CyBOK Wiki: feasibility study Documentation

Lőrinc Thurnay | University for Continuing Education Krems

1. Executive summary

CyBOK is distributed as PDF, a medium inherently tailored for print rather than browser-based accessibility. In this study, we assessed the technical feasibility of creating an automated software framework that publishes CyBOK content to a browser-based Wiki system - **CyBOK Wiki** - thus improving CyBOK's accessibility, discoverability, and user experience.

Publishing CyBOK on a web portal is challenging since CyBOK is written in LaTeX, a complex typesetting language designed for print media, but one without a segmentation and conversion standard for the web. CyBOK itself is also a vast, interlinked resource with complex LaTeX customizations, which adds to the technical complexity. On the other hand, CyBOK's content by its nature fits well with a structured, hyperlinked online resource like Wiki.

This study aims to explore and describe the opportunities, limitations, technical challenges and questions to be answered that relate to the implementation of CyBOK Wiki. It does so to inform CyBOK's stakeholders in the decision whether and how to realize CyBOK Wiki, and lay the foundations to the eventual CyBOK Wiki project's planning, design, and technical implementation. These efforts were driven by the manual exploration of CyBOK's source code and the development of a proof-of-concept software framework. The software pre-processes, segments, converts and publishes CyBOK's LaTeX source to a Wiki system with promising results - proving the feasibility of key concepts of CyBOK Wiki.

Lessons learned suggest that the more complex problems beyond the scope of the proof-of-concept are also surmountable. We identify and describe several complementary tools and methods to implementing conversion mechanisms that are not supported by existing tools. We describe how to handle rarely occurring but complex conversion tasks manually, in a sustainable and reliable way.

We conclude that in some cases, changing some structural elements of the CyBOK LaTeX source may be the best (or only) solution for solving conversion issues. This will create additional work to the CyBOK code base maintainer and is something that future CyBOK authors and content editors should also ideally adopt. There are no changes foreseen that would impact the CyBOK PDF release in any way.

We identify open questions that must be debated and answered to implement CyBOK Wiki beyond the proof-of-concept. These questions are mostly about the change of medium from PDF to a web portal: how we treat CyBOK as a structured document in a web portal, what new functions to give it and how to present it on a new user interface. We finally outline the next steps to be taken towards the realization of CyBOK Wiki. The *Appendix* includes annotated screenshot illustrations of the proof-of-concept software framework.

2. Table of contents

3. Glossary

- **CyBOK PDF**: the current, official and canonical PDF-based release of CyBOK
- **CyBOK Wiki**: the prospective, MediaWiki-driven web portal release of CyBOK of which the study at hand investigates the feasibility.
- **IAGs**: indexes, acronyms and glossary items (collectively)
- **KA**: Knowledge Area
- **MW**: MediaWiki - the open source collaboration and documentation platform that also drives WikiPedia
- **SEO**: Search Engine Optimization
- **UI**: User interface
- **UX**: User experience

4. Introduction

Over 1,000 pages, CyBOK - The Cyber Security Body Of Knowledge - is distributed as PDF, a medium inherently tailored for print rather than browser-based accessibility. Even though it is only available online, CyBOK is published in a format that is not optimal for the computer-based interactive exploration and reading experience, and even less so for smaller handheld devices. Therefore, the purpose of this project is to determine whether it is feasible to implement CyBOK Wiki, an automated software framework that publishes CyBOK content to a browser-based Wiki system to improve its accessibility, discoverability, and user experience.

This is a feasibility study with a technical focus. It aims to explore and describe opportunities, limitations, technical challenges and questions to be answered that relate to the implementation of CyBOK Wiki. The study is successful if its findings help inform CyBOK's stakeholders in the decision whether and how to realize CyBOK Wiki, and lay the foundations to the eventual CyBOK Wiki project's planning, design, and technical implementation.

While this study's aim is not the implementation of a software itself, it does produce a proof-of-concept software framework. The development of this framework drives and informs the above defined focus points. The proof-of-concept attempts the pre-processing, segmentation, conversion and publishing of CyBOK's LaTeX source to a Wiki system.

In the first chapter of this document, we introduce the context – *4.1 Motivation*, *4.2 Challenges*, and *4.3 Methodology* - of this study. We then describe *5 The proof-of-concept framework* software, the development of which drove this study: we give an overview of its functionality and discuss its results. We continue with technical *6 Considerations for implementation*: important lessons learned and ideas to pursue for the implementation of CyBOK Wiki beyond the proof-of-concept. We provide a detailed discussion of *7 Open questions* that must be debated and answered to move forward with realizing CyBOK Wiki. We finish off with *8 Conclusions* and a broad overview of *9 Next steps* to be explored beyond this study. The *Appendix* includes annotated screenshot illustrations of the proof-of-concept software framework.

4.1 Motivation

Wikis are popular browser-based encyclopedia-like knowledge management systems that organize information in pages (or Articles) and offer rich functionality for textual data. Wikis are usually open for the public (or at least a community) to freely edit (this is a function that would not be utilized in CyBOK Wiki).

The idea of CyBOK Wiki is to make CyBOK's Knowledge Areas (KA) and their (sub)sections available as smaller, self-contained, but interlinked Wiki pages, enabling the many benefits of using a cutting-edge web portal instead of a PDF file for exploring and studying CyBOK. These foreseen benefits include:

- **User experience:** Users could get automatic recommendations for relevant CyBOK (sub)sections to read and search CyBOK with a Wiki search engine. They can easily navigate across (sub)sections and have many sections open in browser tabs simultaneously. Copying and pasting from HTML retains formatting, while doing so from PDF often introduces artifacts.
- **Accessibility:** Modern web portals are responsive to screen size: they offer optimal reading and navigation experience on large computer screens as well as small smartphones, whereas reading the CyBOK PDF on a phone is an arduous experience. HTML resources are customizable for users’ accessibility needs, e.g.: by letting them change typeface, size, colours, or contrast. HTML is well suited for screen readers.
- **Discoverability:** While PDFs are searchable, search engines penalize them in their result rankings. Modern web portals are optimized for search engines, making it easier to discover CyBOK, increasing its exposure. As standalone pages, users can link to specific (sub)sections from other websites, blog posts, emails, or bibliographies.

4.2 Challenges

CyBOK is written in LaTeX, a complex, powerful document typesetting system that has been in development for forty years. It allows for heavy customization, templating, and macros - indeed, it is a Turing complete language. LaTeX is created with print media (rather than web publishing) in mind, and it is marked both in its capabilities and design.

While rendering to PDF (an industry standard print-focused document format) is the primary use case of LaTeX nowadays, due to the above factors, segmenting and converting to web-based content is a complex task that is not standardized. Tools such as Pandoc convert from LaTeX to web-based formats (such as markup languages used in Wiki software), but they are not perfect; it cannot be, not without a standard definition of how to convert to web formats. CyBOK heavily relies on complex LaTeX functionality and customization. Therefore, one challenge of CyBOK Wiki is the conversion of LaTeX to a Wiki markup format.

The other challenge has to do with CyBOK: it is itself a long and interlinked document that is edited with print (but at least PDF) as the target medium in focus. As a user interface, a PDF document (and a book especially) is very different from an online portal, and as such, requires different considerations: different navigation, UI elements, formatting concepts and metadata. The automated conversion of this vast resource to fit the requirements and enable the benefits of another UI, is a challenge.

Importantly, the content of CyBOK is fitting for a segmented, hyperlinked online resource: it is heavily segmented into Knowledge Areas and (sub)chapters and uses (cross)references to create linkages with other parts of the document. Larger sections are meant to be comprehensible as standalone resources as well, and smaller (sub)sections are contextualized with surrounding and cross-referenced (sub)sections as well as citations.

4.3 Methodology

This feasibility study was driven by the manual exploration of CyBOK LaTeX source and PDF resources, as well as the development of a proof-of-concept software framework that attempts to pre-process, segment, convert and publish CyBOK's LaTeX source to a Wiki system. The aim of both the exploration and the software development was to identify opportunities, limitations, technical challenges and questions to be answered that are relevant to the eventual implementation of CyBOK Wiki.

The scope of exploration and development were determined by the above goals. We pursued directions that promised relevant lessons to be learned and we tested our hypotheses against the rest of the code base. We did not prioritize implementing robust and reusable code that adheres to industry standards required by a production-quality software project, as the additional effort to reach high code quality in these regards often did not promise significant learnings about CyBOK (having said that, we took care to produce readable code with informative comments, to enable reusability beyond the proof-of-concept). We did not pursue the implementation of complex development tasks that would have taken a too significant portion of the limited resources available to the project - in such cases, we progressed only as far as to be able to assert feasibility, describe the nature of the problem and outline a strategy to tackling it. We did not employ an established systematic methodology for identifying directions for exploration and development; our approach was informed by extensive experience in the exploration and conversion of similarly complex semi-standardized data.

We noted opportunities, limitations, technical challenges and questions to be answered as soon as we identified them, and elaborated on them in this document later.

This study had access to the LaTeX source code and respective PDF releases of three of the 22 CyBOK Knowledge Areas: *Adversarial Behaviours*, *Cryptography*, and *Law and Regulation* (and is informed by the superficial familiarity with the full CyBOK PDF release). Therefore the results, conclusions and recommendations of this study are to be considered with the caveat that they are based only on a small segment of CyBOK.

5. The proof-of-concept framework

In the following, we describe the proof-of-concept CyBOK Wiki software framework, the development of which drove this study: we give an overview of its functionality and discuss its results.

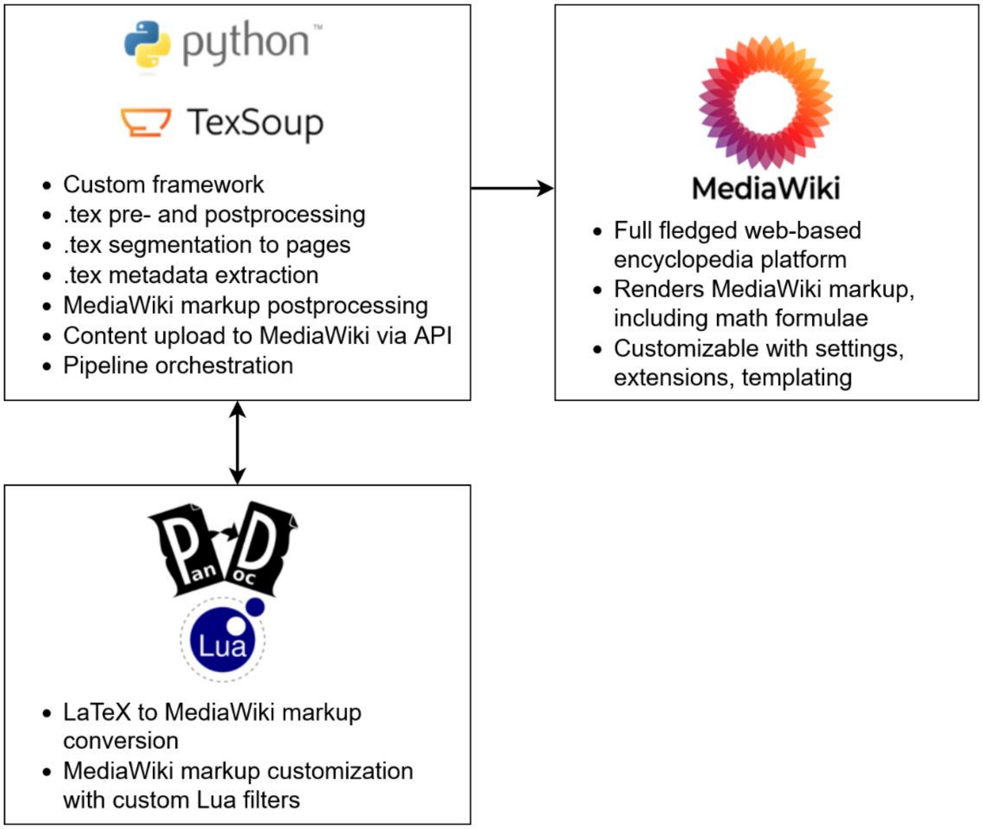

python
TexSoup
• Custom framework
• .tex pre- and postprocessing
• .tex segmentation to pages
• .tex metadata extraction
• MediaWiki markup postprocessing
• Content upload to MediaWiki via API
• Pipeline orchestration
MediaWiki
• Full fledged web-based encyclopedia platform
• Renders MediaWiki markup, including math formulae
• Customizable with settings, extensions, templating
Pandoc
Lua
• LaTeX to MediaWiki markup conversion
• MediaWiki markup customization with custom Lua filters

Figure: the broad architecture of the CyBOK Wiki proof-of-concept framework

The CyBOK Wiki framework consists of a conversion framework and a web platform. The conversion framework consists of two components.

The main component of the conversion framework is a **custom Python framework**, heavily using TexSoup, a "fault-tolerant, Python3 package for searching, navigating, and modifying LaTeX documents." The Python framework is custom written for this project. It handles the pre- and postprocessing of LaTeX files for segmentation and ingestion to the Pandoc converter. It segments large LaTeX documents into smaller self-contained LaTeX documents of CyBOK (sub)sections, to be turned into MediaWiki (MW) pages. It orchestrates Pandoc conversion to MW markup and performs postprocessing on MW markup before uploading to MW via its API.

The other is **Pandoc**, a universal document converter supporting both LaTeX and MediaWiki markup, written in Haskell and extensible by plugins written in Lua. LaTeX is one of Pandoc's main focus points and it supports advanced features of the language including tables, references and custom command definitions (to some extent). Pandoc uses an intermediary, native representation of documents (*abstract syntax tree* or AST) to convert between arbitrary document formats (so in essence, conversion happens twice: LaTeX → AST → MW markup).

The CyBOK Wiki platform is driven by **MediaWiki** (MW). It renders the uploaded pages into the familiar Wiki view, along with tables, math formulae, references and hyperlinks, and also provides complex, full-fledged online encyclopedia and knowledge management features such as search, categories, dynamic content, etc. out-of-the-box and extensible via extensions. MW is used by thousands of websites, including Wikipedia, the seventh most visited website on the Internet. Amongst MW's foci is accessibility. MW is a robust, mature software that has been in active development for over 22 years. As such, it is a reliable framework to build a sustainable project on.

All of the above software and packages are released with permissive open source licenses (Python: PSFL, TexSoup: BSD-2-Clause, Pandoc: GPL, MediaWiki: GPLv2+)

5.1 Proof of concept

The CyBOK Wiki framework created in this feasibility study is a proof-of-concept. Its main purpose is to explore the technical possibilities and limitations of converting a massive, heavily customized and interlinked LaTeX codebase (CyBOK) into a well-segmented Wiki-like platform. It serves to explore suitable tools, architecture, techniques and identify pain points that can be considered when implementing a real, production-ready CyBOK Wiki.

With this, and considering the project's small size, the focus during the software prototyping was breadth rather than depth, exploration rather than industry best practices, efficiency rather than stability. The software is not fully packaged and ready for deployment and has some functions unfinished. Having said all that, care has been taken to achieve code readability, including comments, especially for parts that are not integrated in the main function pipeline and therefore their logic is not informed by context.

Its main function is the processing and segmentation of LaTeX files, conversion to MW markup and upload to MediaWiki. It performs these functions on the three KAs that are in scope of this exploratory project without failure. In addition, it comes with some exploratory tools, some features that are not implemented in the main pipeline function because they are not stable, and some abandoned approaches, fragments of which may still prove useful in the future.

Below is an overview of the codebase with the aim of summarizing functionalities and serve as a guide for exploring and understanding the source code and architecture (for this, it is advisable to follow the source code along with reading).

5.2 Development environment

The framework has been developed and tested on the following system:

- OS: Ubuntu 20.04.6 LTS
- Python 3.8.10
- Pandoc 3.1.12.3 (Lua 5.4)
- MediaWiki 1.41.0 (Dockerized, with MariaDB 11.2.2, also Dockerized)
- Azure VM: Standard D2s v3

- CyBOK: v1.1 (*Adversarial Behavior* KA git commit: bab08478, *Cryptography* KA git commit: 137df867 *Law and Regulation* KA git commit: 71c2527b)

5.3 Installation

- Install additional required Python libraries with `pip3 install -r requirements.txt`
- Get suitable Pandoc executable from GitHub and install via `sudo dpkg -i $DEB` (Ubuntu).
- `config/mediawiki-docker-compose.dist.yaml` can serve as a template to set up a dockerized MW environment.
- Take add CyBOK Wiki-specific settings to MW's `LocalSettings.php` from the end of `config/mediawiki-LocalSettings.php`
- Install MW dependencies for PDF rendering (see also chapter *6.8 PDFs as illustrations*)
- Define the MW API endpoint in `config/config.yaml` (API works anonymously)
- LaTeX projects are to be placed in the `data` folder, e.g.: `data/ab`, `data/c`, etc.

5.4 Main functionality

The main functionality of the proof-of-concept is the LaTeX-to-MW markup segmentation, conversion and posting to MW. This process is started by running `run.py`. `run.py` triggers the main logic, `Segmenter` (in `segmenter.py`) which orchestrates much of the main functionality.

First, the LaTeX files are loaded as strings and LaTeX-agnostic preprocessing is applied by `processors.preprocessor.raw_tex_preprocessor()` on the whole LaTeX documents. (removing non-breaking spaces before citations, removing LaTeX comments, changing `\endnote{}` to `\footnote{}`). Then LaTeX is parsed by TexSoup, resulting in a detailed, searchable, modifiable, tree structure of LaTeX expressions and text tokens that can be later converted back to LaTeX files.

5.4.1 Segmentation

Here `tex_to_pages()` performs the actual segmentation of a "flat" LaTeX file to a list of pages, divided by (sub)section headings. Pages are `Page` object, that (apart from retaining the dictionary of TexSoup objects that represent the LaTeX content) also maintain the properties

- `cybok_page_id`: the (sub)section's ID from the related `\label{}`,
- `cybok_level`: the heading level of the page in the LaTeX document (is it a section, subsection, subsubsection?)
- `cybok_title`: the title of the (sub)section and therefore the page
- `children`: subsections under the page at hand.

`make_pages_hierarchy` creates a hierarchic Page dictionary representing the KA's (sub)section structure.

See chapter *7.1 Which CyBOK PDF (sub)sections should be segmented into CyBOK Wiki pages?* for more detail about the nature of the above task.

5.4.2 Processing and posting pages

`process_pages()` iterates through all pages (recursively) and performs the following:

`add_hierarchic_links()` adds navigation links to parent and child sections (as LaTeX `\ref{}` expressions to be converted to MW markup by Pandoc).

`embed_into_maintex()` embeds the isolated LaTeX markup of the (sub)Section into the `main.tex` (or equivalent) of the KA's LaTeX project. `main.tex` is intended to be the main wrapper of the whole KA's LaTeX project, it imports all LaTeX styles, classes, bibliography items, and other macros, as well as including `content.tex` (or equivalent, holding the actual LaTeX content of the KA). In this function we replace the inclusion of `content.tex` with including the section's contents. This function gives the same necessary context to the section, so when Pandoc converts the page, it can correctly parse CyBOK-specific LaTeX expressions, resolve bibliography items, etc. The result of this function is a new, "flat" LaTeX source of the Page.

`convert_to_mediawiki()` is a wrapper around Pandoc, it orchestrates the Pandoc conversion and captures its results. With `pandoc_debug` set to `True` (must be changed in source), custom Lua filters can be debugged: Pandoc outputs (i.e.: the MW markup) are redirected to a file and print commands specified in Lua get captured and printed by Python.

`mediawiki.connector.post_page()` interfaces with the MW API and posts the page. If there is a `\label{}` associated with the (sub)section being posted, a redirect page is also created where the label is the URL, redirecting to the proper content page (where the title URL). This guarantees that in-text cross-references to this (sub)sections are turned into working hyperlinks during conversion. For more details on this problem and how it requires to change the LaTeX source standard, see chapter *7.2 Extracting (sub)chapter titles and identifiers*. To demonstrate the proof-of-concept's page linking operation the changes proposed in the *below* mentioned chapters have been implemented for the *Adversarial Behaviors* and *Cryptography* KAs, see `data/ab/content.tex` and data `data/c/content.tex`, respectively. The LaTeX source of the *Law and Regulation* KA has not been altered, consequently, cross-(sub)section links are broken for this KA.

