DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies ..; .. (..):1–9

Zeta Avarikioti*, Roman Brunner, Aggelos Kiayias, Roger Wattenhofer, and Dionysis Zindros

The iceberg below the surface: An analysis of dark net content

Abstract: We analyze the type of content present on the “dark web”, the set of websites accessible via Tor. We create a darknet spider and crawl the whole darknet by starting from a bootstrap list and recursively following links so that the whole connected component of more than X websites and Y base URLs is explored. We publish our spider as open source software. We find that the darknet is well-connected through hub websites such as wikis and forums. We perform comprehensive measurements on the content found using machine learning to analyze and categorize the various types of content. We observe that the majority of darknet content belongs to P and Q. We close by discussing the political and ethical implications of these results.

Keywords: anonymity, tor, machine learning, scraping, spider

DOI Editor to enter DOI
Received ..; revised ..; accepted ...

1 Introduction

Describe the scope of the project, refer to previous work (e.g., contrast with Nicolas Christin’s work on Silk Road as well as the work where they ran their own relays to collect content). Clearly state our contributions. Summarize the methodology and results. Introduce visible vs invisible part of the darknet Visible part: the one that can be accessed using the Tor network without any further information needed. Invisible part: the ones that require authentication or any other knowledge to be found. We classify what a normal Tor user can find on the network. Of course, this does not cover all possible usages of the Tor network, but as more users access the tor network, we can be sure that most of them will stay in the visible part (therefor this is relevant)

Ok, here a short summary and why we are making something different than the linked articles/blogs:
https://www.hyperiongray.com/dark-web-map/ : Displays 6.6k websites in a "map". However, the map is sorted by structural similarity and many of the displayed pages are either offline (E.g. taken offline by law enforcement, by the hoster itself (only displays a notice) or are not yet setup ("I congratulate you! You have made a site on the dark side (onion)"). I think it’s clear that this is not what we are doing
TERBIUM LABS Work (I assume 2016 from the files name, but is not stated clearly): Reviewed a sample of 400 URLs, collected by their crawler, classified into one of the following categories

- Legal
- Explicit
- Drugs
- Pharmaceuticals
- Fraud
- Multiple Categories (Illicit)
- Falsified Documents Counterfeits
- Exploitation
- Hacking Exploits
- Weapons
- Extremism
- Weapons of Mass Destruction
- Other Illicit Activity
- Unknown/Site Down
- Downloadable File

They come to the conclusion that the darknet is mainly legal, however their work is based on a very small sample size
Interesting sections for us: "Handling Illegal Content and Exploitation", "Why we used a Random Sample"
Empirical analysis of Tor Hidden Services (ET Information Security, 2015): They collected their data by running a large number of Tor nodes for 6 months and collected 80’000 hidden services that way. They did a size estimation and a content categorization, results seem to approximately match up with *Content and popularity analysis ofTor hidden services*. Since they operated Tor nodes, they also collected which category

***Corresponding Author: Zeta Avarikioti:** Affil, E-mail: zetavar@ethz.ch
Roman Brunner: Affil, E-mail: robrunne@student.ethz.ch
Aggelos Kiayias: Affil, E-mail: aggelos.kiayias@ed.ac.uk
Roger Wattenhofer: Affil, E-mail: wattenhofer@ethz.ch
Dionysis Zindros: Affil, E-mail: dionyziz@gmail.com

produces what amount of traffic. They have a small section on Darknet connectivity and found that 59% of the pages do only link to itself. Topic pretty similar to ours, but the approach is different
Content and popularity analysis of Tor hidden services (2014): They used a flaw in the Tor Protocol to gather their data. That way they collected 39800 pages. They propose how one can deanonymise clients of HS, they did a content analysis as well and they looked at the open ports of the found hosts. This enabled them to give an overview of which software is used to deliver the HS. Topic also pretty similar to ours, but again, our approach is differen.

