Otherwise, either use the `listings` or `lgrind` packages. The `listings` package is easier to use, therefore we present it here. *Important* advice: Only explain important functions and/or structures of your program in your thesis..Especially point out the big picture of your program, for instance, how different modules interact and which important input limitations to respect. Please note: Using special language characters (ê, ü, ä, ...) in your source code is strongly discouraged, as they may cause problems using the `listings` package.

```

1  /*!
2  * This is a Doxygen comment for a function.
3  * \param first operand
4  * \param second operand
5  * \returns a+b
6  */
7  int sum(int a, int b)
8  {
9  return (a + b);
10 }
```

Listing 2.3: A sample listing of a C function. Description of the function is here. Please note that different languages are available.

```

1  entity InterLeavedMul is
2  generic(wide : natural :=8); -- highest bit
3  port(clk : in std_logic;
4  rst : in std_logic;
5  x : in std_logic_vector(wide-1 downto 0);
6  y : in std_logic_vector(wide-1 downto 0);
7  N : in std_logic_vector(wide-1 downto 0);
8  start: in std_logic;
9  done : out std_logic;
10 xyN : out std_logic_vector(wide-1 downto 0));
11 end InterLeavedMul;
```

Listing 2.4: A sample listing of a VHDL entity. Description of the entity is here. Please note that different languages are available.

You should thoroughly document your code using comments and (best case) by using a documentation system like Doxygen. Please ask your supervisor for additional rules (e.g. which repository system to use, etc.). Regularly commit your changes and backup your data!

### 2.2.4.2 Algorithms

For many theses, typesetting algorithms is necessary. There are at least four packages available that allow easy typesetting of algorithms.

- `program` offering the environment `program`.
- `algorithm` offering the environment `algorithm`.
- `algorithmic` offering the environment `algorithmic`.
  - This package sometimes has compatibility problems with `hyperref`.
- `algorithm2e` either offering the environment `algorithm` or `algorithm2e`.

Students are advised to use only *one* of these packages and not mix them. The author of this template suggests to use the package `algorithm2e` with the option `algo2e`. This prevents conflicts with other packages, just in case it is ever required to mix `algorithm` or `algorithmic` with `algorithm2e`.

---

**Algorithm 2.5:** INSERTION-SORT
 

---

**Data:** unsorted array  $A[1 \dots n]$

**Result:** array  $A[1 \dots n]$  with  $A[1] \leq A[2] \leq \dots \leq A[n]$

```

1 begin
2   for  $j \leftarrow 2$  to  $\text{length}[A]$  do
3      $key \leftarrow A[j]$ ;
4     /* Insert  $A[j]$  into the sorted sequence  $A[1 \dots j-1]$  */
5      $i \leftarrow j - 1$ ;
6     while  $i > 0$  and  $A[i] > key$  do
7       ;
8        $A[i+1] \leftarrow A[i]$ ;
9        $i \leftarrow i - 1$ 
10     $A[i+1] \leftarrow key$ 

```

---

2.2.5 Protocols

2.2.5.1 2-Party Protocol Sessions

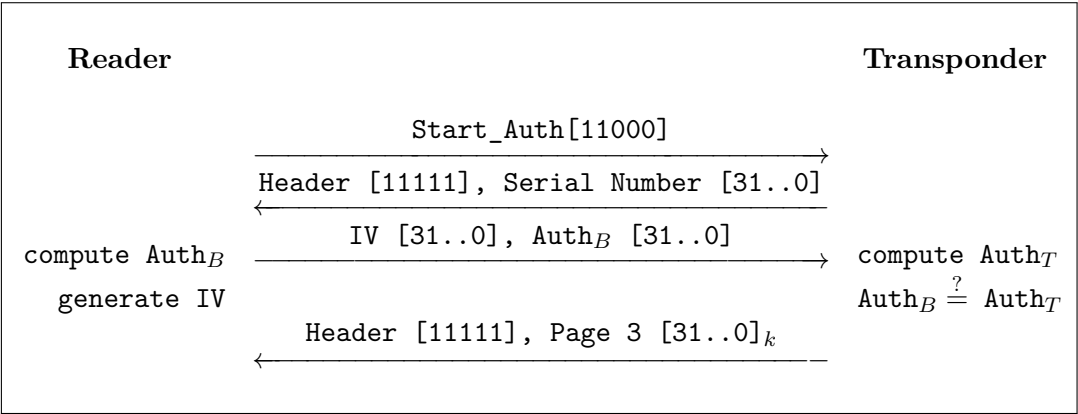


Figure 2.6: Mutual authentication of the HITAG 2 protocol in crypto mode.

2.2.5.2 Protocol Headers

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ID															
QR	Opcode			AA	TC	RD	RA	Z				RCODE			
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

Figure 2.7: DNS Request



## 3 Design

Describe on a high level (on about 5? pages), how your design works, without revealing many implementation details. Implementation details will be provided as a part of the next section.





## 4 Implementation

Now provide more details about your work and about your implementation. This can include state diagrams, class diagrams, workflows, etc.



# 5 Evaluation

This chapter contains:

- Test environment description (machines used to perform evaluation, details about tested libraries)
- If you have performed real scanning, give a number of hosts you have scanned and how long did it take.
- Provide some nice graphs and tables summarizing your evaluation.



## 6 Conclusions and Future Work

This is **not only a summary** of your thesis!

You can briefly summarize the results. But then conclude the thesis and write about what we have learned:

- Are there any interesting evaluation results previously not known?
- Do these results present novel insights? Can we learn anything from this thesis?

Of course, you could not have tackled everything. During your thesis, you have definitely found many interesting research questions that are still open and can be solved in the future theses.

- What can be implemented in the future theses?
- Is there anything specific to look at in the future?
- Can you also estimate how much work it would be?



# Bibliography

- [1] M.R. Albrecht, J. Massimo, K.G. Paterson, and J. Somorovsky: *Prime and prejudice: Primality testing under adversarial conditions*. pp. 281–298, 2018.
- [2] T. Dierks and E. Rescorla: *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246 (Proposed Standard), Aug. 2008. ISSN 2070-1721. <https://www.rfc-editor.org/rfc/rfc5246.txt>, Obsoleted by RFC 8446, updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685, 7905, 7919, 8447.
- [3] J. Newsome and D. Song: *Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software*. In *Symposium on Network and Distributed System Security (NDSS)*, 2005.
- [4] Rohde & Schwarz: *Der korrekte Umgang mit Größen, Einheiten und Gleichungen*. [http://www.rohde-schwarz.de/ps/rus/tools/show\\_8437\\_document/Der\\_korrekte\\_Umgang.pdf](http://www.rohde-schwarz.de/ps/rus/tools/show_8437_document/Der_korrekte_Umgang.pdf), visited on December 8, 2020.
- [5] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov: *Spamming Botnets: Signatures and Characteristics*. ACM SIGCOMM Computer Communication Review, 38(4), 2008.





# List of Figures

2.6	Mutual authentication of the HITAG 2 protocol in crypto mode. . .	13
2.7	DNS Request . . . . .	13

# List of Tables

2.1	This is the short caption for the <i>List of Tables</i> . . . . .	10
-----	---	----

# List of Algorithms

2.5 INSERTION-SORT . . . . . 12

## List of Listings

2.3	A sample listing of a C function. Description of the function is here. Please note that different languages are available. . . . .	11
2.4	A sample listing of a VHDL entity. Description of the entity is here. Please note that different languages are available. . . . .	11

## **A Java Code**