5.4.3 Pandoc

The Pandoc command issued by `convert_to_mediawiki()` specifies `pandoc/mediawiki.lua`, a Lua script of custom filters. In the proof-of-concept, these highlight acronyms with red (for proof-of-concept of a Lua filter) and fix `\ref{}` rendering (by default Pandoc generates intra-page anchor links, this is overriden so proper inter-page hyperlinks are generated).

The command also enables citation rendering, resulting in in-line bracketed [1] links that link to the bibliography element, added by Pandoc at the end of the document, with `pandoc/ieee.csl` is specified for proper bibliographic formatting. `.bib` files containing references are parsed by Pandoc automatically.

5.5 Results

The whole preprocessing, conversion and upload process described above runs for about 78 seconds for the three pilot KAs.

We assess the capabilities of the proof-of-concept implementation by visual comparison of the CyBOK PDF and CyBOK Wiki, as well as side-by-side comparison of LaTeX and WM markup sources. To summarize: with proper preprocessing, settings and custom filters, Pandoc converts LaTeX to MW markup with an impressive quality. See the *Appendix* for illustrations.

- The segmentation process isolates 118 (sub)sections across three KAs and uploads them as MW pages.
- Basic text formatting is converted, e.g.: **bold**, *italics*, bulletpoints, paragraph breaks, headings, etc.
- In-line citations [1] are converted to hyperlinks pointing at the bibliography at the end of each page.
- Bibliographies are generated automatically at the end of each page, with *IEEE* style. Only those bibliographic elements are included that have references on the page at hand, so bibliographies on each page are shorter than the several hundred strong bibliographies at the end of the CyBOK PDF KAs. Bibliographic numbering may differ because the CyBOK PDF numbering happens on KA level while CyBOK Wiki numbering happens on (sub)section level.
- Cross-references to other (sub)sections are turned into hyperlinks to their respective pages. Hyperlink texts are the LaTeX `\label{}` values of the linked (sub)sections – this can be changed, as described in detail in chapter *7.4 How to display section titles?*"
- Notes are correctly referenced and linked in-text and get rendered in the native MW reference style at the end of each page (only the ones relevant to that page).
- Hyperlinks to parent- and subsections are added to ease navigation across the book. See chapter *7.3 How to implement and display CyBOK's structure in MW pages?* for detailed considerations on how to better implement navigation beyond the proof-of-concept.
- Math formulae are rendered just like in LaTeX, both in-line and as blocks.

Not everything gets converted to MW markup correctly. Important and barely noticeable elements alike may be omitted or converted incorrectly by Pandoc or the rest of the conversion pipeline. We do not have a comprehensive list of these discrepancies, as compiling such a resource would require meticulous work beyond the resources available in this project. Nevertheless, in this document we tried to make note of every discrepancy we have noticed during our work - discussed in detail in subsequent chapters or described plainly in the *6.9 To-dos* list. The chapter *6 Considerations for implementation* is focused on different strategies to tackle conversion discrepancies beyond the proof-of-concept.

5.6 Other functions

At the beginning of this study we had not found Pandoc, because it is a universal tool converting between more than 50 formats, the *LaTeX* and *MediaWiki conversion* search terms initially missed this software. For this reason, we started exploring the possibility of building a custom LaTeX to MW markup converter. Without significant effort we developed a minimalistic, extensible framework that proved surprisingly robust and produced good results.

Having discovered Pandoc soon after, we compared the two approaches and eventually decided that relying on Pandoc is more efficient than continuing with the custom converter. The arguments against Pandoc would have been its complicated and multifaceted customization mechanisms and the fact that it is written in Haskell (a language the author does not have expertise to debug or fork). However, Pandoc handles complex LaTeX solutions such as tables, nested environments and custom command definitions very well out-of-the-box - developing these in a custom parser would be far too resource intensive. With this, we abandoned the development of the custom parser, but include its source code as parts of it might be useful in the development of CyBOK Wiki beyond the proof-of-concept - especially pre- and postprocessing snippets such as the properly implemented MW native referencing syntax.

The abandoned custom parser's entrypoint is `tex2wiki.py` - functionality can be tested by running this. The `parsers` dictionary as well as `utils.py` and `postprocessor.py` are used exclusively by this parser. The parser produces a MW markup file saved as `data/wikitext.txt`, without any segmentation or interfacing with the MW API.

A separate tool, `tools/list_cmds.py` lists all the distinct LaTeX command names used in a LaTeX file, counting their instances. It can be useful to go through such a list to verify if all LaTeX commands indeed get converted correctly to MW markup.

The `doc/` folder hosts the documentation at hand.

6. Considerations for implementation

In this chapter we give detailed descriptions of technical considerations for the implementation of CyBOK Wiki beyond the proof-of-concept.

6.1 Tackling LaTeX expressions that are not converted correctly by Pandoc

The CyBOK LaTeX codebase uses complex macros and contributed packages. Pandoc does an impressive job at converting LaTeX source to MW markdown, but not everything works out-of-the-box. Pandoc is a complex and powerful tool, one that has a number of different ways that can be customized: through myriad of settings, different LaTeX engines, templates, plugins and filters (both Haskell and Lua, both for input and output). With both the LaTeX source and the tool we use to tackle it being complex and highly customizable, it is often hard to debug why certain LaTeX expressions are not converted (correctly).

6.1.1 With Pandoc

We find that a good way of getting a sense of how Pandoc processes LaTeX (and thus, where the conversion pipeline might be broken and what the Pandoc-specific approach to fixing it might be) to create a custom Pandoc Lua filter to access the expression in question, and debugging it in the Lua filter, by printing the `elem` (element) variable. With this, we get an idea of what context Pandoc is aware of after having processed the LaTeX expression. If this reveals that Pandoc has the contexts necessary for the proper conversion, we can fix it or implement the Lua filter that converts, ourselves. See the Pandoc manual's Lua filters chapter for documentation and `pandoc/mediawiki.lua` for examples.

6.1.2 With TexSoup

If the element cannot be captured by Lua filter, or important context is shown to be missing while debugging Lua, the LaTeX expression is misprocessed somewhere earlier in the Pandoc process. If this is the case, rather than submerging the rabbit-hole that is Pandoc's robust, complex way of parsing LaTeX, we suggest the creation of a LaTeX preprocessor using the Python TexSoup package and standardize or simplify the problematic LaTeX pattern to an expression that Pandoc has no problem converting. Without deep familiarity with Pandoc, a set of preprocessors implemented in Python is more efficient and more maintainable.

TexSoup parses LaTeX and builds a rich, fine-grained, object-based tree representation of the document. This tree can be navigated, searched and modified much like how JavaScript (or the popular python package BeautifulSoup) does so with HTML DOM. TexSoup can be used to change LaTeX expressions in place so Pandoc could parse them in the desired way. It can add new LaTeX elements to the document to enrich the resulting functionality (e.g.: navigational elements) It can be used to segment large LaTeX document to smaller, self-contained chunks and extract (meta)data from documents that can inform the conversion pipeline's business logic or serve as MW metadata.

6.1.3 With WikiMedia

In conjunction with the above methods, one should always consider whether there are MW Extensions out there that provide the needed functionality. Conceivably, Pandoc converts the LaTeX source correctly, but MW lacks the needed extension to render it properly. We can also consider implementing a custom MW Extension if that seems to be the most straightforward solution to solving the problem. To this end, we can also consider customizing MW CSS.

For example, we may want to render acronyms (LaTeX `\acrshort` and `\acrlong`) such that they appear inline as acronyms and hovering or clicking on them would reveal a tooltip with its definition. This could be implemented e.g. as a Lua filter that produces custom MW markup that is rendered as desired by an MW extension.

6.1.4 In LaTeX

Consider creating custom "LaTeX wrappers" to replace or extend existing LaTeX wrappers (e.g.: `main.tex`), `*.sty` and `*.cls` files. Some of the macros or other LaTeX meta-elements are complex and (some of) their functions may be specific to print media. These could be removed, their functionality simplified or redefined by new LaTeX wrapper files, custom created, tailored for Pandoc MW, and web UX/UI in mind. Ultimately, changing the LaTeX source can also be considered (see dedicated chapter on this *below*).

6.2 Manual tasks in the automated pipeline

There may be LaTeX functions (perhaps for example TikZ illustrations, see the *6.6 TikZ* chapter) that are used only in a few places throughout CyBOK, which are not supported out-of-the-box by Pandoc, and the implementation of which would require significant effort. If the effort of implementing these significantly outweighs the effort of converting them manually, the latter should be considered, as the reduction of future manual efforts by automation may not pay off in reasonable time.

In case the conversion of LaTeX function is left for manual work, it is still recommended that either Python preprocessor, or a Pandoc Lua script is implemented at least to identify the occurrence of cases (if possible) in the LaTeX source code and create a report of them. This way the person responsible for manual conversion will have a reliable and up-to-date checklist of items to be manually converted.

All changes made to pages in MW are tracked by MW. It is crucial to retain the editing history of MW pages, so when a subsequent version of CyBOK Wiki is released, the prior version's manual changes can be (manually) re-applied to the new version as well. To facilitate this process, a reporting tool could be developed for MW, automatically creating a list of manually edited pages with the edits highlighted.

In any case, the number of manual tasks should be kept to a minimum, as having too many of them would turn the conversion process inefficient and ultimately unsustainable.

6.3 Changing the LaTeX source

As a principle, the LaTeX source is the single source of truth of CyBOK and the primary CyBOK medium is its existing PDF release. Here we assume that the goal of a CyBOK Wiki implementation project is to convert this source into Wiki markup without affecting the existing processes of writing, editing, and publishing the CyBOK PDF. Consequently, we must consider the LaTeX source as essentially read-only - a resource that must be interfaced with without the ability of changing it.

Having said that, there might be challenges with the conversion to MW markup where changing the structure CyBOK LaTeX source would be the most pragmatic - or indeed the only - solution. One such example is the association of (sub)section titles with their respective reference IDs (described in the chapter *7.2 Extracting (sub)chapter titles and identifiers*. In such cases, we should analyse the possibility and consequences of changing the LaTeX source, and if justified, suggest such changes to the relevant stakeholders of the CyBOK project.

Any suggestion for changing the CyBOK LaTeX source should be made with the following considerations:

- Give proper justification to suggestions. Make the suggested changes as simple and flexible as possible, and preferably only when there are no alternative solutions.
    - CyBOK is already over 1000 pages long (when rendered in PDF), any new requirement or restriction must be applied to the entirety of the existing corpus, which requires significant resources of the LaTeX source maintainer.

  - The development of CyBOK, both as an educational resource and as large and complex codebase, is a work of dozens of people. New requirement or restriction to the source code will affect their work.
- Understand CyBOK's writing and editing processes and make LaTeX source change suggestions compatible with them.
- Make sure that any suggested change to CyBOK's LaTeX source does not alter the appearance or behavior of the CyBOK PDF in any way.

6.4 KA-specific functionality

CyBOK Knowledge Areas (KAs) cover a diverse selection of disciplines, with most KAs written and edited by different experts. Because of this, KAs are not necessarily uniform in structure and format - neither as a publication, nor in its source code.

CyBOK Wiki must handle KA specific differences. To this end, it is recommended that the CyBOK Wiki migration framework is developed in an extendable manner, such that specific parts of the conversion pipeline can be extended with logic that may only be applied to converting relevant KAs.

For example: the *Law and Regulation* KA starts with a *NOTICE AND DISCLAIMER*, appearing in PDF at the beginning of the KA, but we might want to display this on every MW page that belongs to the KA. This would necessitate KA-specific preprocessing logic.

This is recommended to be realized by the python framework that wraps around Pandoc, leaving Pandoc KA-agnostic. A practical approach would be object-oriented components, where for example a common preprocessor object can be inherited from and extended by a KA-specific object that adds KA-specific logic to it.

We should mention here that CyBOK also has textual content that is outside the scope of KAs, namely, *Preface* and *Acknowledgements*. Automating the conversion of these to CyBOK Wiki might not be worth the effort, though the complexity of the LaTeX implementation of these texts can not be asserted in this project as we do not have access to their source.

6.5 Math

Most LaTeX math expressions (used widely in KA *Cryptography*) are converted to Mediawiki by Pandoc reliably. With the MW Extension:Math enabled, MW renders these expressions reliably and correctly, both in-line and blocked expressions.

6.5.1 Renderer

MW Extension:Math by default uses Wikimedia Cloud Services (WMCS), a free SAAS provider to render math expressions to SVG using node.js application Mathoid. The renderer can be tested with arbitrary LaTeX math expressions on here.

If a MW page contains too many math expressions to render, WMCS can be unreliable. In such case, an in-line error message is rendered on MW.

When implementing CyBOK Wiki beyond the proof-of-concept, consider deploying self-hosted Mathoid server to guarantee Math rendering reliability and avoid having to rely on a SAAS.

There are other options to render math expressions, including the MathML markup language, which supported browsers can render natively (see more at Extension:Math). Consider accessibility benefits and browser compatibility drawbacks when choosing MathML over rendered SVGs.

6.6 TikZ

In KA *Cryptography*, some figures are generated by the non-LaTeX standard `tikz` library. See example from KA *Cryptography* PDF page 15 (LaTeX source here):

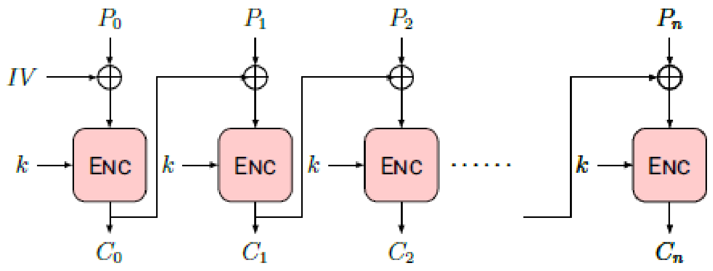


Figure 2: CBC Mode Encryption (All Figures are produced using *TikZ for Cryptographers* *https://www.iacr.org/authors/tikz/*).

Considerations for implementing TikZ support:

6.6.1 WikiMedia extension

TikZ expressions are not supported natively by MW. A MW Extension:PGFTikZ exists but it has not been actively maintained for years. A quick trial of the Extension failed: we installed it and copied the example from the Extension's landing page, as well as one example from the CyBOK LaTeX source - rendering both of them failed without useful error messages. Consider investigating further whether Extension:PGFTikZ can be used for rendering TikZ figures.

6.6.2 Pandoc

TikZ environments embedded in LaTeX are by default omitted by Pandoc. There is a StackOverflow suggestion as to how to implement TikZ conversion to PNG as a Pandas Lua extension, utilizing ImageMagick. This may be feasible, but for CyBOK Wiki, the produced PNG image must be subsequently uploaded and ingested to MW, and embedded to the related Page, to the correct place. As the upload and ingest process is handled by python modules, not Pandoc, using Python instead of Pandoc to manage TikZ rendering may be more practical.

6.6.3 Python

Similarly to the above Pandoc example, a Python preprocessor could be used to manage TikZ rendering, which runs before Pandoc conversion - so by the time the LaTeX source reaches Pandoc, it converts an embedded image, not a TikZ expression.

- Find and extract `{tikzpicture}` environment in LaTeX source by TexSoup.
- Embed the extracted TikZ expression in a minimum LaTeX template, including necessary LaTeX boilerplate to make it a valid standalone LaTeX file.
- Render the (temporary) standalone TikZ LaTeX file, by ImageMagick to PNG.
- Capture and upload the PNG file as media file into MW (use TikZ caption as file name and/or descriptor).
- Embed the uploaded PNG file into the LaTeX source in the place of the original `{tikzpicture}` environment.
- If ImageMagick does not fare well in TikZ conversion (as indicated by some) consider using latex2pdf (or whichever LaTeX to PDF converter is used reliably already by the CyBOK project) to convert the standalone TikZ LaTeX file to PDF, upload and embedding of this PDF as any with any other PDF attachment (see the chapter *6.8 PDFs as illustrations*).

As mentioned in the earlier chapter *6.2 Manual tasks in the automated pipeline,* if the use of TikZ illustrations are rare and isolated, we can consider not automating their conversion, and reverting to manually converting, uploading, and embedding them to the relevant pages.

6.7 Indices, acronyms, glossary (and references)

CyBOK makes heavy use of **index** elements, **acronym**s **and glossary item**s (henceforth referred to collectively as **IAG**s), which - along with sections, illustration and table labels, and respective references - make CyBOK a thoroughly interlinked document. In the CyBOK PDF, they are rendered as lists at the end of KAs or the book, while their inline representations are rendered as hyperlinks that link to the respective definition in these lists. Index elements are not rendered in text: they are marked as anchor points in text without any text element. Their alphabetic lists are rendered at the end of the KA or book, with the associated page numbers rendered as hyperlinks.

CyBOK Wiki, being a web portal, can make excellent use of the linked nature of IAGs: they have the potential to improve UX significantly. Having said that, PDFs and web portals are substantially differing media, specifically regarding their hyperlinking capabilities and good practices. Consequently, the functionality, roles and display of IAGs in CyBOK Wiki should be thoroughly reassessed, considering www and MW-specific UI/UX opportunities and best practices, as well as SEO, and accessibility. In the following, some of these considerations are presented.

6.7.1 Create dedicated pages for acronym and/or glossary item.

On these pages - apart from the title and definition - a list of backlinks [regarding implementation, see: 1, 2] to pages that refer to these IAGs (e.g.: "Related topics") can be displayed, making it easier to discover CyBOK based on cross-cutting topics.

IAG MW pages could be implemented in dedicated (custom) namespaces for indices, acronyms and glossary items. Namespaces are separately templateable, it is easy to implement distinct page, aggregation, search, etc. functionality for them. Namespaces are by default explicitly prefixed in URLs, e.g.: `.../wiki/Glossary:cyberspace` or `.../wiki/Acronym:IoT`, whereas regular articles are by default without a namespace prefix.

6.7.2 Acronyms as tooltips

While glossary descriptions may be several sentences in length, the phrases or concepts that acronyms refer to are typically only a few words long. Acronyms could therefore be rendered such that when the user hovers on (or taps on on their mobile device) them, a hovering tooltip with the full definition appears. This way, the acronym can be disambiguated without the need to navigate away from the article where they are mentioned.

Acronym tooltips could be implemented instead of or in addition to the dedicated MW page for the acronym disambiguation. Dedicated acronym pages would still be useful for providing backlinks to all articles/pages that mention them, and for SEO purposes.

6.7.3 Metadata enrichment with indices

Index items are not rendered in text as they only mark anchor points in text - they do not mark any text element. As such, they would not explicitly appear in MW articles. (Conceivably, they could be displayed as a list of relevant index words for each MW page, or perhaps at the end each paragraph if this is deemed useful.)

Indices, however, are the most widely used type of IAGs throughout CyBOK. In the *Law and Regulation* KA alone there are 2757 `\index{}` elements. Indices essentially offer a high-quality, paragraph (even sentence)-level) human-curated annotation of the entire CyBOK for the most relevant topics mentioned.

Instead of displaying indices in-line, we could extract them and associate them with MW pages (or even sub-units of pages like subsections, paragraphs, even sentences) as MW metadata. This metadata could then enable a number of UX-enhancing functionalities in MW, e.g.:

- **Recommendation engine**: related articles could be recommended for each page, where articles that share the most indices with the page viewed is in the top of the list.
- **Search engine**: metadata coming from indices could be fed into the MW search engine for better results [regarding implementation, see: 1, 2].
- **Data analysis and visualization**: not (necessarily) directly related to or implemented in MW, but the association of indices as metadata with CyBOK (sub)sections can enable all sorts of structured research on CyBOK, e.g.:
    - topic cluster analysis,
    - ontology building,
    - index topics graph visualization, etc.
- **Index list page**: an index list page, listing all indices appearing in CyBOK and their associated MW articles could be implemented - this can be an interesting point of entry into CyBOK for users who browse the resource without a specific aim.
- **Quality assurance services**: indices as metadata could be used to develop internal QA reporting services (probably) independently of MW as python scripts) where the use of indices could be tested against the body of text. For example: see if the text appearing around indices indeed mention the indexed concept, or *vice versa*, to search the text for each index element to find text occurrences where the index is not applied.