Conclusion: *Empirical analysis of Tor Hidden Services* and *Content and popularity analysis of Tor hidden services* do a pretty similar analysis on their data, but
a) The papers are at least 3 years old
b) They either cracked the Tor protocol or used Tor nodes to discover their data. We do our work on the "visible" part of the darknet and therefor do not acces any data the is not meant to be public (or only by human error - however, we are not the ones to blame for)
c) As far as I know there does not exist any work that has tried to apply a scraper to the darknet for classification (except terbium labs, but this is commercial and there paper only looked at 400 random samples by manual inspection. Our proposed technique allows for continuously scraping and classifying the content, which also allows to catch temporal changes in the topology of the network itself)
In short: our contribution: Tested the effectiveness of scraping, a non intrusive, non-deanonymizing technique to gather data from the Tor network. Further it would allow to capture temporal changes in the network. Since we can only look at publicly available pages, we do not compromise any data security or similar.

2 Scraping the Darknet

In order to get a complete scrape of the visible part of the Darknet, we created a scraper, which follows links recursively until it does not find new links. The whole project consists out of the following components:

- Scraper:
  - The scraper itself consists of a conductor, network module, parser module and a db connector.
  - While running the scraper we only use a small and fast, but not very precise uri extractor to get uris from downloaded content to scrape further (Based on cheerio for html files and RegEx for other text files).
- Sends a request over the TOR proxy and receives the response
- Upon response, we first check if this data type even is of interest to us. If not the data is discarded right away
- If it might be of interest, we first run it through the fast uri extractor, which then gets the base urls and paths for most of the links on the page
- To ensure we never have to wait on the database (since disk speed is typically higher than network speed of the TOR network, we want to maximize on used network), we keep a pool of downloadable URIs (size of pool can be configured specifically to the needed setting). These uris can be dispatched as soon as a download finished.
- We found that we need different prioritization schemes in order to decide which uri should be dispatched next. The first used scheme just retrieved new URIs inserted into the database ordered by insertion time. This type of prioritization has several shortcommings. For one, it might be exploring a small space of the darknet for a long time, since some pages link millions of times to themselves or we found clusters of scamming pages linking to each other in order to give themselves credibility (probably not yet jump to conclusions...). This lead not only to performance issues (not exploring new space) but also to technical issues: We did not want to DoS our targets nor overload the circuits, therefor we needed to limit the concurrent connections to the same host. Typical values are between 2 to 8 (http://www.stevesouders.com/blog/2008/03/20/roundup-on-parallel-connections/, needs better resource to cite), rfc 2616 even dicdates 2 (https://www.w3.org/Protocols/rfc2616/rfc2616-sec8.htmlsec8.1.4). However, most modern browsers do not comply with the limit of 2 anymore, therefor 4 seemed to be a sensible choice. After introducing the rate limiting we soon discovered that there are pages like bitcoin explorers, listing every transaction, block and so on ever made. This resulted in a huge increase in memory usage, since we cached uris that could not yet be sent due to too many in progress connections. These circumstances

led to the development of the following prioritization schemes (+ reasons why we needed those/initially introduced them)

  - random: Was the first approach to mitigat the issue with too many concurrent connections. Fast we found, that at some point during the scraping process, a large amount of the available uris were pointing to only a few hosts (If we look at the bitcoin explorer example: a page there has roughly 100 links to the same host, which results in an exponential grow of available paths of those hosts. The same holds true for libraries, where each page of a book is the linked individually)
  - new: Only get uris from hosts that were never scraped before. This works extremly well if we have a large collections of not yet explored links and does advance very fast through the network. However, often the page we found first for a host was not the most interesting or contained a lot of links to other hosts. Therefor this only works well for seed generation but not for data collection.
  - prioritized: Along each uri, we stored how often we already found an uri pointing to this (from any host) as well as from distinct hosts (So if we find an uri twice on two pages of the same host, this counts only once towards the count). It seemed that the more links from distinc pages a page has, the more important it might be. In general this holds true and gives a nice powerlaw distribution [– see powerLawAbnormality.png], however, we see a few abnormalities. The most obivious one stems from bitcoin scamming pages linking to a bitcoin explorer, trying to prove that they are credible. Therefor, it only works for a few pages, after that, we had a large number of not actually important uris that had the same priority and in the worst case even were uris of the same host.
  - recursive: One of the most advanced strategies, therefor it also is one of the most computing intensive (Multiple complex sql queries needed in order to gather all necessary information). The idea behind this prioritization scheme is to solve all of the above mentioned problems at the expense of a bit more computation time needed. The idea goes as follows: We check all available hosts and get one path for each host that has one available (e.g. not yet in progress and has unscraped hosts). Therefor we will make equally progress on all available hosts and newly available hosts are considered as soon as the pool is repopulated. Since a chosen host may have multiple paths available for download, we have to specify how we chose the path. We decided to again look at the number of hits a path has. There one can either take the path with the maximum or the minimum of hits. This is why there exists the inverse strategy, it prioritizes the path with the lowest number of hits.
  - inverse: Is basically the same as recursive, but with inversed prioritization (prioritize paths with a low number of hits over the ones with high numbers of hits).
  - combined: Combines the techniques above and tries to fit the current situation the best. So first it scrapes one path for each host (only unscraped hosts), if we do not have any (or enough, to repopulate the pool) unscraped hosts remaining, it switches to the prioritized Algorithm. If this does not return enough uris (e.g. because it is stuck in a step and only finds uris we already have too many connections to the same host), it uses the randomized approach to get more entries to the pool. This is an approximation to the recursive strategy. It is faster and uses less computational resources, however, it is not as precise as the recursive strategy. (Although it solves some of the weaknesses of its parts, it cannot solve them all, especially in the last fallback of the random algorithm, it cannot solve the issues of this particular algorithm)

- URI extractor: This is run alongside the scraper and rescans the downloaded contents for new (not yet found by the small/fast extractor) base urls (= new hosts) and paths of such hosts. This one needs to only run once through all the downloaded contents
- DB: The DB needs to store all the information needed for the scraper to find which entries were already scraped and which ones should be scraped next. For this we have a baseUrl table, a path table (a host has one baseUrl but can have multiple paths) and a contents table, where the downloaded

contents are stored. For classification we needed to clean up the content. The cleaned up contents then are stored in the clenContents table. to classify we built an inverted positional index which consisted of a posting, a term and a position table. Further a language table was required to assign each content a language (via foreign key). Last we have a labels table, which stores the class labels.

- TOR proxy: In order to speed up the scraping process, multiple tor instances were used, which were handled by a tor proxy server (https://github.com/znetstar/tor-router). Round robin distribution was used to distribute the download jobs to the multiple instances. The number of instances is configurable, in order to address different network performances for given machines.
- Data preprocessor: The datapreprocessor extracted text only from html and other textual representations. The extracted text then was split up into terms and inserted into the positional index. Further the language was guessed and referenced in the clean contents table.
- Classifier: A basic bag of words approach was used along with an SVM. In order to make training as efficient as possible, an active learning approach was used, where we first only manually classified a small dataset and then incrementally classified the entries for which the trained algorithm was most unsure. After that we retrained the algorithm and rerun it until we reached [F1 score, precision ... whatever quality measure we'll have]

2.1 Structure of the spider

The spider itself is written in Javascript and relies on NodeJS as the runtime environment. NodeJS was used since the spider is waiting most of the time for requests to return, be that to the database or to a hidden website. Therefor the NodeJS event loop came in handy and did not add a lot of overhead in development due to task or process handling. To store the data, we used PostgresDB Insert schema, which not only had to store the downloaded content but also all the links that were not yet visited. To access the Tor network we needed to add a Tor proxy in front of the spider itself.