While indices are the most used IAG, metadata can be further enriched with acronyms and glossary items in the same fashion.

Extracting indices and associating them with MW pages as metadata could be implemented as part of the python preprocessing and submission pipeline, while MW-specific logic (recommendation, search, index list page) as custom MW Extensions.

6.7.4 References

Similarly to IAGs, references are also represented as structured data in LaTeX, with inline `\ref{}` expressions and reference details sourced from separate `.bib` files. References are - to a limited degree - reused across (sub)sections and perhaps across KAs. Thus, in addition to their primary intended function, references could also be used as IAGs: they could have dedicated pages, linked from the bibliography, with backlinks for discovering related pages, as well as links added to the original publications (where available), or to prepared search query URLs to academic literature search engines such as scholar.google.com or Scopus. Conceivably, bibliographic information such as authors or journals of references could further enrich metadata, allowing for more data analysis (e.g.: topics or sections associated with authors and journals, enabling all sorts of subsequent bibliometric analysis).

6.8 PDFs as illustrations

Some illustrations are included in CyBOK as PDFs. In the three KAs that this exploratory project is concerned with, only one such case exists (*Figure 1* in *Adversarial Behavior* KA). If indeed there are only a handful of such illustrations in the entirety of CyBOK, we can consider not automating their inclusion to CyBOK Wiki (see chapter: *6.2 Manual tasks in the automated pipeline* of this document), as automating this process is quite complicated, as outlined in the following.

6.8.1 Setup process

- MW is installed by default (at least in its official Docker release) with Extension:PdfHandler enabled. This Extension converts PDF files to JPG for embedding into articles.
- Extension:PdfHandler has external dependencies, `ghostscript`, `imagemagick` and `xpdf-utils`. These need to be installed (imagemagick is installed by default in the official MW Docker image). Once they are installed, the Extension integrates with them out-of-the-box without further setup needed.
- PDF files must be allowed for upload in MW settings by adding `$wgFileExtensions[] = 'pdf';` to `LocalSettings.php`

6.8.2 Automation process

- Identify and isolate the `\includegraphics[]{}` LaTeX expression with the TexSoup python preprocessor, and extact the file name from the curly brackets. There is no need to manipulate the expression.
- Rebuild the absolute path of the file.

- Use Python requests and the MW API to upload the file. The upload script must have the token of a logged-in user (files must be owned by a user, so even if we allow anonymous users to upload files, it will not work)
- `\includegraphics[]{}` is by default converted to an internal link to the file by Pandoc, e.g.: `[[AttackTree.pdf]]`. With `Extension:PdfHandler` properly configured, this the desired image will be rendered by MW instead of the hyperlink text. We must however specify an image width, otherwise the image is displayed too large (does not fit the viewport). The syntax is the following: `[[File:AttackTree.pdf|page=1|600px]]`. The necessary arguments can be added with a Lua filter, (e.g.: if link ends in ".pdf"), or can be circumvented with custom MW CSS.

Note: the above steps have been manually verified but have not implemented them in the software proof-of-concept.

6.9 To-dos

Below is a list of miscellaneous technical to-do items for the implementation of CyBOK Wiki beyond the proof-of-concept. The list is not prioritized and the order of items has no significance; we added them to the list as we happened upon them. The list is by no means exclusive.

The proof-of-concept's source code also includes a large number of `# TODO` items which may be relevant, however, these are more directly tied to the current implementation of the proof-of-concept.

- Change the LaTeX CyBOK source such that `\labels` would move above or in-line with the labelled expression, so clicking on the generated HTML anchors links would jump above the named item, making it visible.
- Glossary items are case-sensitive. To render sentence-beginning glossary items that are maintained on record with lower case, `\Gls` instead of `\gls` is used. This behavior is not supported by Pandoc, so the resulting MW markup results in a lower case sentence beginning.
- The `\begin{framed}` LaTeX environment, resulting in PDF as a bordered textbox (see example), is not supported by Pandoc by default, the resulting MW is `<span class="framed">CONTENT</span>`. Consider either
  - using a python preprocessor to change the environment to another fitting one that Pandoc can render,
  - implementing a Pandoc Lua filter to associate the environment with a fitting MW formatting style (e.g. `<blockquote>`),
  - implement custom MW CSS to display the environment as desired.
  - leave as-is and lose framed textboxes in CyBOK Wiki.
- Formatted text or special character tags in (sub)section headings gets removed or incorrectly converted by the python Segmenter (or Pandoc?). For example:
  - \subtopic{\emph{De minimis} exceptions to crimes against information systems} results in the MW page title *"Exceptions to crimes against information systems"*. Or

  - In \subtopic{Breach of contract \& remedies}, the `&` character doesn't get escaped, resulting in the MW page title `Breach of contract \& remedies`
  - (While investigating this, we came across subsection title *"Σ-Protocols"* (*Cryptography* KA subsection 9.3.1), which gets converted to a correct heading by Pandoc, but in the CyBOK PDF, in the PDF bookmark list, the Σ is omitted.)

- `\fullref` is consistently omitted, even though it should be supported by Pandoc. See for example in KA-final cross-reference tables.
- The *Cybersecurity* KA final cross-reference table is converted to invalid MW markup, while the similar table at the end of the *Adversarial Behaviour* KA rendered correctly.
- Implement proper MW citations with ieee.cls, yielding a nicer view and more functions, similar to how notes already work. Use Lua to format in-line, citation points and a python preprocessor (or postprocessor)? to add *References* heading and bibliography data at the bottom of pages. Interfacing directly with `.bib` files using the `bibtexparser` python library may be necessary for the preprocessors.
- References to tables should work, e.g.: Table~\ref{tab:ka_law:standards} should link here. For this, Pandoc generates a broken MW link `[[tab:ka_law:standards]]` as if the table were a MW page. Instead, it should generate a link to the subsection that hosts the table (subsection *"A more holistic approach to legal risk analysis"*), with an anchor link to the table (correct MW markup: `[[sect:ka_law:holistic#tab:ka_law:standards|Example standards of proof]]`) To implement this the python preprocessor should extract the (sub)section IDs where a table belongs, and, find the `\ref{}` references to the table and replace their reference attribute as `section_id#table_id`.
- Table descriptions are sometimes omitted during conversion, see e.g.: *Cybersecurity* KA, Figure 1.
- Citations that in the CyBOK PDF appear aligned right in a separate row at the beginning of KAs or main sections not aligned right nor displayed separate row in CyBOK Wiki (see *Appendix* for example).
- Some in-line math expressions are not properly separated from the word before or after them (see *Appendix* for example).
- LaTeX `{array}` and `{eqnarray*}` environments are not rendered at all, the reason for this needs to be further investigated.

7. Open questions

In this chapter, we provide a detailed discussion of open questions that must be considered, debated and eventually answered to enable moving forward with the implementation of CyBOK Wiki beyond the proof-of-concept.

7.1 Which CyBOK PDF (sub)sections should be segmented into CyBOK Wiki pages?

CyBOK is a heavily hierarchical document. KAs are divided into section, subsections (and further smaller units). In LaTeX, this is achieved by the commands `\section{}`, `\subsection{}`, and `\subsubsection{}`, as well as their CyBOK-specific synonyms `\topic{}` and `\subtopic{}` (all of these henceforth collectively referred to as *(sub)sections*).

In the CyBOK PDF, these expressions are rendered as hierarchical headings, prepended with incrementing section numbers (and sub-numbers for lower hierarchy elements, subsections), and are fed into the document's Table of Contents (ToC). Asterisked commands (e.g.: `\section*{}`) form an exception: they are rendered with the same style as their asteriskless counterparts, but without numbering.

In LaTeX, (sub)sections may have related `\label{}` expressions. `\label{}`s label any point in a LaTeX document with a unique label (not rendered), and thus the labelled point can be referenced to from other parts of the document. For example, if the text in Section *A* makes a reference to subsection *B.C*, section *B.C* needs a label so the reference to it can be rendered (as a hyperlink). Not all (sub)sections have a related `\label{}` expression - these are not referenced to by other parts of the text in CyBOK (but they still get assigned a section number and appear in the ToC).

To take advantage of CyBOK's heavily hierarchical nature, (sub)sections can be implemented in MW as separate pages that link to one another. Here we discuss possible strategies and considerations regarding how this could be implemented:

Pages should not be too short, or too long: It is not the task of this project to propose an ideal (and minimum, or maximum) length of MW pages. However, if such guidelines are to be set, they can help inform the decision about the segmentation strategy. The segmentation strategy of Wikipedia may be of use.

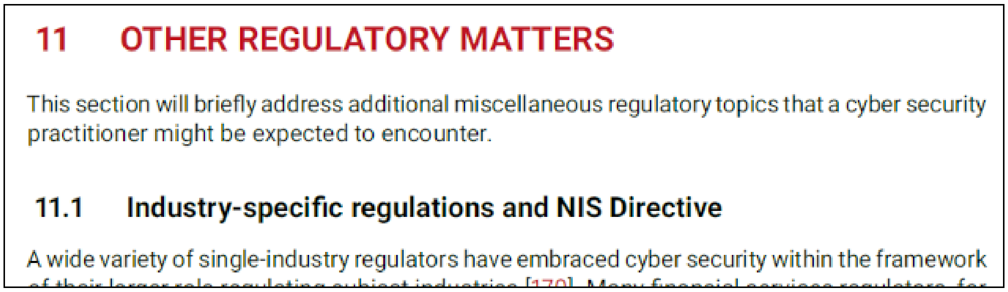
Knowledge Area page: KAs are not marked by the above sections system: they are the top level in CyBOK hierarchy (apart from KA groups) and are segmented into LaTeX project folders. Those texts of a KA that are not segmented into (sub)sections should belong under the KA's page - a KA landing page of sorts - along with links to the (sub)sections belonging to the KA.

Every (sub)section with a label becomes a page: This is the practical minimum strategy to divide (sub)sections into pages. (Sub)sections that have labels may have other parts of the text make reference to them. These references should appear in CyBOK Wiki as hyperlinks - to the dedicated page of the referred (sub)section.

Every (sub)section becomes a page (even ones without a label): Unlabelled (sub)sections are not referenced by other parts of CyBOK, but they can still be segmented into dedicated pages. This way, while browsing CyBOK Wiki, the reader may open these sections interesting to them for later reading in new browser tabs, reference these sections directly in their own work, share them directly in emails or social media, etc.

Auxiliary (asterisked) (sub)sections do not become pages: Similarly to the CyBOK PDF, these pages may not need to be segmented into dedicated pages. Typical examples: *Introduction*, *Conclusion*, *Cross-reference tables*, or *Further reading* logically belong to their parent section's page (typically the KA page) (this is not implemented in the proof of concept).

Avoiding very short pages: Some sections have very little content on their own as they mostly serve as a collection for other subsections. See for example *Law and regulation* KA section 11, with only one content sentence before its subsections begin:



11 OTHER REGULATORY MATTERS

This section will briefly address additional miscellaneous regulatory topics that a cyber security practitioner might be expected to encounter.

11.1 Industry-specific regulations and NIS Directive

A wide variety of single-industry regulators have embraced cyber security within the framework of their larger role regulating subject industries [170]. Many financial services regulators for

We could consider not creating separate pages for such sections as such small pages with no substantial content offer little value to the user. How to identify and handle such cases, however, is not a trivial question.

- Automatic identification of short pages: this could be achieved with the Python TexSoup preprocessor (i.e.: by counting the word length of the page after the segmentation and identifying ones under a certain threshold), however, this may not be optimal: in certain cases, short pages might be justified.
- Manual identification of short pages: the CyBOK LaTeX standard could be expanded by redefining the (sub)sections commands with a new optional parameter that would indicate that the (sub)section must not be segmented into a page. This parameter would not have any effect in rendering the CyBOK PDF, it would work only for CyBOK Wiki. Parsing this argument would be implemented as a Python preprocessor.
- Embed short section in their parent's page: instead of creating a dedicated page for short contents, they could be embedded to their parent's page.

(Sub)sections that do not become pages should be converted into headings: MW automatically assigns heading elements with a unique HTML `id`, making them linkable as anchors inside their parent page.

7.2 Extracting (sub)chapter titles and identifiers

As described *above*, (sub)section titles may have related `\label{}` LaTeX expressions that label a certain point in the document that can then be referenced from other points in the document using `\ref{}`. In other words, when referencing a section, the (sub)section titles are not referenced directly, rather the accompanying label is referenced. The (sub)section title and label do not necessarily belong together: (sub)sections can be defined without a label (see *above*) and labels can be placed anywhere in the document.

The way (sub)section titles and labels are used together in CyBOK is not defined or consistent. Sometimes labels follow (sub)section titles directly:

```
\topic{The Elements of a Malicious Operation}
\label{sect:elements}
```

while in other instances there are `\index{}` expressions separating the (sub)section title and the corresponding label (and we expect further variations may exist in other KAs):

```
\topic{Information-theoretically Secure Constructions}
\index{information-theoretic security}
\index{information theory}
\label{crypto:sec:IT}
```

Linking (sub)section titles with labels is a crucial task: (sub)section titles are the expressions that segment the LaTeX source into MW pages and their values become the title of these pages (as well as the primary identifier of them, in MW), but references to them in the LaTeX source are made by the label. We can only retain linkages between in-text references to pages and the referenced pages if we can make a linkage between (sub)section titles and labels. However, for the above described lack of standardization in the use of (sub)section titles and labels makes it hard to reliably associate them with one another.

For this reason, we recommend that the LaTeX source is changed such that label expressions are embedded in the (sub)section title expressions, e.g.:

```
\topic{The Elements of a Malicious Operation\label{sect:elements}}
```

This way the relationship between the (sub)section title and label is inseparable and unambiguous. Identifying and maintaining these linkages in a python preprocessor and subsequently using them for reference preparation, metadata addition, etc. is a straightforward task with the TexSoup library.

This proposed standard does not change the rendering of the CyBOK PDF in any way but is crucial to one of the core innovations of CyBOK Wiki: segmenting (sub)sections into dedicated MW pages.

To demonstrate the software proof-of-concept's in-text cross-reference capabilities the LaTeX changes proposed here have been implemented for the *Adversarial Behaviors* and *Cryptography* KAs, see `data/ab/content.tex` and data `data/c/content.tex`, respectively. The LaTeX source of the *Law and Regulation* KA has not been altered, consequently, cross-reference links are broken for this KA.

7.3 How to implement and display CyBOK's structure in MW pages?

Structure in the CyBOK PDF (and the underlying LaTeX source) is implemented by separate LaTeX projects (for KAs) as well as (sub)sections in LaTeX files. These are then rendered in text as numbered (sub)headings, chapter indicators in PDF page footers, table of contents, and PDF bookmarks. These are logical and practical ways of implementing and displaying structure for the PDF format and print, but not necessarily best practice (or feasible) for a web portal such as MW. MW is not an intrinsically hierarchical platform in terms of how it manages pages, but it offers several ways to implement hierarchical page relations. In the following, we discuss the following considerations for implementing and displaying different levels of structure in CyBOK Wiki.

7.3.1 Namespaces

Namespaces by design separate content pages from other kinds of MW pages such as MW users' profiles, files, or discussion pages. However, custom namespaces may be defined - these could be used for one level of structure of CyBOK Wiki. Namespaces are individually templatable - if this is deemed useful for CyBOK Wiki, they should be considered. However, they only offer one level of namespacing, thus are not ideal for structuring the multi-level hierarchy of CyBOK. Perhaps it may be useful for the highest level (KAs).

7.3.2 Categories

"Categories provide automatic indexes that are useful as tables of contents. You can categorise pages and files by adding one or more Category tags to the content text. These tags create links at the bottom of the page that take you to the list of all pages in that category, which makes it easy to browse related articles."

With this, categories might be the most fitting solution to structuring CyBOK Wiki. They are hierarchic, an integral concept to MW, and offer display mechanisms by default. Semantically, *category* is not the same as sections as they appear in CyBOK, and the functions that Categories provide might not be all sufficient (or relevant to) CyBOK Wiki, but it might still be the ideal concept to base CyBOK's hierarchy on.

Instead of Category pages (displaying a list of pages in a category, without any content), could be considered. With this extension, a collapsable tree of the (sub)categories, and the pages belonging to them can be rendered automatically. This could be embedded to pages that have related (sub)chapters.

7.3.3 Subpages

"Subpages introduce some hierarchical organization into wiki pages, with levels of the hierarchy separated by slashes (/)."

By default, subpages are disabled for the main *Namespace* (where content pages exist), but it can be enabled. With this, namespaced URLs may be implemented, e.g.: `.../wiki/Law_and_Regulation/Jurisdiction/Territorial_jurisdiction`, which may be deemed necessary to provide users context on the URL level (or the opposite: it may be found to be too verbose). Subpages offer breadcrumbs (see for example here). Built-in templating functions can be used to display all subpages of any page. MW documentation suggests that Categories are in general better suited for organizing hierachical information, so subpages should be considered primarily for their verbose URLs.

7.3.4 Embedding child sections

Embedding the contents of child sections into the parent page can be considered (in addition to, or instead of linking to the child page). It can be implemented in MW on the templating level simply by referring to a section with a special syntax. With this, when opening a parent page (perhaps even the page of a whole KA), all of its contents would be rendered and could be browsed without having to move away of open separate tabs. The utility of this is questionable: this will essentially result in duplicate content on the front end (this can be confusing for users and potentially detrimental to SEO), and may result in undesirably long pages. If implemented on the KA level, the added value of CyBOK Wiki (compared to the original CyBOK PDF release) is arguably diminished, as it essentially reduces the segmentation that Wiki enables. With all that said, embedding can be considered for certain cases (perhaps individual cases).

7.3.5 Displaying structural elements

In addition to the above possibilities, other structure indicating UI elements such as breadcrumbs, sidebars, info boxes, etc. may be considered. Either way, on each page, the following related pages should be linked to in some way:

- parent (sub)section
- child (sub)section(s)
- previous section and next section (as if turning pages in the CyBOK PDF)
- (sibling sections?)

These may have to be implemented in MW templating as custom functionality, depending on which one of the above options is selected.

7.3.6 Namespaces in URLs

By default, the title of a MW page is also the unique part of the MW URL to that page. CyBOK may have (sub)sections that share with another (sub)section across the 22 KAs (e.g.: *Conclusions*). For such cases at least, some namespacing should be added in the URL, otherwise one or the other (sub)section will be overwritten. We consider Subpage's above described approach, or Wikipedia's disambiguation convention, e.g.: Flash (DC Comics character) and Flash (lake).

In any case, we may want to include some namespacing to URLs to give context to the topic. E.g.: section *9.1.3 Conceptual Models* is an ambiguous title. We may want to add the KA's title for context (e.g.: *Conceptual Models (Forensics)*). At the same time, adding the KA title as context to all titles may in cases prove superfluous and repetitive, for example consider: *"Why is risk assessment and management important? (Risk Management and Governance)"*.

We can thus see that deciding whether the KA's title should be added to a page's URL for context is neither trivial nor easy to automate. With this in mind, we can consider changing the CyBOK LaTeX standard by adding an optional parameter to (sub)section LaTeX expressions that indicate whether the parent KA's title should be included in the URL (and perhaps the page title?) for namespacing (without any behaviour when rendering the CyBOK PDF.)

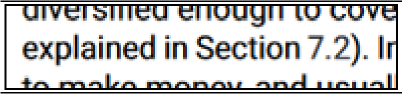
7.4 How to display section titles?