FORMATTING To speed up the scraping process, we ran 100 concurrent requests to the Tor network. In order to prevent overload on the Tor nodes within the circuit, we used a pool of Tor of equal size, such that per request one Tor instance was used. The Tor instances were scheduled in a round robin manner

Describe the architectural decisions put in designing the spider. Mention the technologies we're using (node.js, postgres, the tor libraries). Include a diagram for our software architecture. Discuss how we bypass rate limits (e.g. by randomization of visits and by using multiple tor circuits in parallel) and what the rate limiting problems are (including rate limits by individual websites as well as by the tor circuit / network itself). How we avoid downloading illegal content (whitelists and blacklists, content types). Talk about bootstrapping the list, exploration depth. Show the numbers such as how many sites exist, how many base URLs exist, how many links exist, summarize them nicely in a table. Try to aggregate data from our database to create insights for the reader. Extract the "hubs" and state that, contrary to popular opinion, the darknet is well connected; specifically discuss which this hubs are, what their purpose is, and how many links they have for the top 10 hubs.

3 Analyzing content

Describe the types of content that we found. Talk about our methodology in how we analyzed the content using machine learning and NLP. How were the categories chosen? Discuss how we rank base URLs as well as individual paths. Show diagrams (a pie chart or bars?) with the popularity of various content categories.

4 Political and Ethical Discussion

Discuss the political implications of tor. Is it mostly illegal content? Does it matter? Discuss the ethical considerations of our research and make sure you mention that we didn't subvert the protocol itself but only collected public data.

5 Conclusion

Summarize our results and discuss directions for future work. One issue with the proposed method is the high volatility of the network. We found that up to 30% of the found hosts switch from being online to offline and vice versa in only a week. This circumstance could be faced by adding a temporal resolution to the data gathering process. The provided code already supports temporal resolutions for the hosts and paths, however, one has to match the links by timestamp.One addition could be that along the existing data collection, one also stores to which round of rescraping any entry (whether it now be a host, a path or a link) belongs.

6 Editorial Policy

6.1 Choice of reviewers

The Editor responsible for a given area of physics, turns to experts of the subject for opinion. Research articles and communications are reviewed by minimum two reviewers, review papers by at least three.

6.2 Suggestions from authors

Authors are requested to suggest persons competent to review their manuscript. However, please note that this will be treated only as a suggestion, the final selection of reviewers is exclusively the Editor's decision. The reviewers remain anonymous in any case.

The Editor is fully responsible for decision about the manuscript. The final decision, whether to accept or reject a paper, rests with him/her. The Managing Editor only communicates the final decision and informs the author about further processing.

6.3 Revised manuscript submission

When revision of a manuscript is requested, authors should return the revised version of their manuscript as soon as possible. Prompt action may ensure fast publication, if the paper is finally accepted for publication in .... If it is the first revision of an article authors need to return their revised manuscript within 60 days. If it is the second revision authors need to return their revised manuscript within 14 days. If these deadlines are no met, and no specific arrangements for completion have been made with the Editor, the manuscript will be treated as a new one and will receive a new identification code along with a new registration date.

6.4 Final proofreading

Authors will receive a PDF file with the edited version of their manuscript for final proofreading. This is the last opportunity to view an article before its publication on the journal's web site. No changes or modifications can be introduced once it is published. Thus authors are requested to check their proof pages carefully against manuscript within 3 working days and prepare a separate document containing list of all the changes that should be introduced. Authors are sometimes asked to provide additional comments and explanations in response to remarks and queries from the language and technical editors. In case the authors do not deliver the list of corrections to proofs in the requested time the manuscript will be published as is.

6.5 Reprints

Because the journal is published in an Open Access model, and has no printed version, the authors receive no reprints.