In the CyBOK PDF, (sub)section titles are automatically assigned an incremental unique section number, e.g: *3*, *3.1*, or *3.1.3*, etc. based on the hierarchic depth and the order of the (sub)section (except for asterisked auxiliary sections like `\section*{}`. Section numbers are automatically prepended to (sub)section titles and their corresponding table of content entries and PDF bookmarks. Section numbers are also used when referring to (sub)sections in-text - in this case, only the section numbers are rendered. E.g.: In the Cryptography KA the section *Standard protocols* is assigned section number *8*, so the section title is rendered "*8 Standard protocols*", reference to this section may be made by the LaTeX fragment `in Section \ref{crypto:sec:protocols} we discuss` which will be rendered simply as "*in Section 8 we discuss*".

Unlike a PDF, a Wiki is a non-linear medium, and CyBOK (sub)sections are written in such a manner that they may be - to a large extent - read in a non-linear order. Therefore, in CyBOK Wiki we do not necessarily have to maintain section numbering (implying that there is an order amongst sections) and instead may always render only the (sub)section's title.

This is a straightforward change to implement for MW page titles, but in-text references offer a bit more nuanced dilemma. Throughout CyBOK, the text is edited so that section references are always referred to as "*Section **NUMBER***" (i.e.: the number is always prefixed with "*Section*"). The question must be asked whether switching the numbers to the corresponding (sub)section's title will reliably leave the text logical and easy to read. To this end, the following considerations are offered:

- The references will be visibly differentiated because of the hyperlink - this should be of help with readability.
- Quotation marks around the referenced (sub)section title could be added to further ease readability (see examples below).
- The *Section* prefix of references could be altogether removed. From a technical perspective, this is bit more complex to implement: in a python preprocessor, `\ref{}` expressions must be isolated and the preceding text block altered if it ends with the word "Section".

PDF original	 diversified enough to cove explained in Section 7.2). Ir to make money and usuall
like PDF original	...explained in Section 7.2...
with number and title	...explained in Section 7.2 "A Characterisation of Adversaries"...
only title	...explained in Section "A Characterisation of Adversaries"...
removing *Section* prefix	...explained in "A Characterisation of Adversaries"...

All of the above is valid also to the titles of and references to figures, tables, illustrations, etc.

7.4.1 Numbering level

In the full CyBOK PDF release (that is, the >1000 pages long PDF release with all KAs), the first number of the section number is the number of the KA in the order as they appear (e.g.: *10.5.2 Message Authentication Codes* on page 336). In the separate KA PDF releases, the KA number is omitted and only the (sub)section numbers are used (e.g.: *5.2 Message Authentication Codes* on page 17). If section numbering is retained in CyBOK Wiki, decision must be made whether to use the full or KA-level section numbering.

7.5 Migrating LaTeX metadata to MediaWiki

CyBOK KAs have related metadata that in the CyBOK PDF are mostly rendered on the title page and following lead-in pages, including:

- Title,
- Version,
- Author(s) and their affiliation(s),
- Editor(s) and their affiliation(s),
- Reviewer(s) and their affiliation(s),
- Copyright information,
- Changelog,
- Release month and year (in page footers).

These metadata are not defined in the main content LaTeX file, rather in separate LaTeX files, as well as LaTeX class and style files, along with their complicated design definitions.

Extracting these metadata in an automated manner is feasible by implementing custom python TexSoup processors that interface with these files and isolate and extract the metadata, and subsequently feed it into the main python preprocessing and segmenting processes. While this is feasible, this is a relatively complex undertaking, given the unique and formatting-heavy definition of each of these metadata, especially considering that the automation runs only 22 times (i.e.: for each KA). We may consider extracting these metadata manually, maintaining them in an easy-to-process structured format (e.g. in `yaml` files) for each KA (see chapter *6.4 KA-specific functionality*).

If KA metadata is extracted, we must consider how MW should display or otherwise use them. These metadata describe KAs in the CyBOK PDF, however we might consider adding them to each page that belong to the KA. The could be displayed in an Infobox, or in the page footer. The concept and function of CyBOK Wiki differs from most MW use cases (that is, open encyclopediae like Wikipedia) that do not (explicitly) display much of the above metadata, so finding the most fitting display of them will require dedicated UX/UI design effort and implementing them might require custom templating work.

These metadata can also be used to enrich MWs interactive functionality such as search or recommendations – see more on this in subchapter *6.7.3 Metadata enrichment with indices* of this document.

In addition to KA-specific metadata, consider also additional CyBOK wide metadata (that this exploratory project does not access the LaTeX source of), such as the names of CyBOK-level editors, project managers, production personnel, professional and academic advisory board members, steering committee members, etc.

7.6 Versioning

If CyBOK Wiki is implemented and successful, eventually new versions of CyBOK may be released on it. This is a more distant concern than others mentioned in this document, but it is one that should be considered already at the inception of CyBOK Wiki. Here we will only outline some issues and possible ways to tackle them, without going into much detail.

7.6.1 Are older versions of CyBOK Wiki to be retained?

The answer is probably yes: the older version of the CyBOK PDF (v1.0) is still available for download, as indeed it may be referenced and relied on by existing resources. There is no reason why the same consideration should not apply to CyBOK Wiki.

7.6.2 Namespacing as versioning

We have discussed the issue of namespacing regarding structuring, *above*. The Namespace concept of MW might be used for versioning, such that each page's URL belonging to a version would be prefixed with the version number. Ideally, version URL namespacing should be added already at the first ever release, so when a subsequent release is made, the older URLs do not need to be changed (to guarantee link longevity without having to maintain a complex redirecting functionality).

7.6.3 Continuity

Like with any versioning, it would be practical to show users visiting older pages that a newer version of the page is also available. This is only possible if a connection is made between old and new versions of the same page. This is feasible if pages (or rather, their underlying LaTeX code) have persistent and uniquely identifying features - which currently they do not. The labels of (sub)section titles should all be unique (otherwise referencing to them would not work), however, they are not guaranteed to persist across versions, and not all (sub)sections that may be turned into pages have corresponding labels. We can consider enforcing a stronger, persisting use of labels in the CyBOK LaTeX standard, however, the question remains as to what happens to restructured (sub)sections (e.g.: (sub)sections' content remains but gets split into several new (sub)sections, or several (sub)sections joined in one, etc.).

SEO must be considered so search engine users would by default always find the most up-to-date CyBOK versions, but could also find old versions if need be - this is especially important if no continuity between page versions is established.

8. Conclusions

The primary question of this study was the technical feasibility of CyBOK Wiki, an automated software framework that publishes CyBOK content to a browser-based Wiki system. The proof-of-concept software framework demonstrate that the general concept of CyBOK Wiki is indeed feasible. It produces a browser-based knowledge portal with sufficient UI and UX. CyBOK is transferred to WikiMedia with rendered textual formatting, headings, tables, math formulae preserved, references and cross-references included as hyperlinks, and (sub)sections segmented into standalone pages.

The feasibility of key qualities and features expected of CyBOK Wiki is demonstrated:

- **User experience:** Basic navigation and cross-references are added, demonstrating improved navigation across (sub)sections (compared to the CyBOK PDF). Copying and pasting from HTML retains formatting, while doing so from PDF is mostly unsuccessful.
- **Accessibility:** MediaWiki ships with a responsive layout out-of-the-box, improving accessibility of CyBOK, especially on mobile devices. Cybok Wiki is rendered by MediaWiki in standard HTML, much better suited for screen readers than PDF.
- **Discoverability:** Cybok Wiki is rendered by MediaWiki in HTML adhering to contemporary standards that are preferred by search engines. CyBOK Wiki segments (sub)sections into standalone pages - users can link directly to them, rather than KA PDFs only.

The lessons learned of the development process suggest that the more complex problems beyond the scope of the proof-of-concept are also surmountable. Implementing LaTeX-to-MediaWiki conversion mechanisms that are not supported by the conversion software Pandoc can be a complex task but there are several complementary tools and methods to employ. Rare but complex conversion tasks may be left out of automation - how to do this in a sustainable, reliable way is discussed in details in chapter *6.1 Tackling LaTeX expressions that are not converted correctly by Pandoc*.

In some cases, changing the CyBOK LaTeX source (that the PDF release is also generated from) may be the best (or only) solution for solving conversion issues. Changing LaTeX source might be the most immediate effect that a prospective CyBOK Wiki project may have on the CyBOK (PDF) parent project, and indeed, in case of implementing in-text cross-references between (sub)sections, this may be necessary. Strategies and best practices to this end are discussed in details in chapter *6.3 Changing the LaTeX source*.

As a first call-for-action, the chapter *7 Open questions* outlines issues that must debated and answered to implement CyBOK Wiki beyond the proof-of-concept. These questions are mostly concerned with how the change of medium from PDF to a web portal affects how we treat CyBOK as a structured document, what new functions we want to give it and how we want to present it on a new user interface. We finally outline the next technical steps to be taken towards the realization of CyBOK Wiki the final chapter *below*.

8.1 Risks and limitations

The results of this feasibility study are promising but the implementation of CyBOK Wiki beyond the proof-of-concept is not without risks.

As discussed the chapter *7.2 Extracting (sub)chapter titles and identifiers*, the realization of CyBOK Wiki would likely require some degree of interference of the CyBOK LaTeX source code. There are no such changes or standardizations foreseen that would impact the appearance or functions of the canonical PDF release of CyBOK. However, changes to and standardization of the LaTeX code base will require additional work: editing the existing code base (which would not concern content, only code syntax or structure) and enforcing new standards to subsequent releases. The latter would impact the work of authors and content editors (or in case this was to be avoided: create additional work for the maintainer of the LaTeX code base).

CyBOK is a complex and vast resource. It is written in a complex, dynamic language, LaTeX, converted by another complex technology, Pandoc. This complexity makes it difficult to estimate resources required for a prospective CyBOK Wiki implementation. The definition of a minimum viable product (MVP), establishing clear success indicators and an iterative, agile implementation process that involves all relevant stakeholders would be a good strategy to navigate such a complex project to success.

This project has not attempted to create a systematic and comprehensive QA reporting tool that would identify and keep track of LaTeX features that Pandoc does not convert (correctly). We assume that developing such a tool is itself a complex undertaking. A realistic MVP, clear success indicators and a good manual QA testing processes can circumvent the need to create such a systematic tool.

The results and conclusions provided here are limited by the fact that only three of the 22 Knowledge Areas were accessible to the study. If the study were to be expanded to CyBOK as a whole, some of the results or conclusions might differ or indeed fail. The feasibility of this study had a technical focus - a successful implementation project would also require strategic, operational, editorial, managerial, organizational, financial, etc. planning.

9. Next steps

In this final chapter we outline a list of further steps to consider when planning and implementing CyBOK Wiki beyond the proof-of-concept. Again, we only focus on technical aspects.

- Continue the exploratory research on the remaining 19 KAs. Expand the proof-of-concept software to explore the feasibility and fitting strategies for new findings. Define additional (and refine existing) technical considerations and open questions.
- Design a systematic, deterministic, precise and (to the extent that it is possible) automatic workflow to track conversion quality, as well as omitted, or incorrectly converted LaTeX features. This could be the backbone of CyBOK Wiki's quality assurance mechanism against the canonical PDF release.
- Select and apply a suitable UX/UI/Service design methodology to plan and design CyBOK Wiki. Prospective users' needs should not be assumed but be researched systematically. It is important to keep in mind that CyBOK Wiki should not just be a browser-based clone of the CyBOK PDF - it is a new medium and service with other functions, use cases, risks, and limitations than a PDF release. CyBOK Wiki is a chance to rethink what CyBOK is, and what else it may become.
- Consider building the CyBOK Wiki conversion pipeline upon the proof-of-concept, or design a new architecture potentially reusing snippets from the proof-of-concept.
- Create plenty of unit tests. One way of guaranteeing consistent and reliable conversion results as the number of pre- and postprocessors custom Lua filters and potentially other conversion tools grow. Test python units separately, but also consider creating integration tests with complex LaTeX test snippets.

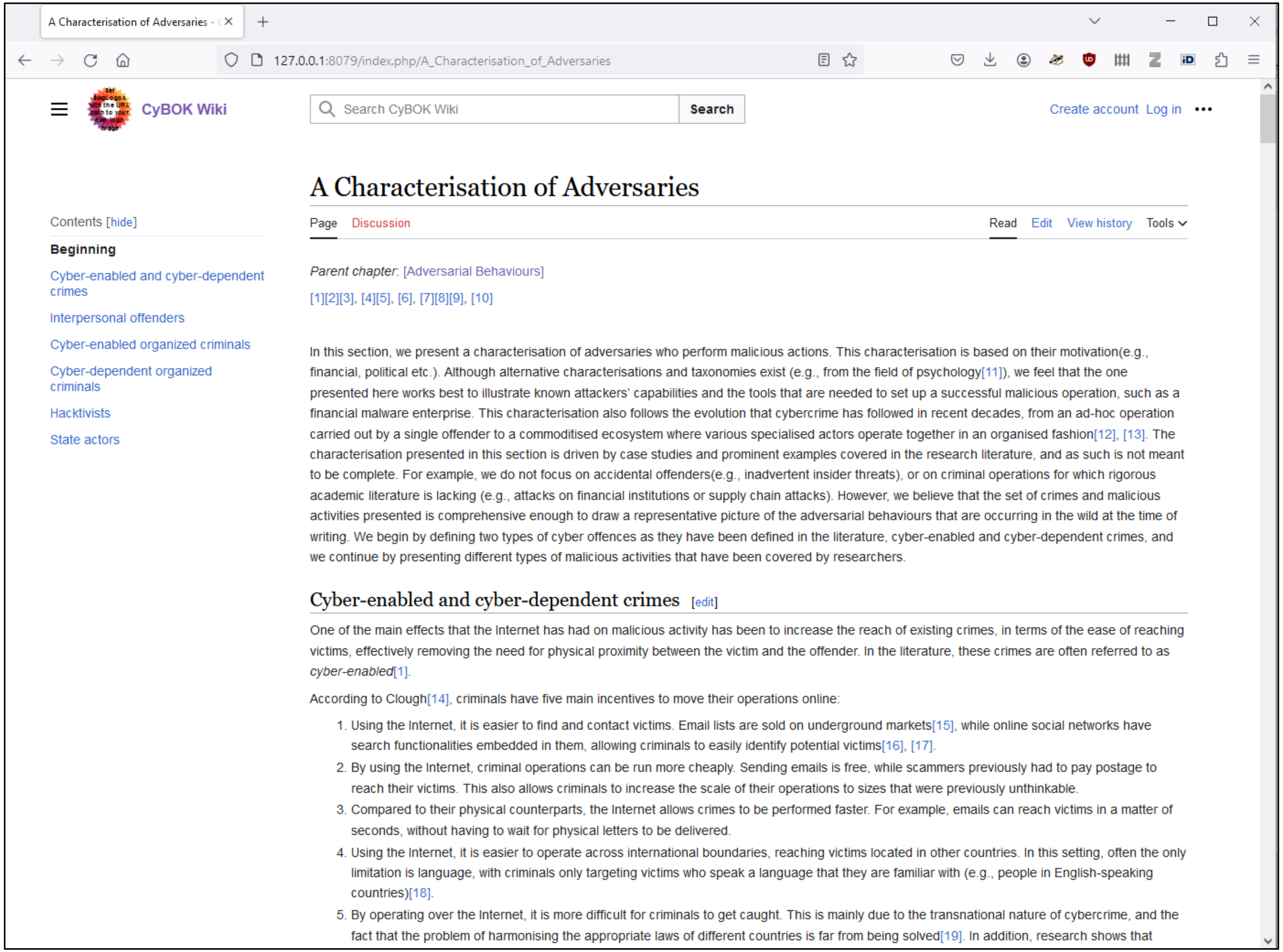
- Debate and answer open questions gathered in this document (and new ones that further research might yield). These will inform the process of defining and designing CyBOK Wiki as a platform and as a service.
- Define the requirements for a MVP and create a prioritized task list.
- Consider an agile method for developing CyBOK Wiki beyond the proof-of-concept. Establish the whole conversion pipeline and platform early to be able to assess results early and often. Estimating the resource intensity of implementing the conversion of different LaTeX phenomena is difficult; seemingly straightforward concepts might prove to be deep rabbit holes, while other objectively complex cases might be handled by Pandoc seamlessly. Therefore, an agile method would be useful to be able to change priorities of an MVP during the project and remain efficient and effective.
- Harden MediaWiki. CyBOK Wiki will not be the typical MW use case in that it will most likely not have an open collaborative element, therefore much of MW's functionality will not be used. For the benefit of security and UX, disable every functionality that does not serve MW's use case, both on the frontend and the backend. After content ingestion, disable also all functionalities that were used for content ingestion, for example the API.
- Consider methods of data collection on the usage of CyBOK Wiki in a manner that provides CyBOK editors with valuable insights, but without tracking individual users.
- The development of CyBOK Wiki requires the methodological and meticulous probing and processing of the CyBOK LaTeX code base. This provides for the emerging opportunity to create (at low additional cost) auxiliary quality assurance reporting scripts, linters, or even semantic, syntactic or bibliometric analysers of the CyBOK code base and body of text. Consider seizing these opportunities.

CyBOK Wiki: feasibility study

Appendix

Lőrinc Thurnay | University for Continuing Education Krems

CyBOK Wiki


A Characterisation of Adversaries - (
127.0.0.1:8079/index.php/A_Characterisation_of_Adversaries
CyBOK Wiki
Search CyBOK Wiki
Search
Create account
Log in
Page
Discussion
Read
Edit
View history
Tools

A Characterisation of Adversaries

Parent chapter: [Adversarial Behaviours]

[1][2][3], [4][5], [6], [7][8][9], [10]

In this section, we present a characterisation of adversaries who perform malicious actions. This characterisation is based on their motivation(e.g., financial, political etc.). Although alternative characterisations and taxonomies exist (e.g., from the field of psychology[11]), we feel that the one presented here works best to illustrate known attackers' capabilities and the tools that are needed to set up a successful malicious operation, such as a financial malware enterprise. This characterisation also follows the evolution that cybercrime has followed in recent decades, from an ad-hoc operation carried out by a single offender to a commoditised ecosystem where various specialised actors operate together in an organised fashion[12], [13]. The characterisation presented in this section is driven by case studies and prominent examples covered in the research literature, and as such is not meant to be complete. For example, we do not focus on accidental offenders(e.g., inadvertent insider threats), or on criminal operations for which rigorous academic literature is lacking (e.g., attacks on financial institutions or supply chain attacks). However, we believe that the set of crimes and malicious activities presented is comprehensive enough to draw a representative picture of the adversarial behaviours that are occurring in the wild at the time of writing. We begin by defining two types of cyber offences as they have been defined in the literature, cyber-enabled and cyber-dependent crimes, and we continue by presenting different types of malicious activities that have been covered by researchers.