6.6 Erratum

If any errors are detected in the published material they should be reported to the Managing Editor. The corresponding authors should send appropriate corrected material to the Managing Editor via email. This material will be considered for publication in form of erratum in the earliest available issue of ....

6.7 Copyright

All authors retain copyright, unless – due to their local circumstances – their work is not copyrighted. The non-commercial use of each article will be governed by the Creative Commons Attribution-NonCommercial-NoDerivs license. The corresponding author grants De Gruyter Open the exclusive license to commercial use of the article, by signing the License to Publish. Scanned copy of license should be sent by e-mail to the Managing Editor of the journal, as soon as possible.

7 Paper writing guide

7.1 Paper elements

1. title page with:
   (a) title (short title),
   (b) full name(s) of author(s),
   (c) name and address of workplace(s),
   (d) personal e-mail address(es),
2. abstract,
3. up-to five keywords,
4. text,
5. reference lists.

7.1.1 Abstract

An abstract must accompany every article. It should be a brief summary of the significant items of the main paper. An abstract should give concise information about the content of the core idea of your paper. It should be informative and not only present the general scope of the paper but also indicate the main results and conclusions. An abstract should not normally exceed 200 words. It should not contain literature citations or allusions to the tables or illustrations. All non-standard symbols and abbreviations should be defined.

In combination with the title and key-words, the abstract is an indicator of the content of the paper. Authors should remember that on-line systems rely heavily on the content of titles and abstracts to identify articles in electronic bibliographic databases and search engines. They are therefore requested to take great care in preparing these elements.

7.1.2 Text

7.1.2.1 General rules for writing

- use simple and declarative sentences, avoid long sentences, in which the meaning may be lost by complicated construction;
- be concise, avoid idle words;
- make your argumentation complete; use commonly understood terms; define all non-standard symbols and abbreviations when you introduce them;
- explain all acronyms and abbreviations when they first appear in the text;
- use all units consistently throughout the article;
- be self-critical as you review your drafts.

Fig. 1. A figure caption should be placed **below** the figure.

7.1.2.2 Structure of a paper

Research papers and review articles should follow a strict structure. Generally a standard scientific paper is divided into:

- introduction: you present the subject of your paper clearly, you indicate the scope of the subject, you present the goals of your paper and finally the organization of your paper;
- main text: you present all important elements of your scientific message;
- conclusion: you summarize your paper.

Experimental part and/or calculations should be presented in sufficient details to enable reader to repeat the original work.

7.1.2.3 Footnotes/End-notes/Acknowledgments

We encourage authors to restrict the use of footnotes. If necessary, please make end-notes rather than footnotes. Allowable footnotes/end-notes may include:

- the designation of the corresponding author of the paper;
- the current address of an author (if different from that shown in the affiliation);
- traditional footnote content.

7.1.2.4 Tables

Authors should use tables only to achieve concise presentation, or where the information cannot be given satisfactorily in other ways. Tables should be numbered consecutively using Arabic numerals and referred to in the text by number. Each table should have an explanatory caption which should be as concise as possible.

7.1.2.5 Figures

Authors may use line diagrams and photographs to illustrate theses from their text. The figures should be clear, easy to read and of good quality. Styles and fonts should match those in the main body of the article. All figures must be mentioned in the text in consecutive order and be numbered with Arabic numerals.

Fig. 2. A figure caption for Fig. 2.

7.1.2.6 Typesetting

Type main text in roman (upright) font. The chemical symbols and compounds, units of measure, most multi-letter operators and functions should are written in roman upright as well. The variables, constants, symbols for particles, most single-letter operators, axes and planes, channels, types (e.g., n, p), bands, geometric points, angles, lines, chemical prefixes, symmetry designations, transitions, critical points, color centers, quantum-state symbols in spectroscopy, and most single-letter abbreviations should be written in roman italic. Boldface roman type is reserved for indicating vectors and in some special cases matrices.