Cyber-enabled and cyber-dependent crimes [edit]

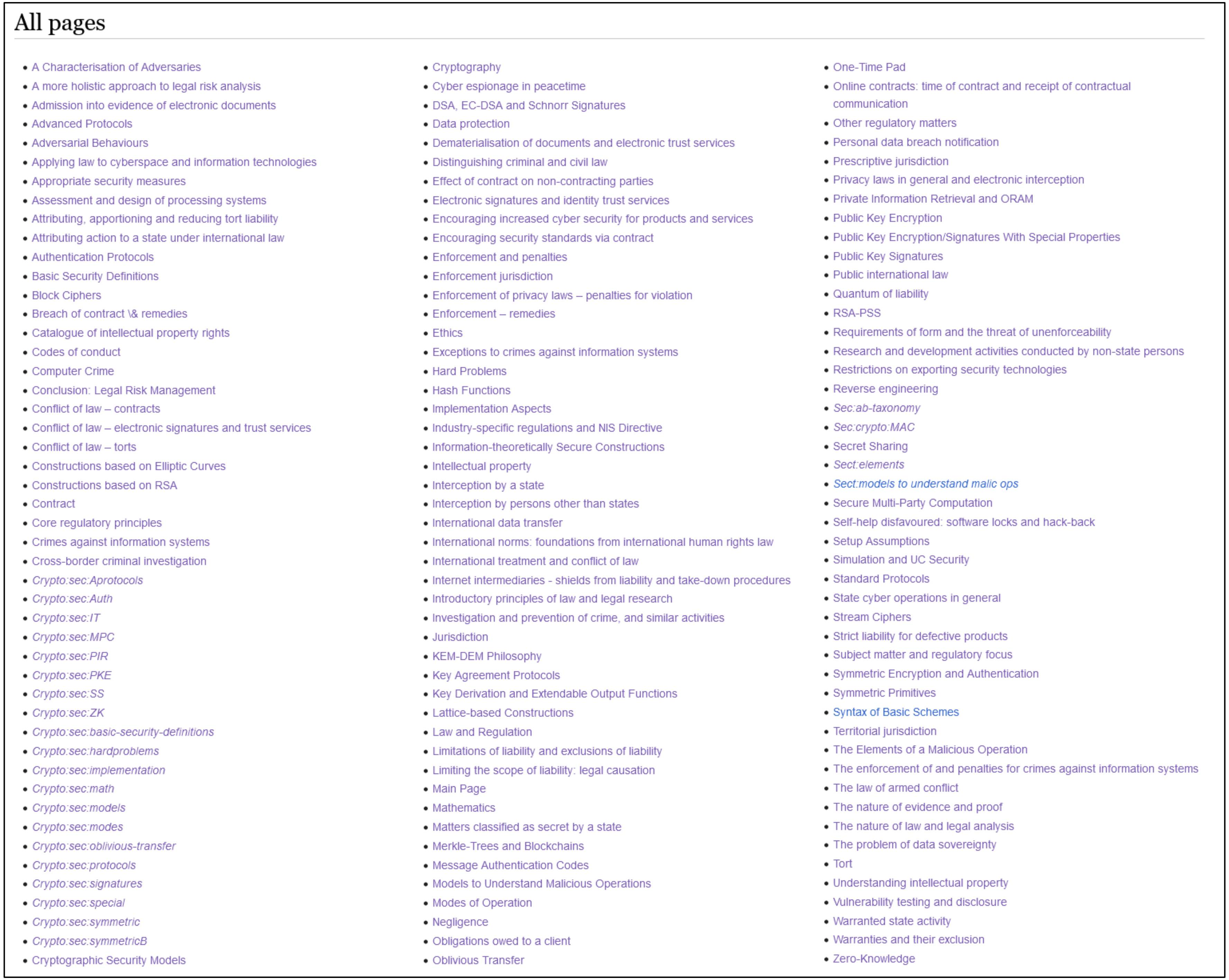
One of the main effects that the Internet has had on malicious activity has been to increase the reach of existing crimes, in terms of the ease of reaching victims, effectively removing the need for physical proximity between the victim and the offender. In the literature, these crimes are often referred to as *cyber-enabled*[1].

According to Clough[14], criminals have five main incentives to move their operations online:

1. Using the Internet, it is easier to find and contact victims. Email lists are sold on underground markets[15], while online social networks have search functionalities embedded in them, allowing criminals to easily identify potential victims[16], [17].
2. By using the Internet, criminal operations can be run more cheaply. Sending emails is free, while scammers previously had to pay postage to reach their victims. This also allows criminals to increase the scale of their operations to sizes that were previously unthinkable.
3. Compared to their physical counterparts, the Internet allows crimes to be performed faster. For example, emails can reach victims in a matter of seconds, without having to wait for physical letters to be delivered.
4. Using the Internet, it is easier to operate across international boundaries, reaching victims located in other countries. In this setting, often the only limitation is language, with criminals only targeting victims who speak a language that they are familiar with (e.g., people in English-speaking countries)[18].
5. By operating over the Internet, it is more difficult for criminals to get caught. This is mainly due to the transnational nature of cybercrime, and the fact that the problem of harmonising the appropriate laws of different countries is far from being solved[19]. In addition, research shows that

Browser screenshot displaying the entire MW interface. Section headings appear in the sidebar automatically.

CyBOK Wiki



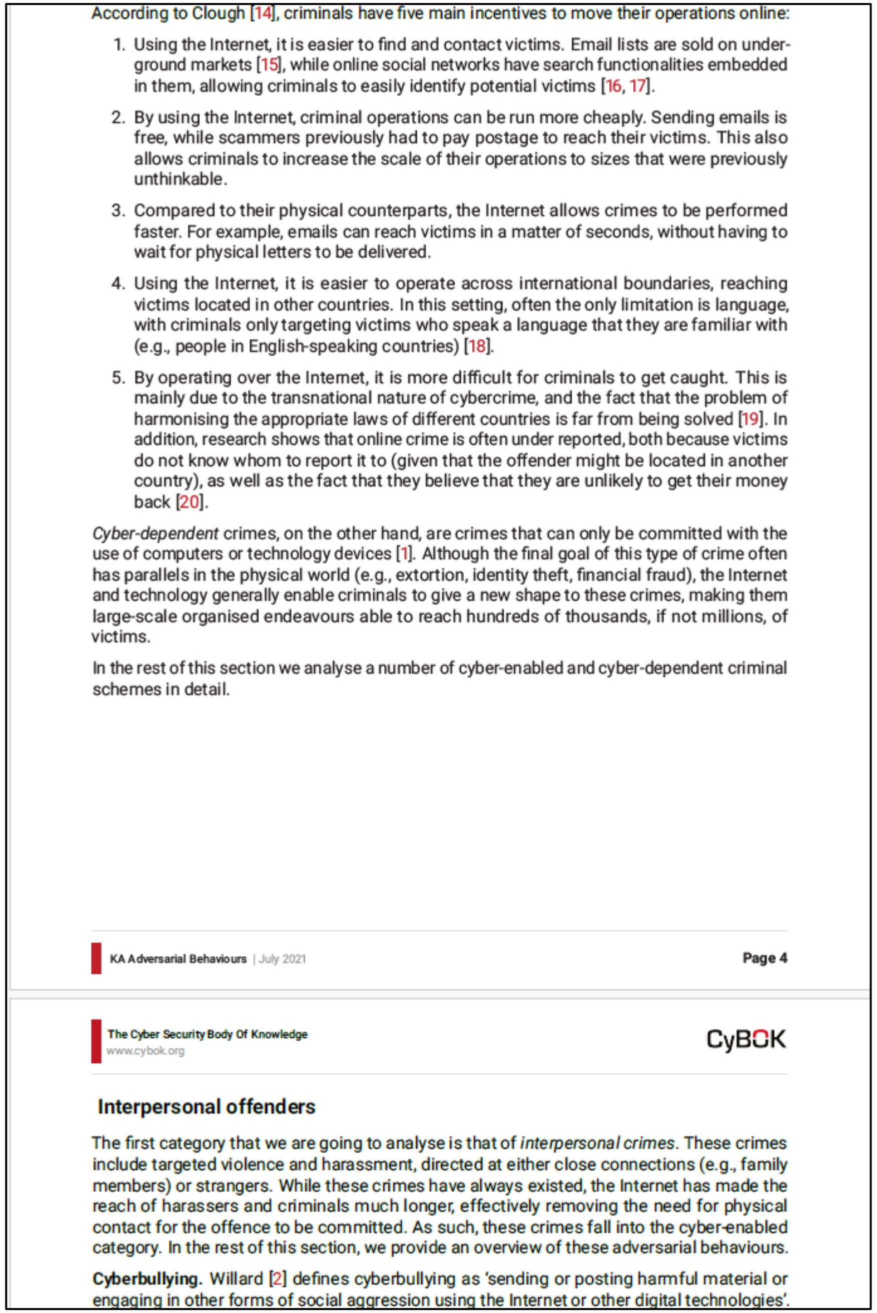
All pages

- A Characterisation of Adversaries
- A more holistic approach to legal risk analysis
- Admission into evidence of electronic documents
- Advanced Protocols
- Adversarial Behaviours
- Applying law to cyberspace and information technologies
- Appropriate security measures
- Assessment and design of processing systems
- Attributing, apportioning and reducing tort liability
- Attributing action to a state under international law
- Authentication Protocols
- Basic Security Definitions
- Block Ciphers
- Breach of contract \& remedies
- Catalogue of intellectual property rights
- Codes of conduct
- Computer Crime
- Conclusion: Legal Risk Management
- Conflict of law – contracts
- Conflict of law – electronic signatures and trust services
- Conflict of law – torts
- Constructions based on Elliptic Curves
- Constructions based on RSA
- Contract
- Core regulatory principles
- Crimes against information systems
- Cross-border criminal investigation
- *Crypto:sec:Aprotocols*
- *Crypto:sec:Auth*
- *Crypto:sec:IT*
- *Crypto:sec:MPC*
- *Crypto:sec:PIR*
- *Crypto:sec:PKE*
- *Crypto:sec:SS*
- *Crypto:sec:ZK*
- *Crypto:sec:basic-security-definitions*
- *Crypto:sec:hardproblems*
- *Crypto:sec:implementation*
- *Crypto:sec:math*
- *Crypto:sec:models*
- *Crypto:sec:modes*
- *Crypto:sec:oblivious-transfer*
- *Crypto:sec:protocols*
- *Crypto:sec:signatures*
- *Crypto:sec:special*
- *Crypto:sec:symmetric*
- *Crypto:sec:symmetricB*
- Cryptographic Security Models
- Cryptography
- Cyber espionage in peacetime
- DSA, EC-DSA and Schnorr Signatures
- Data protection
- Dematerialisation of documents and electronic trust services
- Distinguishing criminal and civil law
- Effect of contract on non-contracting parties
- Electronic signatures and identity trust services
- Encouraging increased cyber security for products and services
- Encouraging security standards via contract
- Enforcement and penalties
- Enforcement jurisdiction
- Enforcement of privacy laws – penalties for violation
- Enforcement – remedies
- Ethics
- Exceptions to crimes against information systems
- Hard Problems
- Hash Functions
- Implementation Aspects
- Industry-specific regulations and NIS Directive
- Information-theoretically Secure Constructions
- Intellectual property
- Interception by a state
- Interception by persons other than states
- International data transfer
- International norms: foundations from international human rights law
- International treatment and conflict of law
- Internet intermediaries - shields from liability and take-down procedures
- Introductory principles of law and legal research
- Investigation and prevention of crime, and similar activities
- Jurisdiction
- KEM-DEM Philosophy
- Key Agreement Protocols
- Key Derivation and Extendable Output Functions
- Lattice-based Constructions
- Law and Regulation
- Limitations of liability and exclusions of liability
- Limiting the scope of liability: legal causation
- Main Page
- Mathematics
- Matters classified as secret by a state
- Merkle-Trees and Blockchains
- Message Authentication Codes
- Models to Understand Malicious Operations
- Modes of Operation
- Negligence
- Obligations owed to a client
- Oblivious Transfer
- One-Time Pad
- Online contracts: time of contract and receipt of contractual communication
- Other regulatory matters
- Personal data breach notification
- Prescriptive jurisdiction
- Privacy laws in general and electronic interception
- Private Information Retrieval and ORAM
- Public Key Encryption
- Public Key Encryption/Signatures With Special Properties
- Public Key Signatures
- Public international law
- Quantum of liability
- RSA-PSS
- Requirements of form and the threat of unenforceability
- Research and development activities conducted by non-state persons
- Restrictions on exporting security technologies
- Reverse engineering
- *Sec:ab-taxonomy*
- *Sec:crypto:MAC*
- Secret Sharing
- *Sect:elements*
- *Sect:models to understand malic ops*
- Secure Multi-Party Computation
- Self-help disfavoured: software locks and hack-back
- Setup Assumptions
- Simulation and UC Security
- Standard Protocols
- State cyber operations in general
- Stream Ciphers
- Strict liability for defective products
- Subject matter and regulatory focus
- Symmetric Encryption and Authentication
- Symmetric Primitives
- Syntax of Basic Schemes
- Territorial jurisdiction
- The Elements of a Malicious Operation
- The enforcement of and penalties for crimes against information systems
- The law of armed conflict
- The nature of evidence and proof
- The nature of law and legal analysis
- The problem of data sovereignty
- Tort
- Understanding intellectual property
- Vulnerability testing and disclosure
- Warranted state activity
- Warranties and their exclusion
- Zero-Knowledge

The segmentation process isolates 118 (sub)sections across three KAs and uploads them as MW pages. Page titles in *italics* are redirect pages, facilitating hyperlinked in-text references.

This document's other chapters discuss segmentation strategies, section titles and the structuring of MW pages into a logical, easy-to-navigate online resource, in detail.

CyBOK PDF



According to Clough [14], criminals have five main incentives to move their operations online:

1. Using the Internet, it is easier to find and contact victims. Email lists are sold on underground markets [15], while online social networks have search functionalities embedded in them, allowing criminals to easily identify potential victims [16, 17].
2. By using the Internet, criminal operations can be run more cheaply. Sending emails is free, while scammers previously had to pay postage to reach their victims. This also allows criminals to increase the scale of their operations to sizes that were previously unthinkable.
3. Compared to their physical counterparts, the Internet allows crimes to be performed faster. For example, emails can reach victims in a matter of seconds, without having to wait for physical letters to be delivered.
4. Using the Internet, it is easier to operate across international boundaries, reaching victims located in other countries. In this setting, often the only limitation is language, with criminals only targeting victims who speak a language that they are familiar with (e.g., people in English-speaking countries) [18].
5. By operating over the Internet, it is more difficult for criminals to get caught. This is mainly due to the transnational nature of cybercrime, and the fact that the problem of harmonising the appropriate laws of different countries is far from being solved [19]. In addition, research shows that online crime is often under reported, both because victims do not know whom to report it to (given that the offender might be located in another country), as well as the fact that they believe that they are unlikely to get their money back [20].

Cyber-dependent crimes, on the other hand, are crimes that can only be committed with the use of computers or technology devices [1]. Although the final goal of this type of crime often has parallels in the physical world (e.g., extortion, identity theft, financial fraud), the Internet and technology generally enable criminals to give a new shape to these crimes, making them large-scale organised endeavours able to reach hundreds of thousands, if not millions, of victims.

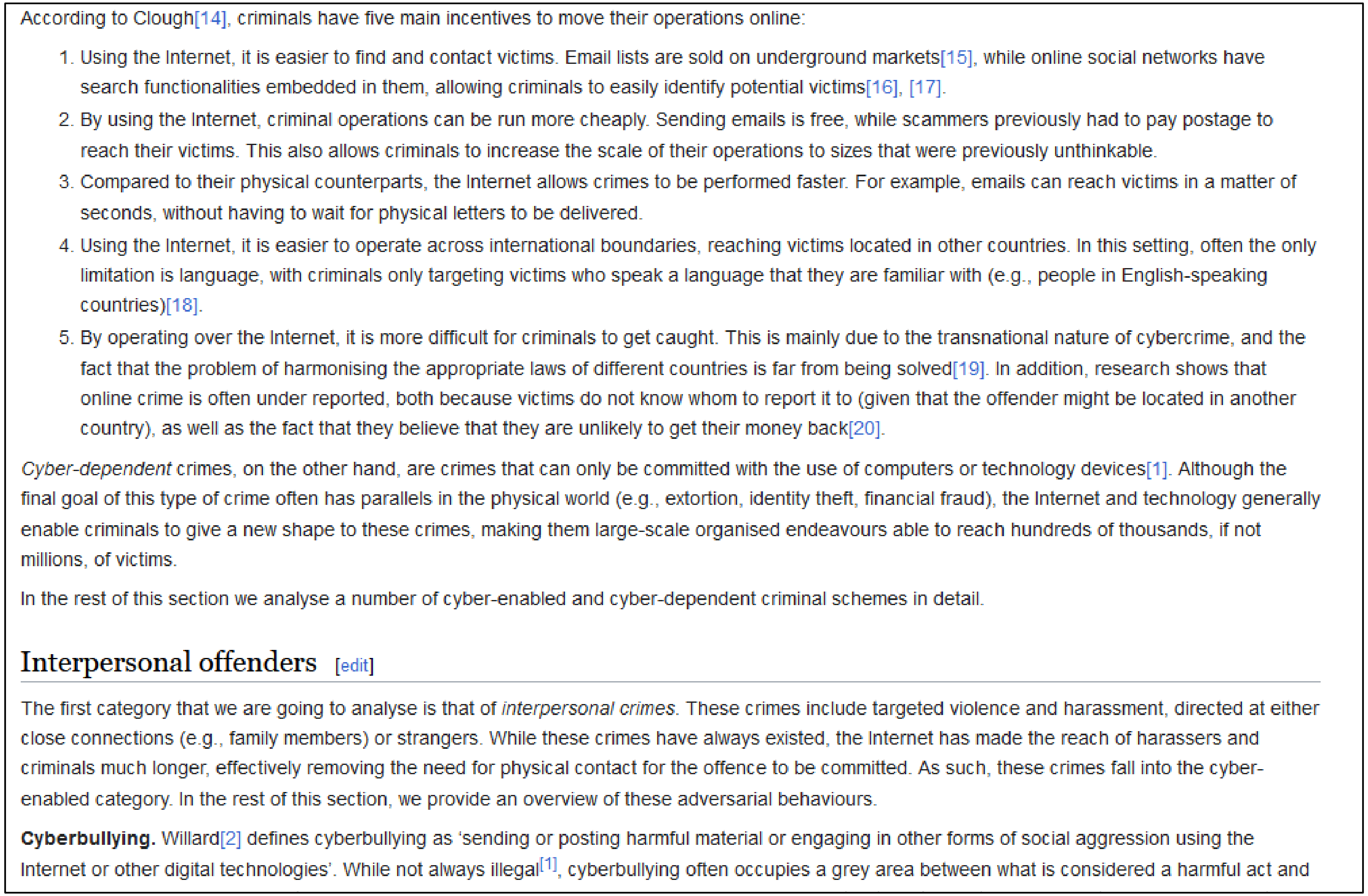
In the rest of this section we analyse a number of cyber-enabled and cyber-dependent criminal schemes in detail.

Interpersonal offenders

The first category that we are going to analyse is that of *interpersonal crimes*. These crimes include targeted violence and harassment, directed at either close connections (e.g., family members) or strangers. While these crimes have always existed, the Internet has made the reach of harassers and criminals much longer, effectively removing the need for physical contact for the offence to be committed. As such, these crimes fall into the cyber-enabled category. In the rest of this section, we provide an overview of these adversarial behaviours.

Cyberbullying. Willard [2] defines cyberbullying as 'sending or posting harmful material or engaging in other forms of social aggression using the Internet or other digital technologies'.

CyBOK Wiki



According to Clough[14], criminals have five main incentives to move their operations online:

1. Using the Internet, it is easier to find and contact victims. Email lists are sold on underground markets[15], while online social networks have search functionalities embedded in them, allowing criminals to easily identify potential victims[16], [17].
2. By using the Internet, criminal operations can be run more cheaply. Sending emails is free, while scammers previously had to pay postage to reach their victims. This also allows criminals to increase the scale of their operations to sizes that were previously unthinkable.
3. Compared to their physical counterparts, the Internet allows crimes to be performed faster. For example, emails can reach victims in a matter of seconds, without having to wait for physical letters to be delivered.
4. Using the Internet, it is easier to operate across international boundaries, reaching victims located in other countries. In this setting, often the only limitation is language, with criminals only targeting victims who speak a language that they are familiar with (e.g., people in English-speaking countries)[18].
5. By operating over the Internet, it is more difficult for criminals to get caught. This is mainly due to the transnational nature of cybercrime, and the fact that the problem of harmonising the appropriate laws of different countries is far from being solved[19]. In addition, research shows that online crime is often under reported, both because victims do not know whom to report it to (given that the offender might be located in another country), as well as the fact that they believe that they are unlikely to get their money back[20].

Cyber-dependent crimes, on the other hand, are crimes that can only be committed with the use of computers or technology devices[1]. Although the final goal of this type of crime often has parallels in the physical world (e.g., extortion, identity theft, financial fraud), the Internet and technology generally enable criminals to give a new shape to these crimes, making them large-scale organised endeavours able to reach hundreds of thousands, if not millions, of victims.

In the rest of this section we analyse a number of cyber-enabled and cyber-dependent criminal schemes in detail.

Interpersonal offenders [edit]

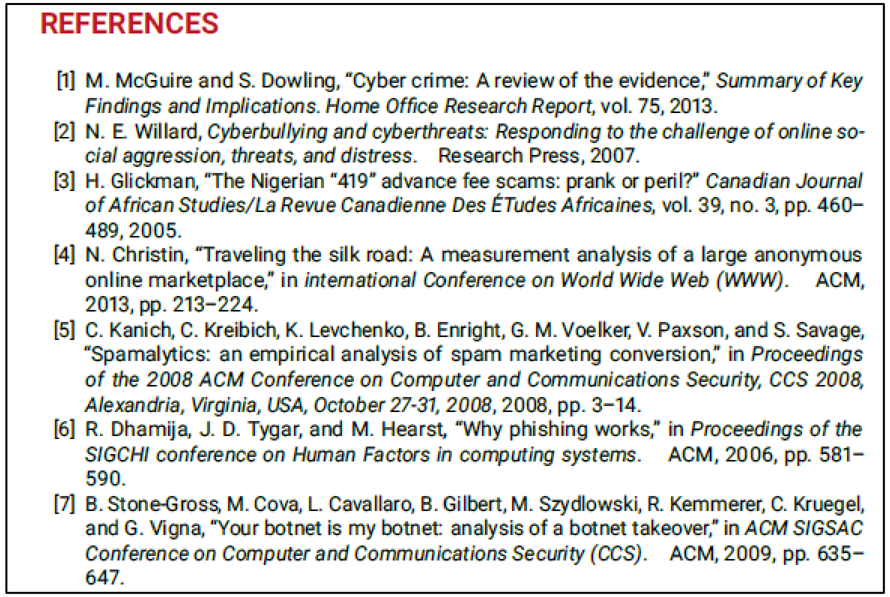
The first category that we are going to analyse is that of *interpersonal crimes*. These crimes include targeted violence and harassment, directed at either close connections (e.g., family members) or strangers. While these crimes have always existed, the Internet has made the reach of harassers and criminals much longer, effectively removing the need for physical contact for the offence to be committed. As such, these crimes fall into the cyber-enabled category. In the rest of this section, we provide an overview of these adversarial behaviours.

Cyberbullying. Willard[2] defines cyberbullying as 'sending or posting harmful material or engaging in other forms of social aggression using the Internet or other digital technologies'. While not always illegal[1], cyberbullying often occupies a grey area between what is considered a harmful act and

Basic text formatting is converted, e.g.: **bold**, *italics*, bulletpoints, paragraph breaks, headings.

In-line citations [1] are converted as hyperlinks, pointing at the bibliography at the end of each page.

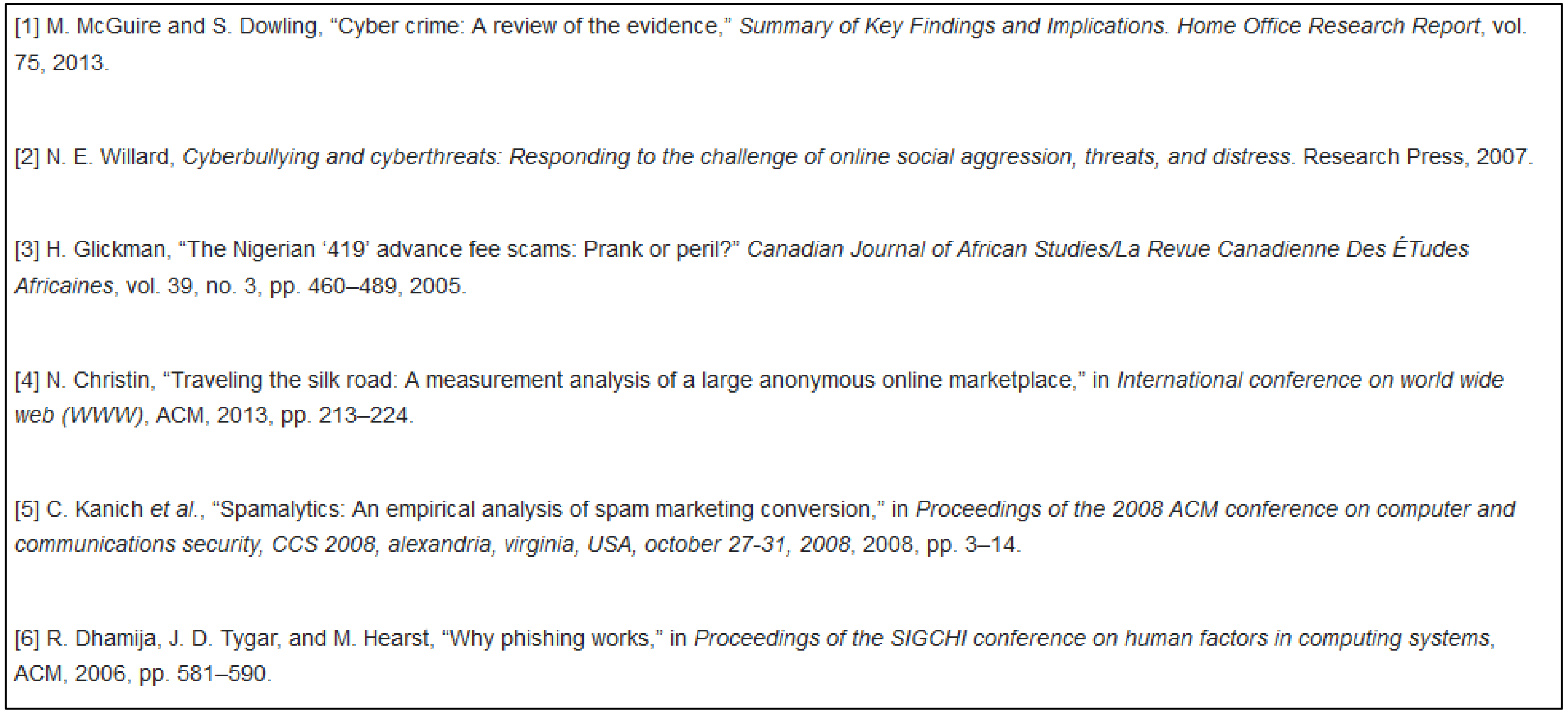
CyBOK PDF



REFERENCES

[1] M. McGuire and S. Dowling, "Cyber crime: A review of the evidence," *Summary of Key Findings and Implications. Home Office Research Report*, vol. 75, 2013.
[2] N. E. Willard, *Cyberbullying and cyberthreats: Responding to the challenge of online social aggression, threats, and distress.* Research Press, 2007.
[3] H. Glickman, "The Nigerian "419" advance fee scams: prank or peril?" *Canadian Journal of African Studies/La Revue Canadienne Des ÉTudes Africaines*, vol. 39, no. 3, pp. 460–489, 2005.
[4] N. Christin, "Traveling the silk road: A measurement analysis of a large anonymous online marketplace," in *international Conference on World Wide Web (WWW).* ACM, 2013, pp. 213–224.
[5] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage, "Spamalytics: an empirical analysis of spam marketing conversion," in *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008*, 2008, pp. 3–14.
[6] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI conference on Human Factors in computing systems.* ACM, 2006, pp. 581–590.
[7] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeover," in *ACM SIGSAC Conference on Computer and Communications Security (CCS).* ACM, 2009, pp. 635–647.

CyBOK Wiki (as implemented)



[1] M. McGuire and S. Dowling, "Cyber crime: A review of the evidence," *Summary of Key Findings and Implications. Home Office Research Report*, vol. 75, 2013.

[2] N. E. Willard, *Cyberbullying and cyberthreats: Responding to the challenge of online social aggression, threats, and distress*. Research Press, 2007.

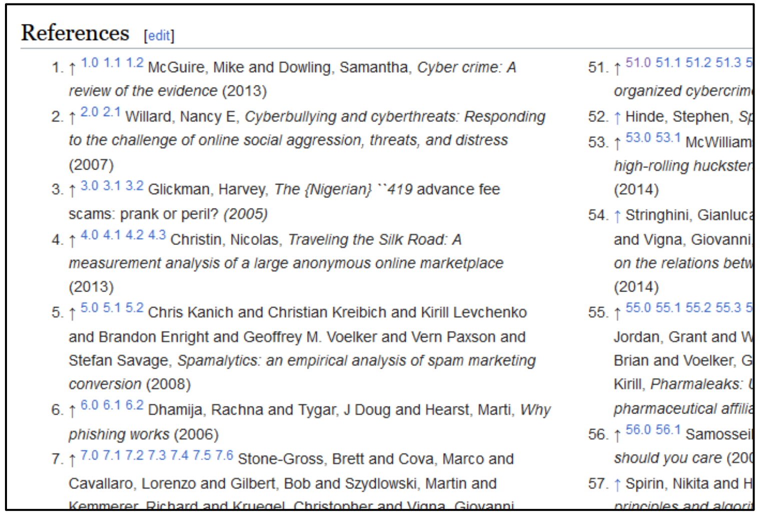
[3] H. Glickman, "The Nigerian '419' advance fee scams: Prank or peril?" *Canadian Journal of African Studies/La Revue Canadienne Des ÉTudes Africaines*, vol. 39, no. 3, pp. 460–489, 2005.

[4] N. Christin, "Traveling the silk road: A measurement analysis of a large anonymous online marketplace," in *International conference on world wide web (WWW)*, ACM, 2013, pp. 213–224.

[5] C. Kanich *et al.*, "Spamalytics: An empirical analysis of spam marketing conversion," in *Proceedings of the 2008 ACM conference on computer and communications security, CCS 2008, alexandria, virginia, USA, october 27-31, 2008*, 2008, pp. 3–14.

[6] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI conference on human factors in computing systems*, ACM, 2006, pp. 581–590.

CyBOK Wiki to-be (with native MW citation rendering)



References [edit]

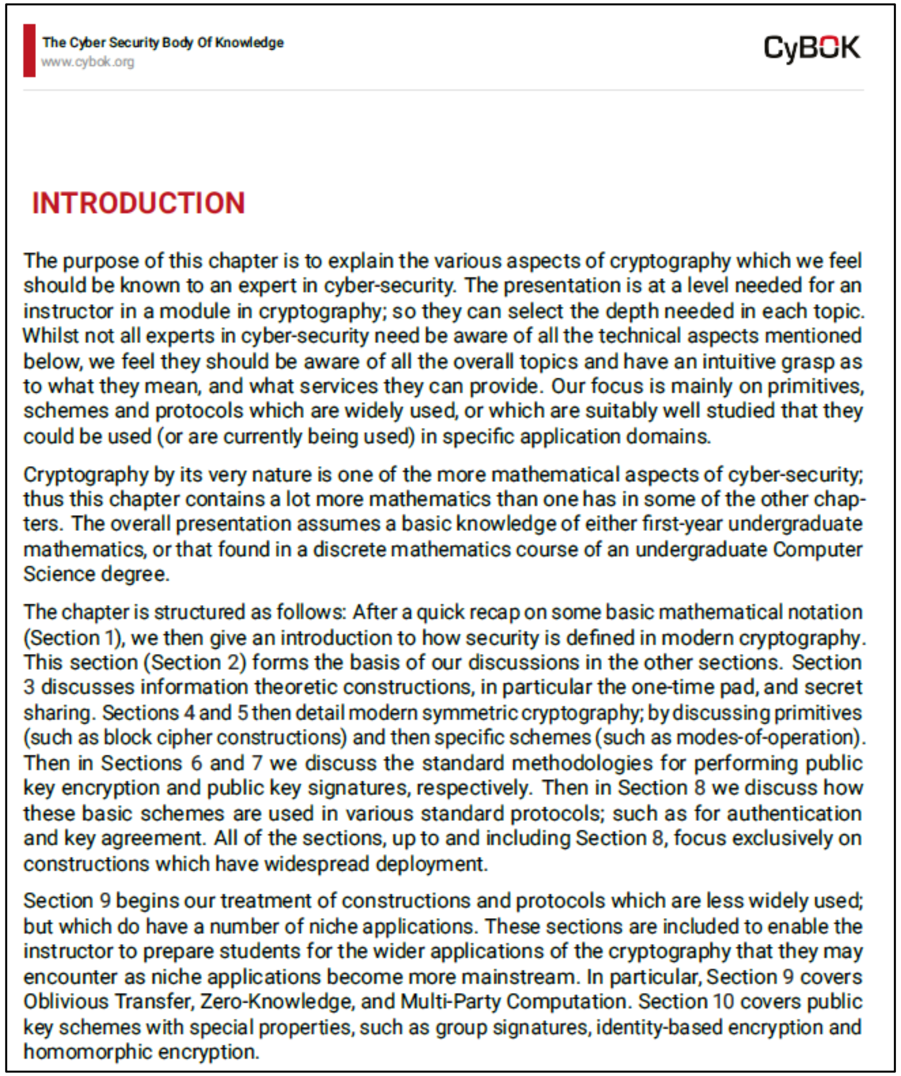
1. ↑ 1.0 1.1 1.2 McGuire, Mike and Dowling, Samantha, *Cyber crime: A review of the evidence* (2013)
2. ↑ 2.0 2.1 Willard, Nancy E, *Cyberbullying and cyberthreats: Responding to the challenge of online social aggression, threats, and distress* (2007)
3. ↑ 3.0 3.1 3.2 Glickman, Harvey, *The {Nigerian} ``419* advance fee scams: prank or peril? *(2005)*
4. ↑ 4.0 4.1 4.2 4.3 Christin, Nicolas, *Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace* (2013)
5. ↑ 5.0 5.1 5.2 Chris Kanich and Christian Kreibich and Kirill Levchenko and Brandon Enright and Geoffrey M. Voelker and Vern Paxson and Stefan Savage, *Spamalytics: an empirical analysis of spam marketing conversion* (2008)
6. ↑ 6.0 6.1 6.2 Dhamija, Rachna and Tygar, J Doug and Hearst, Marti, *Why phishing works* (2006)
7. ↑ 7.0 7.1 7.2 7.3 7.4 7.5 7.6 Stone-Gross, Brett and Cova, Marco and Cavallaro, Lorenzo and Gilbert, Bob and Szydlowski, Martin and Kemmerer, Richard and Kruegel, Christopher and Vigna, Giovanni

51. ↑ 51.0 51.1 51.2 51.3 5 *organized cybercrim*
52. ↑ Hinde, Stephen, *Sp*
53. ↑ 53.0 53.1 McWilliam *high-rolling huckster* (2014)
54. ↑ Stringhini, Gianluca and Vigna, Giovanni, *on the relations betw* (2014)
55. ↑ 55.0 55.1 55.2 55.3 5 Jordan, Grant and W Brian and Voelker, G Kirill, *Pharmaleaks: pharmaceutical affilia*
56. ↑ 56.0 56.1 Samossei *should you care* (200
57. ↑ Spirin, Nikita and H *principles and algori*

Bibliographies are generated automatically at the end of each page, with *IEEE* style. Only those bibliographic elements are included that have references to on the page at hand, so bibliographies on each page are shorter than the several hundred strong bibliographies at the end of CyBOK PDF KAs.

With proper customization (custom Lua filter and python preprocessor, see to-dos) native WM-style citation can be implemented, which look more beautiful and offer more functionality. A proof-of-concept of this has been attempted (not implemented in the main functionality and not utilizing Pandoc – this must be reworked).

CyBOK PDF


The Cyber Security Body Of Knowledge
www.cybok.org

CyBOK

INTRODUCTION

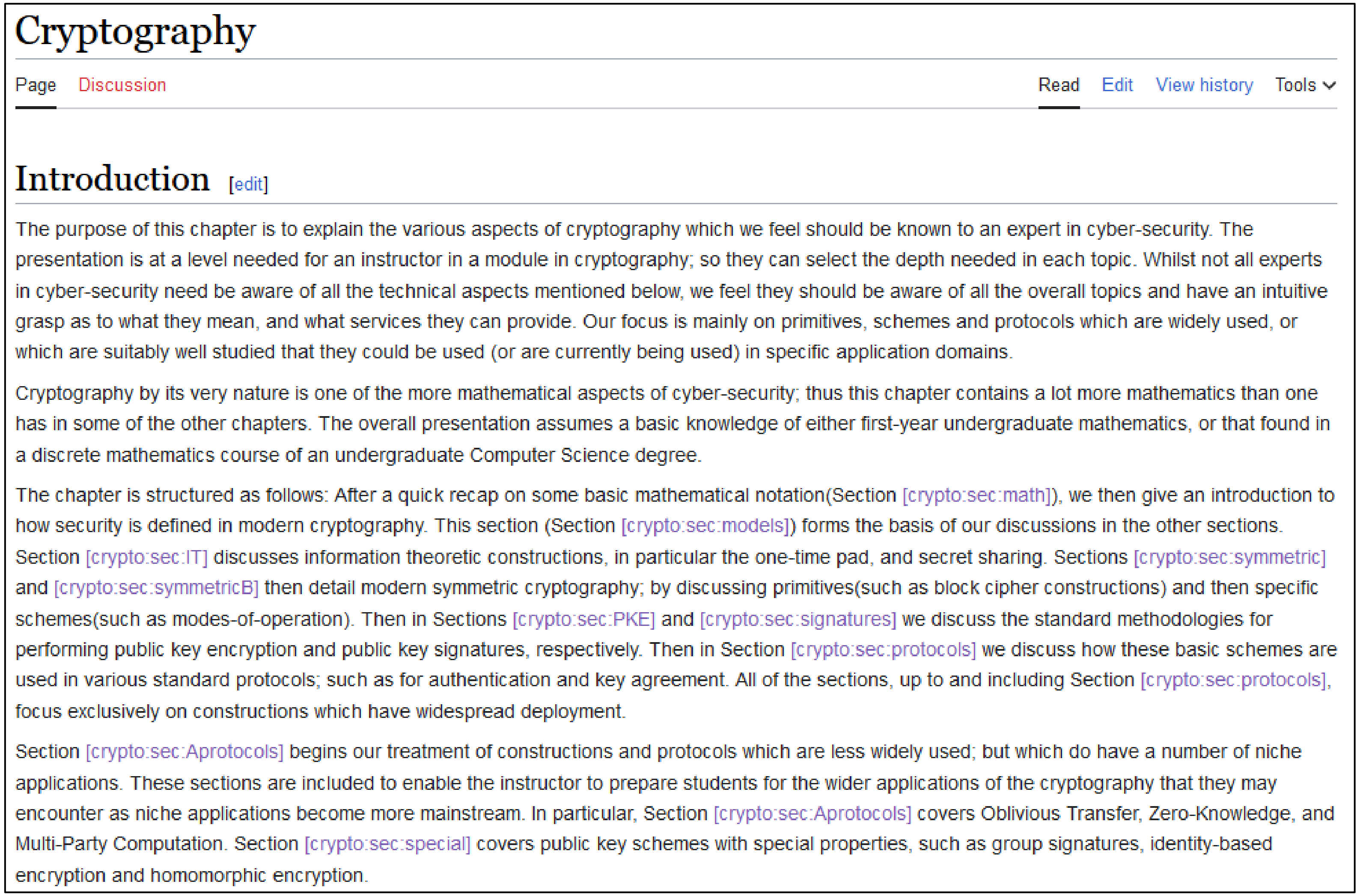
The purpose of this chapter is to explain the various aspects of cryptography which we feel should be known to an expert in cyber-security. The presentation is at a level needed for an instructor in a module in cryptography; so they can select the depth needed in each topic. Whilst not all experts in cyber-security need be aware of all the technical aspects mentioned below, we feel they should be aware of all the overall topics and have an intuitive grasp as to what they mean, and what services they can provide. Our focus is mainly on primitives, schemes and protocols which are widely used, or which are suitably well studied that they could be used (or are currently being used) in specific application domains.

Cryptography by its very nature is one of the more mathematical aspects of cyber-security; thus this chapter contains a lot more mathematics than one has in some of the other chapters. The overall presentation assumes a basic knowledge of either first-year undergraduate mathematics, or that found in a discrete mathematics course of an undergraduate Computer Science degree.