7.1.2.7 Mathematical symbols

The multiplication signs are reserved for a vector product ($\mathbf{A} \times \mathbf{B}$) and simple dot product ($\mathbf{A} \cdot \mathbf{B}$). The only exception are numbers expressed in scientific notation (9.7×10^3 MeV).

7.1.2.8 Units

Units and dimensions should be expressed according to the metric system and SI units. This system is based on: meter (m), second (s), kilogram (kg), ampere (A), kelvin (K), mole (mol), and candela (cd). Most units are spaced off from the number, e.g. 12 mV. The only exceptions are:

$$1\%, 1‰, 1°\mathrm{C}, 1°, 1', 1''.$$

Decimal multiples or sub-multiples of units are indicated by the use of prefixes

$$\mathrm{µ}=10^{-6}, \mathrm{m}= 10^{-3}, \mathrm{c}= 10^{-2}, \mathrm{d}= 10^{-1}, \mathrm{da}= 10^{1},$$
$$\mathrm{h}= 10^{2}, \mathrm{k}= 10^{3}, \mathrm{M}= 10^{6}, \mathrm{G}= 10^{9}, \textit{etc.}$$

Compound units are written as

$$4221.9\ \mathrm{J\ kg^{-1}\ K^{-1}}\ \text{or}\ 4221.9\ \mathrm{J/(kg\ K)},$$

with a thin space between unit parts.

Authors should indicate precisely in the main text **where tables and figures should be inserted**, if these elements are given at the end in the original version of the manuscript (or supplied in separate files). If this information is not provided along with the manuscript, we will assume that the figures and/or tables should be insert at the closest position to first reference to them in the published paper.

7.1.2.9 Multimedia and images

Authors can attach files in most popular formats, including (for example):

– images in BMP, GIF, JPEG formats,
– multimedia files in MPEG or AVI formats.

However please keep to file types that are read by standard media players (e.g. RealPlayer, Quicktime, Windows Media Player) and/or standard office applications (Adobe Acrobat Reader, Microsoft Office etc.).

Your attachments may be accessible through links to external locations or to our internal locations (if you choose the second option, please remember to send us your attachments).

Please remember that your images, video and animation clips are intended for Internet use and we need to consider the needs of users with slow Internet connections. Please try to minimize file sizes by using a lower resolution or number of colors for images and animations (as long as the material is still clear). To help you in formatting your images (including tables and figures) or multimedia files, please submit your paper with separate attachments, which are used in your paper.

7.1.2.10 English language

Journal is published only in English. Make sure that your manuscripts are clearly and grammatically written. Please note that authors who are not native-speakers of English can be provided with help in rewriting their contribution in correct English. Try to prepare your manuscript in an easily readable style; this will help avoid severe misunderstandings which might lead to rejection of the paper.

7.1.3 Reference list

A complete reference should give the reader enough information to find the relevant article. All authors (unless there are six or more) should be named in the citation. If there are six or more, list the name of the first one followed by “et al”. Please pay particular attention to spelling, capitalization and punctuation here. Completeness of references is the responsibility of the authors. A complete reference should comprise the following:

7.1.3.1 Reference to an article in a journal

Elements to cite: Author's Initials. Surname, – if more authors, see examples below, Title of journal – abbreviated according to the ISI standards[1], volume number, page or article number (year of publication). Please supply DOI or URL for e-version of the papers. See Refs. [1–8] for example.

7.1.3.2 Reference to a book

Elements to cite: Author's Initials. Surname, Title, Edition – if not the first (Publisher, Place of publication, Year of publication) [9].

7.1.3.3 Reference to a part/chapter book

Elements to cite: Author's Initials. Surname, In: Editor's Initials. Editor's Surname (Ed.), Book Title, Edition – if not the first, (Publisher, Place of publication, Year of publication) page number [10].

7.1.3.4 Reference to a preprint

Elements to cite: Author's Initials. Surname, arXiv:preprint-number and version [11, 12].