The chapter is structured as follows: After a quick recap on some basic mathematical notation (Section 1), we then give an introduction to how security is defined in modern cryptography. This section (Section 2) forms the basis of our discussions in the other sections. Section 3 discusses information theoretic constructions, in particular the one-time pad, and secret sharing. Sections 4 and 5 then detail modern symmetric cryptography; by discussing primitives (such as block cipher constructions) and then specific schemes (such as modes-of-operation). Then in Sections 6 and 7 we discuss the standard methodologies for performing public key encryption and public key signatures, respectively. Then in Section 8 we discuss how these basic schemes are used in various standard protocols; such as for authentication and key agreement. All of the sections, up to and including Section 8, focus exclusively on constructions which have widespread deployment.

Section 9 begins our treatment of constructions and protocols which are less widely used; but which do have a number of niche applications. These sections are included to enable the instructor to prepare students for the wider applications of the cryptography that they may encounter as niche applications become more mainstream. In particular, Section 9 covers Oblivious Transfer, Zero-Knowledge, and Multi-Party Computation. Section 10 covers public key schemes with special properties, such as group signatures, identity-based encryption and homomorphic encryption.

CyBOK Wiki



Cryptography

Introduction [edit]

The purpose of this chapter is to explain the various aspects of cryptography which we feel should be known to an expert in cyber-security. The presentation is at a level needed for an instructor in a module in cryptography; so they can select the depth needed in each topic. Whilst not all experts in cyber-security need be aware of all the technical aspects mentioned below, we feel they should be aware of all the overall topics and have an intuitive grasp as to what they mean, and what services they can provide. Our focus is mainly on primitives, schemes and protocols which are widely used, or which are suitably well studied that they could be used (or are currently being used) in specific application domains.

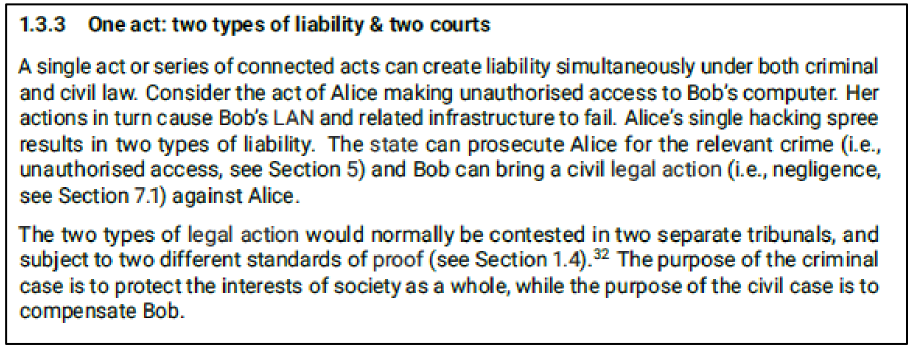
Cryptography by its very nature is one of the more mathematical aspects of cyber-security; thus this chapter contains a lot more mathematics than one has in some of the other chapters. The overall presentation assumes a basic knowledge of either first-year undergraduate mathematics, or that found in a discrete mathematics course of an undergraduate Computer Science degree.

The chapter is structured as follows: After a quick recap on some basic mathematical notation(Section [crypto:sec:math]), we then give an introduction to how security is defined in modern cryptography. This section (Section [crypto:sec:models]) forms the basis of our discussions in the other sections. Section [crypto:sec:IT] discusses information theoretic constructions, in particular the one-time pad, and secret sharing. Sections [crypto:sec:symmetric] and [crypto:sec:symmetricB] then detail modern symmetric cryptography; by discussing primitives(such as block cipher constructions) and then specific schemes(such as modes-of-operation). Then in Sections [crypto:sec:PKE] and [crypto:sec:signatures] we discuss the standard methodologies for performing public key encryption and public key signatures, respectively. Then in Section [crypto:sec:protocols] we discuss how these basic schemes are used in various standard protocols; such as for authentication and key agreement. All of the sections, up to and including Section [crypto:sec:protocols], focus exclusively on constructions which have widespread deployment.

Section [crypto:sec:Aprotocols] begins our treatment of constructions and protocols which are less widely used; but which do have a number of niche applications. These sections are included to enable the instructor to prepare students for the wider applications of the cryptography that they may encounter as niche applications become more mainstream. In particular, Section [crypto:sec:Aprotocols] covers Oblivious Transfer, Zero-Knowledge, and Multi-Party Computation. Section [crypto:sec:special] covers public key schemes with special properties, such as group signatures, identity-based encryption and homomorphic encryption.

Links to other (sub)sections are turned into hyperlinks to their respective pages. Hyperlink texts are the LaTeX `\label{}` values of the linked (sub)sections – this can be changed, as described in detail in chapter "*How to display section titles?*"

CyBOK PDF

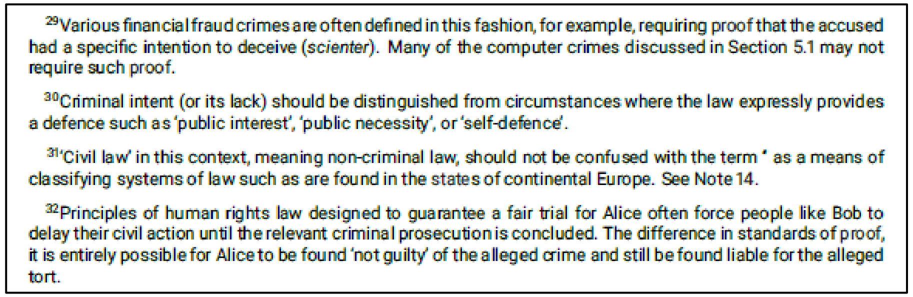


1.3.3 One act: two types of liability & two courts

A single act or series of connected acts can create liability simultaneously under both criminal and civil law. Consider the act of Alice making unauthorised access to Bob's computer. Her actions in turn cause Bob's LAN and related infrastructure to fail. Alice's single hacking spree results in two types of liability. The state can prosecute Alice for the relevant crime (i.e., unauthorised access, see Section 5) and Bob can bring a civil legal action (i.e., negligence, see Section 7.1) against Alice.

The two types of legal action would normally be contested in two separate tribunals, and subject to two different standards of proof (see Section 1.4).[32] The purpose of the criminal case is to protect the interests of society as a whole, while the purpose of the civil case is to compensate Bob.

...



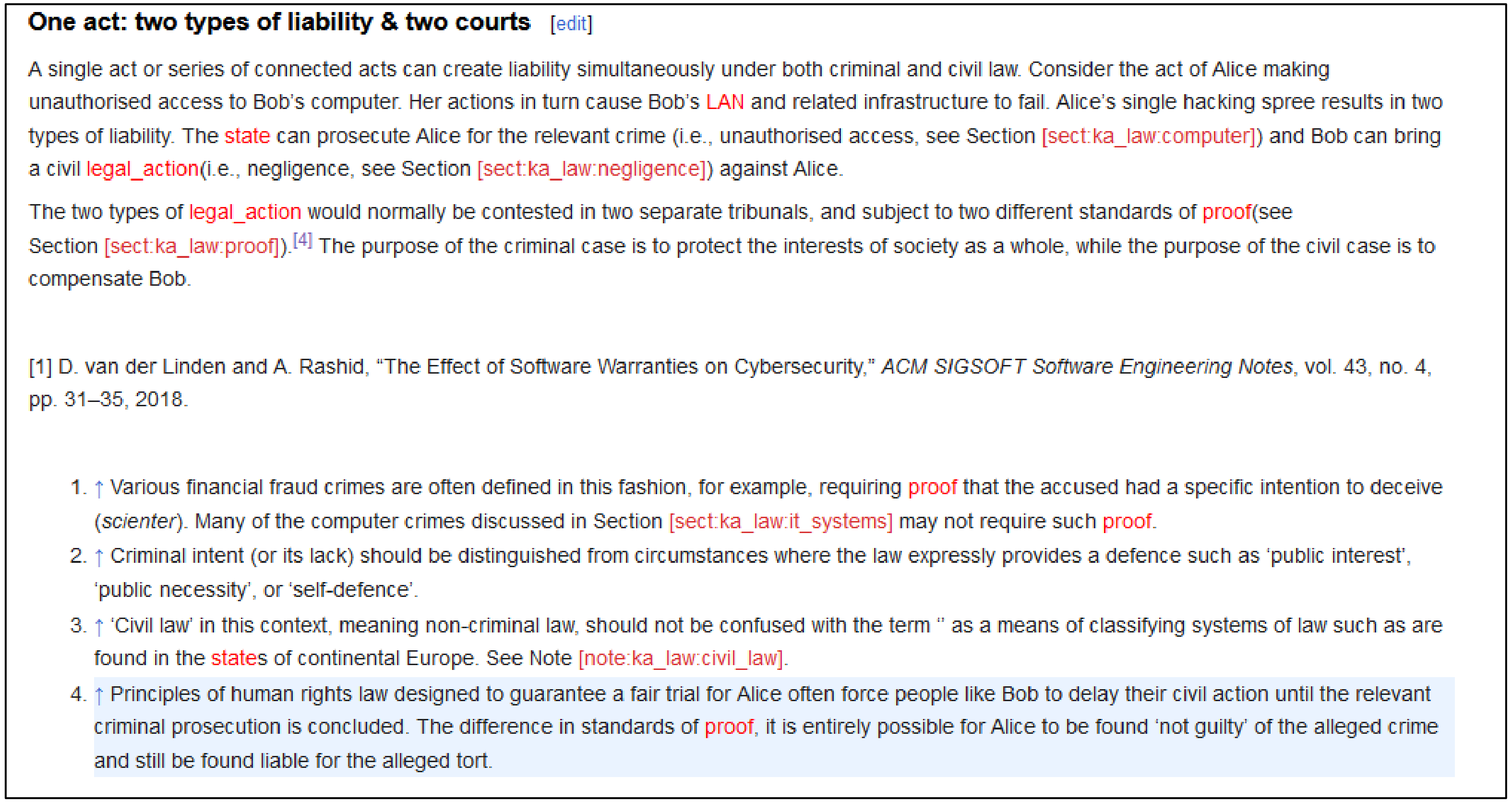
[29] Various financial fraud crimes are often defined in this fashion, for example, requiring proof that the accused had a specific intention to deceive (*scienter*). Many of the computer crimes discussed in Section 5.1 may not require such proof.

[30] Criminal intent (or its lack) should be distinguished from circumstances where the law expressly provides a defence such as 'public interest', 'public necessity', or 'self-defence'.

[31] 'Civil law' in this context, meaning non-criminal law, should not be confused with the term ' as a means of classifying systems of law such as are found in the states of continental Europe. See Note 14.

[32] Principles of human rights law designed to guarantee a fair trial for Alice often force people like Bob to delay their civil action until the relevant criminal prosecution is concluded. The difference in standards of proof, it is entirely possible for Alice to be found 'not guilty' of the alleged crime and still be found liable for the alleged tort.

CyBOK Wiki



One act: two types of liability & two courts [edit]

A single act or series of connected acts can create liability simultaneously under both criminal and civil law. Consider the act of Alice making unauthorised access to Bob's computer. Her actions in turn cause Bob's LAN and related infrastructure to fail. Alice's single hacking spree results in two types of liability. The state can prosecute Alice for the relevant crime (i.e., unauthorised access, see Section [sect:ka_law:computer]) and Bob can bring a civil legal_action(i.e., negligence, see Section [sect:ka_law:negligence]) against Alice.

The two types of legal_action would normally be contested in two separate tribunals, and subject to two different standards of proof(see Section [sect:ka_law:proof]).[4] The purpose of the criminal case is to protect the interests of society as a whole, while the purpose of the civil case is to compensate Bob.

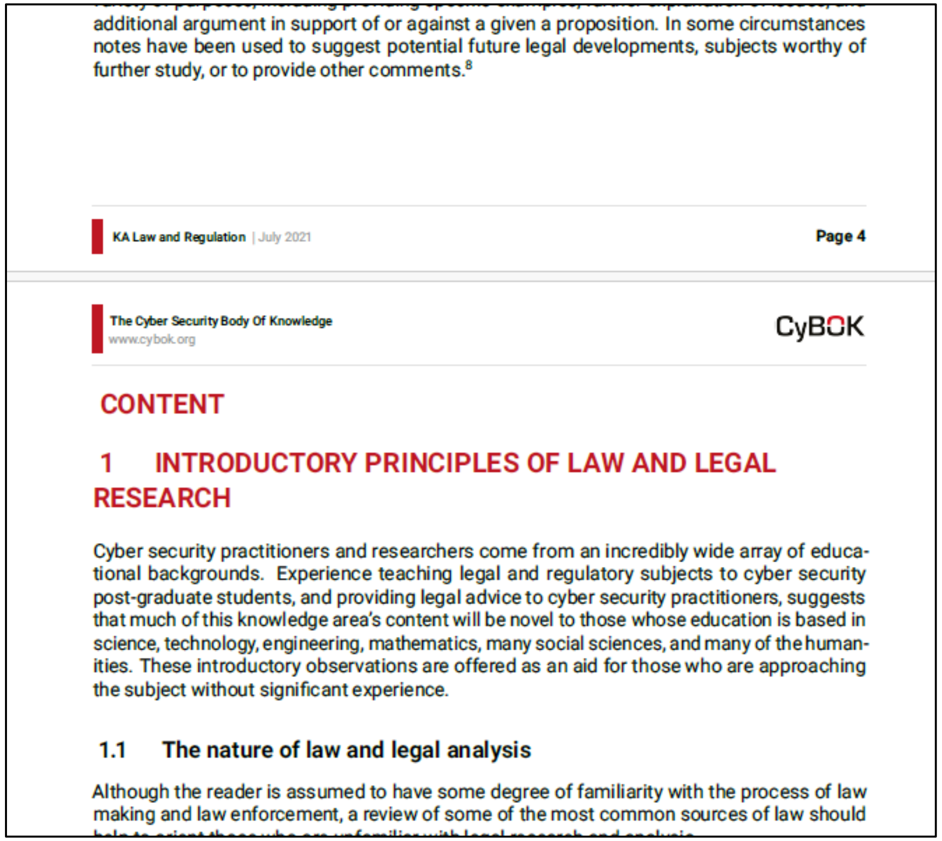
[1] D. van der Linden and A. Rashid, "The Effect of Software Warranties on Cybersecurity," *ACM SIGSOFT Software Engineering Notes*, vol. 43, no. 4, pp. 31–35, 2018.

1. ↑ Various financial fraud crimes are often defined in this fashion, for example, requiring proof that the accused had a specific intention to deceive (*scienter*). Many of the computer crimes discussed in Section [sect:ka_law:it_systems] may not require such proof.
2. ↑ Criminal intent (or its lack) should be distinguished from circumstances where the law expressly provides a defence such as 'public interest', 'public necessity', or 'self-defence'.
3. ↑ 'Civil law' in this context, meaning non-criminal law, should not be confused with the term '' as a means of classifying systems of law such as are found in the states of continental Europe. See Note [note:ka_law:civil_law].
4. ↑ Principles of human rights law designed to guarantee a fair trial for Alice often force people like Bob to delay their civil action until the relevant criminal prosecution is concluded. The difference in standards of proof, it is entirely possible for Alice to be found 'not guilty' of the alleged crime and still be found liable for the alleged tort.

Notes[1] are correctly referenced and linked in-text and get rendered in the native MW reference style at the end of each page (only the ones relevant to that page)

As a proof-of-concept to creating custom Lua filters, acronyms and glossary terms are highlighted in red. Their intended functionality is not implemented (or indeed decided, see chapter "*Indices, acronyms, glossary (and references)*")

CyBOK PDF chapter



additional argument in support of or against a given a proposition. In some circumstances notes have been used to suggest potential future legal developments, subjects worthy of further study, or to provide other comments.[8]

CONTENT

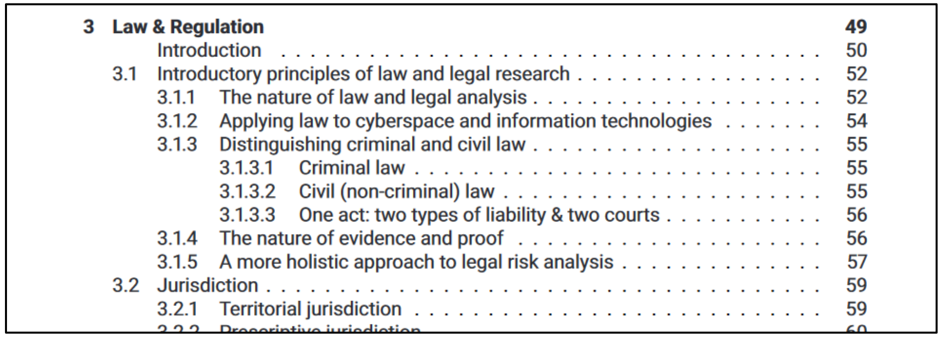
1 INTRODUCTORY PRINCIPLES OF LAW AND LEGAL RESEARCH

Cyber security practitioners and researchers come from an incredibly wide array of educational backgrounds. Experience teaching legal and regulatory subjects to cyber security post-graduate students, and providing legal advice to cyber security practitioners, suggests that much of this knowledge area's content will be novel to those whose education is based in science, technology, engineering, mathematics, many social sciences, and many of the humanities. These introductory observations are offered as an aid for those who are approaching the subject without significant experience.

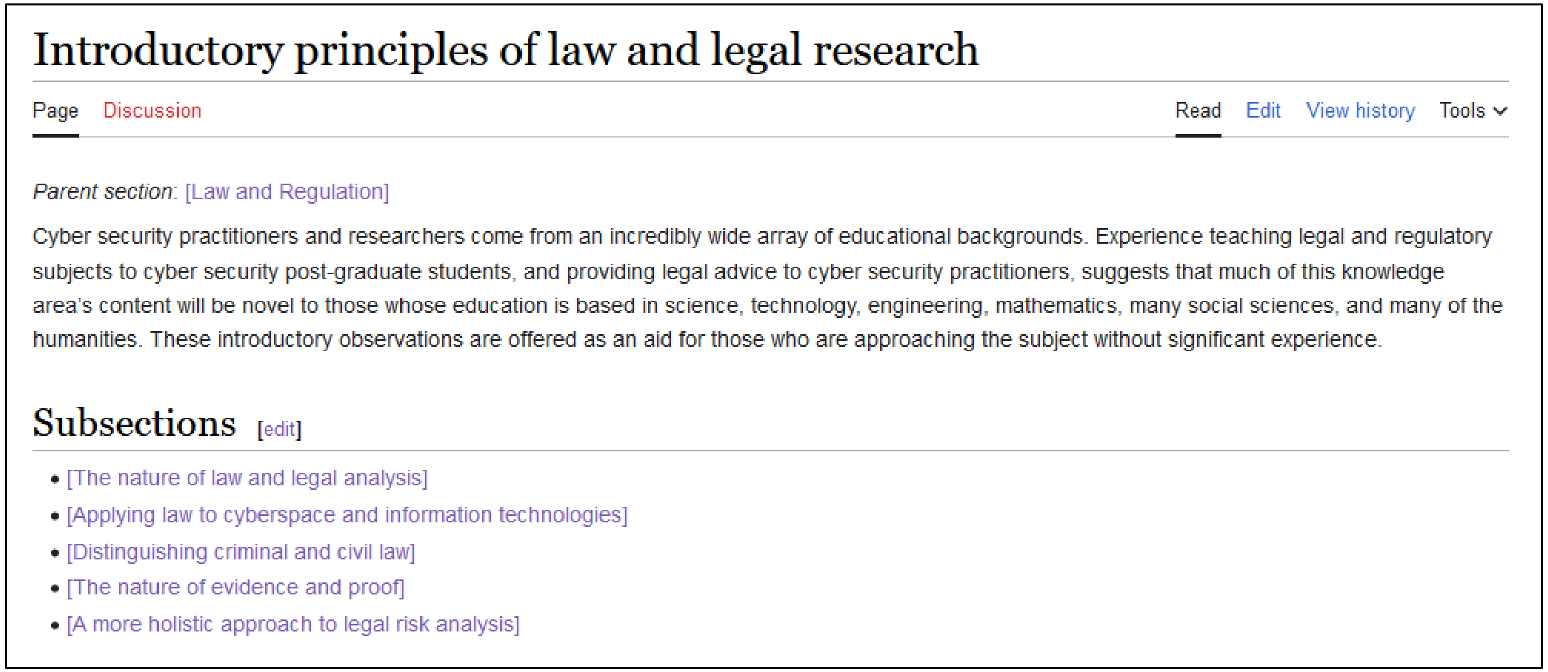
1.1 The nature of law and legal analysis

Although the reader is assumed to have some degree of familiarity with the process of law making and law enforcement, a review of some of the most common sources of law should

CyBOK PDF ToC



CyBOK Wiki



Introductory principles of law and legal research

Parent section: [Law and Regulation]

Cyber security practitioners and researchers come from an incredibly wide array of educational backgrounds. Experience teaching legal and regulatory subjects to cyber security post-graduate students, and providing legal advice to cyber security practitioners, suggests that much of this knowledge area's content will be novel to those whose education is based in science, technology, engineering, mathematics, many social sciences, and many of the humanities. These introductory observations are offered as an aid for those who are approaching the subject without significant experience.

Subsections [edit]

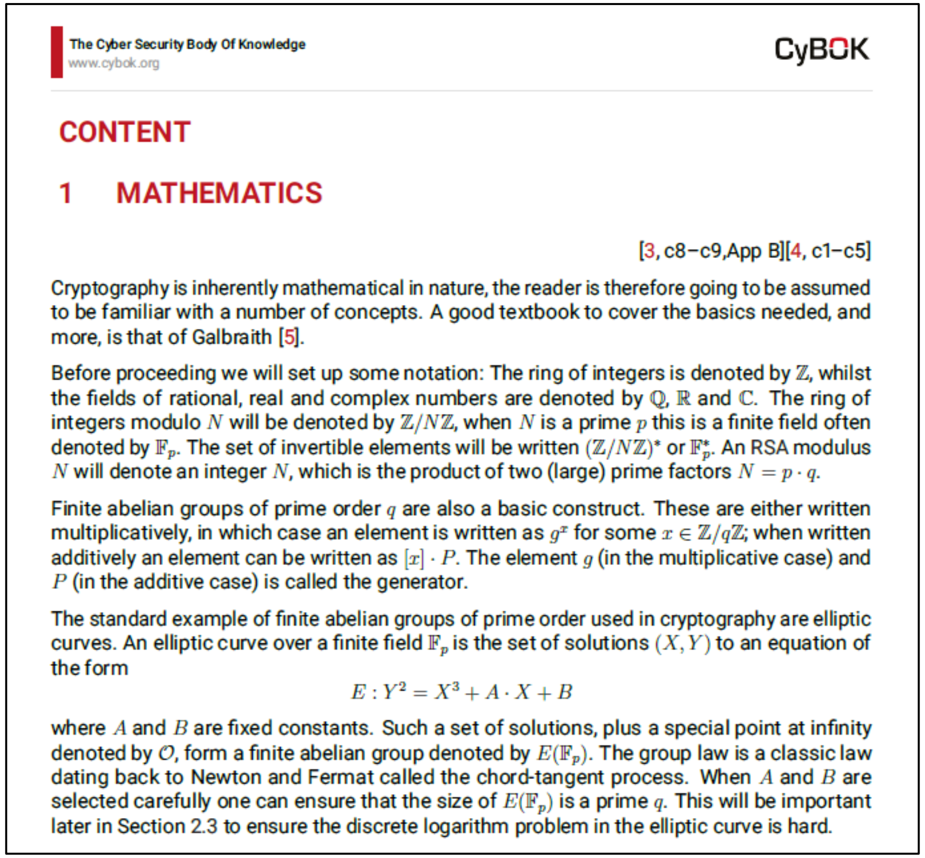
- [The nature of law and legal analysis]
- [Applying law to cyberspace and information technologies]
- [Distinguishing criminal and civil law]
- [The nature of evidence and proof]
- [A more holistic approach to legal risk analysis]

Hyperlinks to parent- and subsections are added to ease navigation across the book.

See chapter "*How to implement and display CyBOK's structure in MW pages?*" for detailed considerations on how to better implement navigation beyond the proof-of-concept.

CyBOK PDF

CyBOK Wiki



CONTENT

1 MATHEMATICS

[3, c8–c9,App B][4, c1–c5]

Cryptography is inherently mathematical in nature, the reader is therefore going to be assumed to be familiar with a number of concepts. A good textbook to cover the basics needed, and more, is that of Galbraith [5].

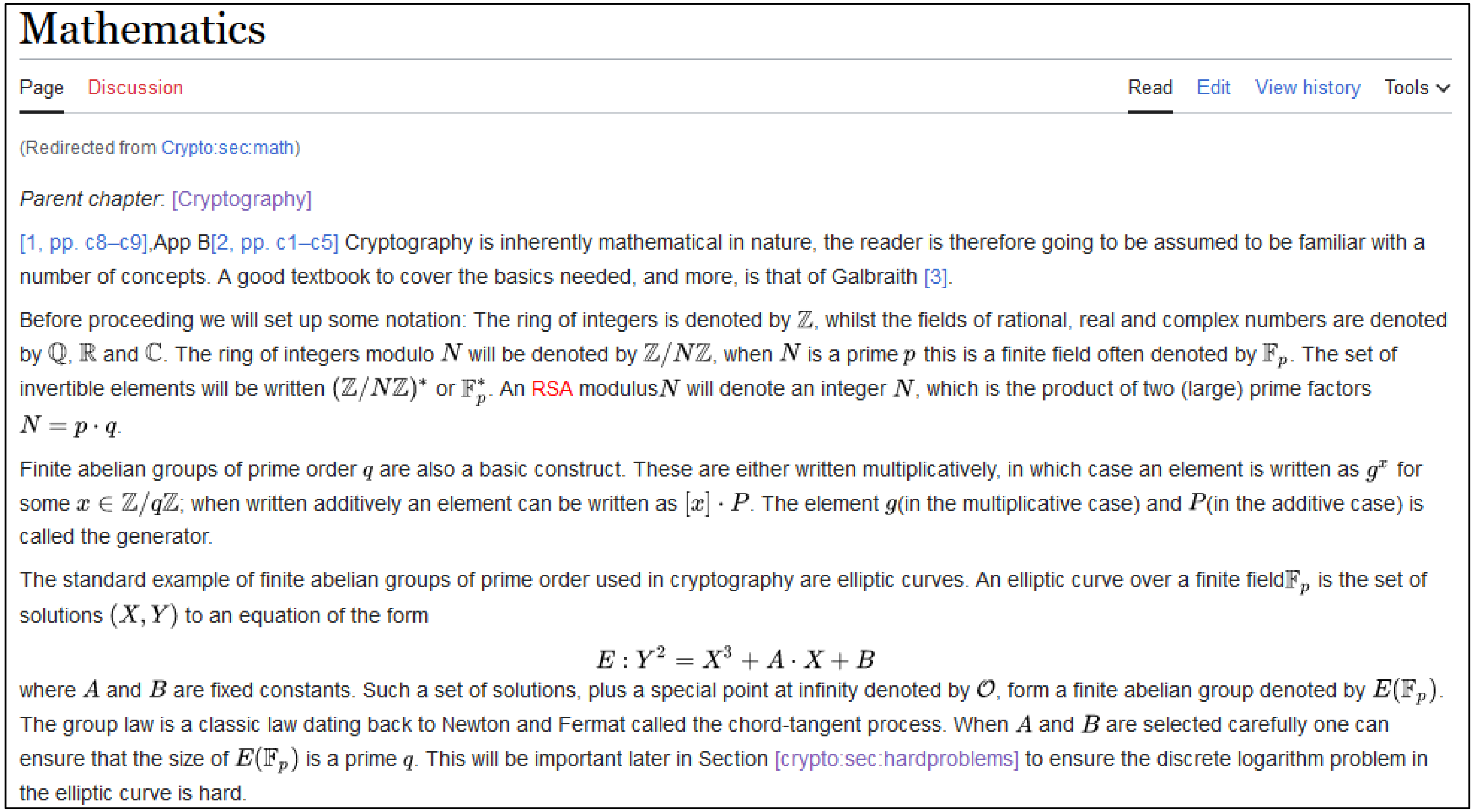
Before proceeding we will set up some notation: The ring of integers is denoted by $\mathbb{Z}$, whilst the fields of rational, real and complex numbers are denoted by $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$. The ring of integers modulo N will be denoted by $\mathbb{Z}/N\mathbb{Z}$, when N is a prime p this is a finite field often denoted by $\mathbb{F}_p$. The set of invertible elements will be written $(\mathbb{Z}/N\mathbb{Z})^*$ or $\mathbb{F}_p^*$. An RSA modulus N will denote an integer N, which is the product of two (large) prime factors $N = p \cdot q$.

Finite abelian groups of prime order q are also a basic construct. These are either written multiplicatively, in which case an element is written as g^x for some $x \in \mathbb{Z}/q\mathbb{Z}$; when written additively an element can be written as $[x] \cdot P$. The element g (in the multiplicative case) and P (in the additive case) is called the generator.

The standard example of finite abelian groups of prime order used in cryptography are elliptic curves. An elliptic curve over a finite field $\mathbb{F}_p$ is the set of solutions (X, Y) to an equation of the form

$$E : Y^2 = X^3 + A \cdot X + B$$

where A and B are fixed constants. Such a set of solutions, plus a special point at infinity denoted by $\mathcal{O}$, form a finite abelian group denoted by $E(\mathbb{F}_p)$. The group law is a classic law dating back to Newton and Fermat called the chord-tangent process. When A and B are selected carefully one can ensure that the size of $E(\mathbb{F}_p)$ is a prime q. This will be important later in Section 2.3 to ensure the discrete logarithm problem in the elliptic curve is hard.



Mathematics

(Redirected from Crypto:sec:math)

Parent chapter: [Cryptography]

[1, pp. c8–c9],App B[2, pp. c1–c5] Cryptography is inherently mathematical in nature, the reader is therefore going to be assumed to be familiar with a number of concepts. A good textbook to cover the basics needed, and more, is that of Galbraith [3].

Before proceeding we will set up some notation: The ring of integers is denoted by $\mathbb{Z}$, whilst the fields of rational, real and complex numbers are denoted by $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$. The ring of integers modulo N will be denoted by $\mathbb{Z}/N\mathbb{Z}$, when N is a prime p this is a finite field often denoted by $\mathbb{F}_p$. The set of invertible elements will be written $(\mathbb{Z}/N\mathbb{Z})^*$ or $\mathbb{F}_p^*$. An RSA modulus$N$ will denote an integer N, which is the product of two (large) prime factors $N = p \cdot q$.

Finite abelian groups of prime order q are also a basic construct. These are either written multiplicatively, in which case an element is written as g^x for some $x \in \mathbb{Z}/q\mathbb{Z}$; when written additively an element can be written as $[x] \cdot P$. The element g(in the multiplicative case) and P(in the additive case) is called the generator.

The standard example of finite abelian groups of prime order used in cryptography are elliptic curves. An elliptic curve over a finite field$\mathbb{F}_p$ is the set of solutions (X, Y) to an equation of the form

$$E : Y^2 = X^3 + A \cdot X + B$$

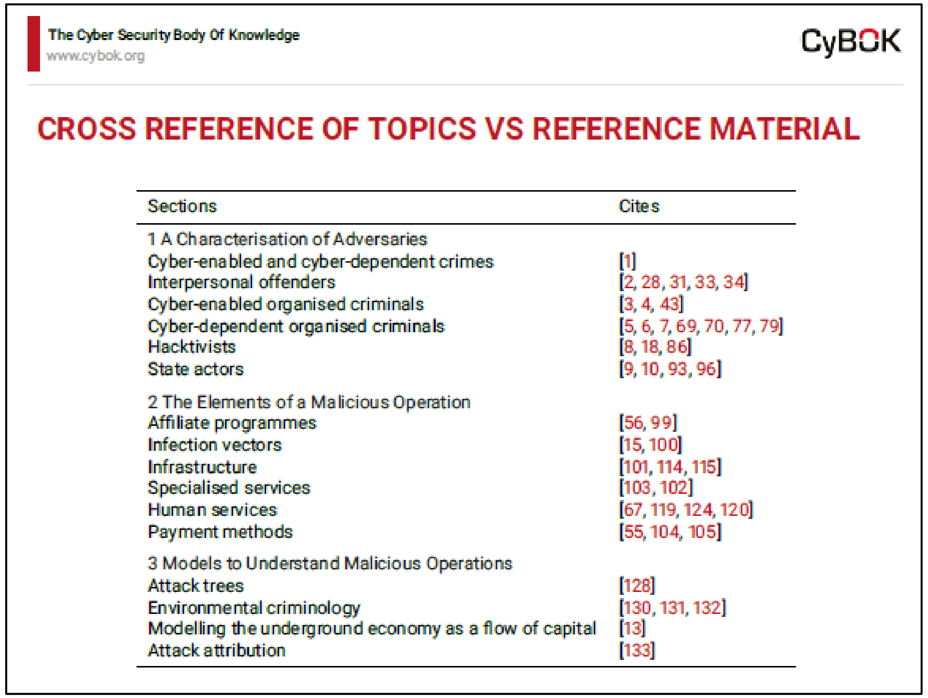
where A and B are fixed constants. Such a set of solutions, plus a special point at infinity denoted by $\mathcal{O}$, form a finite abelian group denoted by $E(\mathbb{F}_p)$. The group law is a classic law dating back to Newton and Fermat called the chord-tangent process. When A and B are selected carefully one can ensure that the size of $E(\mathbb{F}_p)$ is a prime q. This will be important later in Section [crypto:sec:hardproblems] to ensure the discrete logarithm problem in the elliptic curve is hard.

Math formulae are rendered just like in LaTeX, both in-line and as blocks.

Not everything gets converted to MW markup correctly. For example the chapter leading citations are not aligned right in a separate row, and some in-line math expressions are not properly separated from the word before or after them (for example: A modulus$N$ will de).

The chapter "*Considerations for implementation*" is largely about discussing different strategies to tackle conversion discrepancies beyond the proof-of-concept.

CyBOK PDF



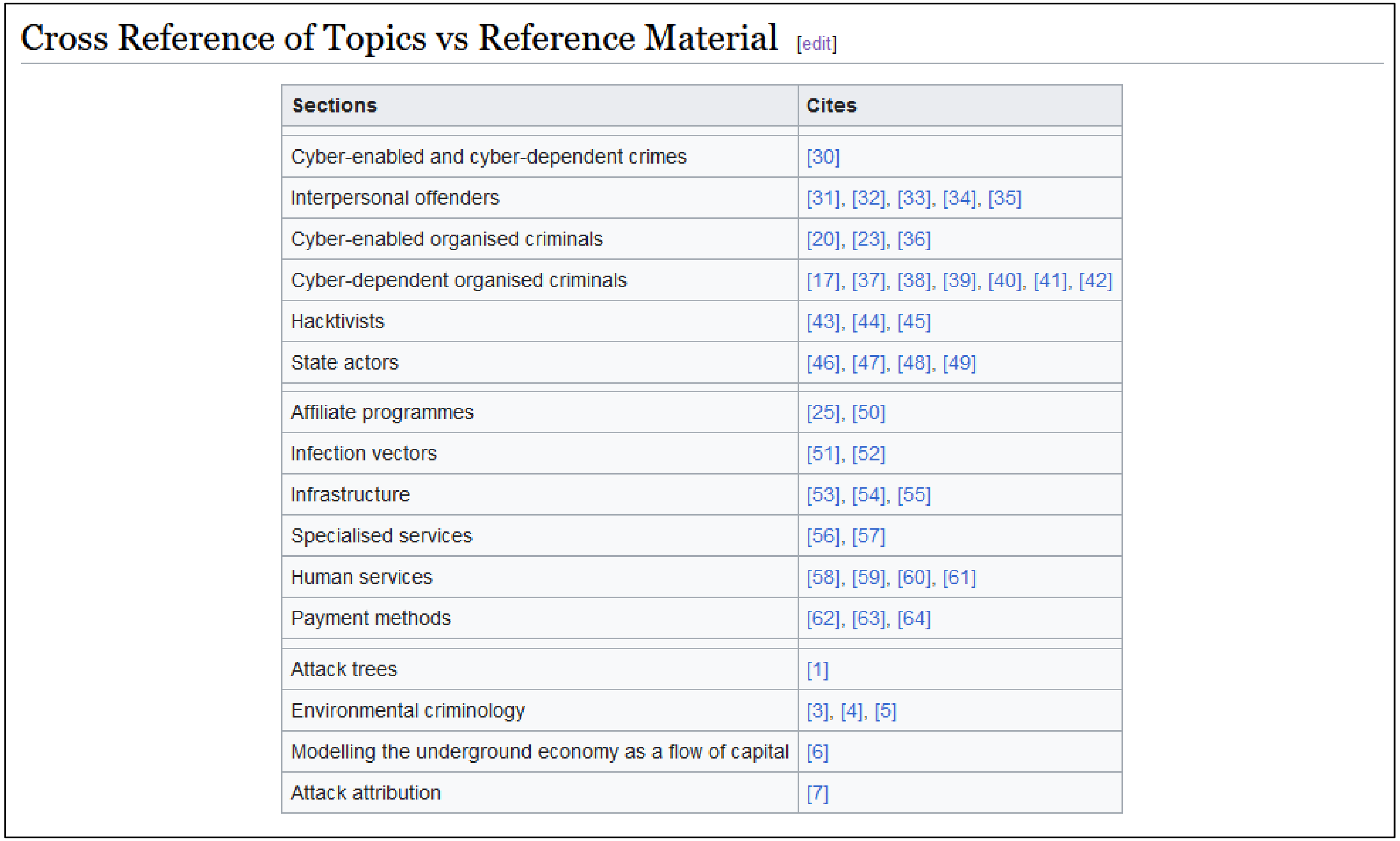
The Cyber Security Body Of Knowledge
www.cybok.org

CyBOK

CROSS REFERENCE OF TOPICS VS REFERENCE MATERIAL

Sections	Cites
1 A Characterisation of Adversaries	
Cyber-enabled and cyber-dependent crimes	[1]
Interpersonal offenders	[2, 28, 31, 33, 34]
Cyber-enabled organised criminals	[3, 4, 43]
Cyber-dependent organised criminals	[5, 6, 7, 69, 70, 77, 79]
Hacktivists	[8, 18, 86]
State actors	[9, 10, 93, 96]
2 The Elements of a Malicious Operation	
Affiliate programmes	[56, 99]
Infection vectors	[15, 100]
Infrastructure	[101, 114, 115]
Specialised services	[103, 102]
Human services	[67, 119, 124, 120]
Payment methods	[55, 104, 105]
3 Models to Understand Malicious Operations	
Attack trees	[128]
Environmental criminology	[130, 131, 132]
Modelling the underground economy as a flow of capital	[13]
Attack attribution	[133]

CyBOK Wiki



Cross Reference of Topics vs Reference Material [edit]

Sections	Cites
Cyber-enabled and cyber-dependent crimes	[30]
Interpersonal offenders	[31], [32], [33], [34], [35]
Cyber-enabled organised criminals	[20], [23], [36]
Cyber-dependent organised criminals	[17], [37], [38], [39], [40], [41], [42]
Hacktivists	[43], [44], [45]
State actors	[46], [47], [48], [49]
Affiliate programmes	[25], [50]
Infection vectors	[51], [52]
Infrastructure	[53], [54], [55]
Specialised services	[56], [57]
Human services	[58], [59], [60], [61]
Payment methods	[62], [63], [64]
Attack trees	[1]
Environmental criminology	[3], [4], [5]
Modelling the underground economy as a flow of capital	[6]
Attack attribution	[7]

Tables get converted correctly out-of-the box, though some have issues (see the *To-do* list).

Bibliographic numbering differs because CyBOK PDF numbering happens on KA level while CyBOK Wiki numbering happens on (sub)section level.