7.1.3.5 Reference to a conference proceedings

Elements to cite: Author's Initials. Surname, In: Editor's Initials. Editor's Surname (Ed.), Conference, date, place (town and country) of conference (Publisher, place of publication, year of publication) page number [13].

7.1.3.6 Reference to a thesis

Elements to cite: Author's Initials. Surname, D.Sc./Ph.D./M.Sc./B.Sc. thesis, University, (town, country, year of publication) [14].

7.1.3.7 Reference to an article in a newspaper

Elements to cite: Author's Initials. Surname, Newspaper Title, date of publication, page number [15, 16].

7.1.3.8 Reference to a patent

Elements to cite: Originator, Series designation which may include full date [17].

7.1.3.9 Reference to a standard

Elements to cite: Standard symbol and number, Title [18, 19].

Please add language of publication for materials which are not written in English. Indicate materials accepting for publications by adding "(in press)". Please avoid references to unpublished materials, private communication and web pages.

You should make sure the information is correct so that the linking reference service may link abstracts electronically. For the same reason please separate each reference from the others.

Before submitting your article, please ensure you have checked your paper for any relevant references you may have missed.

7.1.4 Submission formats

Manuscripts for ... should be submitted in the $\LaTeX$ format with figures in EPS, PDF or PNG format. Authors are strongly encouraged to register their manuscript in arXiv preprint server and submit it to our Editorial Manager using arXiv's paper ID.

7.1.5 Supplementary data

You can also submit any supplementary data files as well. These may include long tables (in HTML or plain TXT format) or movies (preferably in AVI format).

References

[1] A. P. Raposo, H. J. Weber, D. E. Alvarez–Castillo, M. Kirchbach, Cent. Eur. J. Phys. 5, 253 (2007)

[2] J. Barth et al. (SAPHIR Collaboration), Phys. Lett. B 572, 127 (2003)

[3] S. Chekanov et al., Eur. Phys. J. C 51, 289 (2007)

[4] K. Malarz, Postepy Fizyki 57, 235 (2006) (in Polish)

[5] G. Meng, Cent. Eur. J. Phys., DOI:10.2478/s11534-007-0038-1

[6] R. Hegselmann, U. Krause, Journal of Artificial Societies and Social Simulation (2006), http://jasss.soc.surrey.ac.uk/9/3/10.html

[7] A. Dybala, Cent. Eur. J. Chem. (in press)

[1] http://images.isiknowledge.com/WOK46/help/WOS/0-9_abrvjt.html

[8] A. Dybala, Przeglad chemiczny (in Polish, in press)

[9] M. Lister, Fundamentals of Operating Systems, 3rd edition (Springer-Verlag, New York, 1984)

[10] C. K. Clenshaw, K. Lord, In: B. K. P. Scaife (Ed.), Studies in Numerical Analysis (Academic Press, London and New York, 1974) 95

[11] M. Majewski, K. Malarz, arXiv:cond-mat/0609635v2 [cond-mat.stat-mech]

[12] J. A. C. E. Solano, arXiv:0707.1343v1 [astro-ph]

[13] A. Kaczanowski, K. Malarz, K. Kulakowski, In: T. E. Simos (Ed.), International Conference of Computational Methods in Science and Engineering, Sep. 12-16, 2003, Kastoria, Greece (World Scientific, Singapore 2003) 258

[14] A. J. Agutter, Ph.D. thesis, Edinburgh University (Edinburgh, UK, 1995)

[15] A. Sherwin, The Times, Jul. 13, 2007, 1

[16] M. Dzierzanowski, Wprost, Jul. 8, 2007, 18 (in Polish)

[17] Philip Morris Inc., European patent application 0021165 A1, Jan. 7, 1981

[18] ISO 2108:1992, Information and documentation — International standard book numbering (ISBN)

[19] ISO/TR 9544:1988, Information processing — Computer-assisted publishing — Vocabulary