Preprint from https://www.gipp.com/pub/

C. Ihle et al. "A First Step Towards Content Protecting Plagiarism Detection". In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Aug. 2020. DOI: 10.1145/3383583.3398620

A First Step Towards Content Protecting Plagiarism Detection

Cornelius Ihle[1], Moritz Schubotz[2], Norman Meuschke[3], and Bela Gipp[4]

[1]Daimler AG & Univ. of Wuppertal, Germany ({first.last}@daimler.com)
[2]FIZ-Karlsruhe, Germany ({first.last}@fiz-karlsruhe.de)
[3,4]Universities of Wuppertal & Konstanz, Germany ({last}@uni-wuppertal.de)

May 26, 2020

Abstract

Plagiarism detection systems are essential tools for safeguarding academic and educational integrity. However, todays systems require disclosing the full content of the input documents and the document collection to which the input documents are compared. Moreover, the systems are centralized and under the control of individual, typically commercial providers. This situation raises procedural and legal concerns regarding the confidentiality of sensitive data, which can limit or prohibit the use of plagiarism detection services. To eliminate these weaknesses of current systems, we seek to devise a plagiarism detection approach that does not require a centralized provider nor exposing any content as cleartext. This paper presents the initial results of our research. Specifically, we employ Private Set Intersection to devise a content-protecting variant of the citation-based similarity measure Bibliographic Coupling implemented in our plagiarism detection system HyPlag. Our evaluation shows that the content-protecting method achieves the same detection effectiveness as the original method while making common attacks to disclose the protected content practically infeasible. Our future work will extend this successful proof-of-concept by devising plagiarism detection methods that can analyze the entire content of documents without disclosing it as cleartext.

1 Introduction

Plagiarism, i.e., the unacknowledged reuse of ideas or content, is a severe form of academic misconduct. Today, educational and research institutions, academic

publishers, and funding agencies increasingly rely on plagiarism detection systems (PDS) to identify plagiarized content [21]. Typical PDS require users to submit input documents, which the systems then compare to a large, typically proprietary database of documents. The systems retrieve documents with similar content as the input document and highlight the similar content to support user inspection [4].

Researchers and practitioners have criticized PDS for their poor detection accuracy and opaque computations [21], as well as their centralized and nontransparent data management [20, p. 72ff.]. In past research, we have addressed the first issue. We improved detection rates for heavily disguised instances of academic plagiarism by integrating the analysis of text-independent content elements, such as academic citations, images, and mathematical content with text-based detection methods into the hybrid plagiarism detection system HyPlag [14]. In this paper, we focus on the second issue.

Disclosing the full content of input and comparison documents to a central service provider whose detection methods and data protection measures are nontransparent, inherently raises concerns regarding the security and confidentiality of sensitive data, e.g., in unpublished research grant proposals or research theses compiled in cooperation with companies. The mere disclosure of such content to a third party can violate non-disclosure agreements, as well as data protection and copyright laws [20, p. 73f.]. Such legal concerns can limit the use of plagiarism detection systems. The general risk of data breaches further aggravates the problem. In Germany, many universities, therefore, prohibit the use of PDS entirely.

As we described in our vision paper [8], we seek to address the weaknesses of current PDS by devising a blockchain-backed decentralized approach to plagiarism detection (PD) that does not require disclosing any content as cleartext.

As a first step towards this vision, this paper reports on devising a content-protecting variant of the citation-based similarity measure Bibliographic Coupling (BC) [10] implemented in our PDS HyPlag [14]. We present an approach to securely mask bibliographic references and an adaption of the Private Set Intersection (PSI) approach to compute the bibliographic coupling strength (BCS) of papers without revealing the cleartext of the references.

2 Related Work

Our research shall enable plagiarism detection systems to identify similar content in documents without disclosing the content. Data security research has yielded two approaches to protect content while allowing the identification of the cleartext: Revertible functions (encryption) and lossy one-way functions (hashing).

Encryption is conceptually less secure than hashing because the encrypted content contains the full information of the cleartext. A malicious party can gain access to the cleartext by obtaining the decryption key. Additionally, encryption methods considered secure today can become vulnerable in the future due to

undiscovered flaws or increases in computing power [2].

Hash functions are lossy one-way functions that map cleartext of any size to a fixed-sized value (hash). Due to the lossy mapping, the cleartext cannot be recomputed from the hash. The only option for a malicious party to ascertain the correspondence of a hash and the cleartext is to guess the possible cleartext, compute its hash, and compare it to the hashes disclosed for a document. This approach, known as a *preimage attack*, is feasible if the set of hash inputs is finite and known [18]. Privacy-focused messaging applications like Signal face a similar problem called private contact discovery due to the finite set of phone numbers [11]. Signal solved this issue by performing PSI using a secured part of the CPU [12]. This solution does not apply to our distributed detection use case, as it still requires trust in the hardware underlying the service.

Researchers predominantly employed hashing to detect document similarity without revealing the documents' content.

The work of Unger et al. [19] is most related to ours since it also protects the content of documents during the plagiarism detection process. Their approach relies on a central authority (root node) that manages the access of peripheral nodes to content in the distributed system. The peripheral nodes represent documents as word chunks, which the nodes process using a collision-resistant hash function with a globally shared salt. The computed hashes are added to a count-min sketch [3] for tracking the frequency of word chunks in the document. The count-min sketches are shared with the root node and can be queried by the peripheral nodes. The privacy of all communication within the system is secured using the TLS protocol, with the root node acting as a certificate authority. While the approach greatly improves the confidentiality of content compared to traditional PDS, the count-min sketches are vulnerable to dictionary attacks. Moreover, the root node receives meta-data about documents, which can include the documents' subject matter and information on the authors' writing style.

Murugesan et al. [16] proposed the use of bloom filters in combination with hashes to protect the semantic meaning of content but maintain knowledge about the content's composition to perform similarity detection tasks. Bloom-filters are conceptually related to count-min sketches. Garbled Boom Filters track the frequency of content chunks in a document using a fixed-sized map while filling the empty positions with noise.

A drawback that all of the aforementioned hash-based approaches share is their vulnerability to preimage attacks. Furthermore, the approaches rely on a central authority.

Secure Multi-Party Computation (SMPC) [7] describes methods that overcome the need for a central authority. In SMPC, parties jointly compute a function over inputs, which the parties keep private. Most SMPC protocols, however, require translating all computations to binary circuits [17]. Employing SMPC for plagiarism detection would thus require new implementations of PD methods.

Many PD tasks represent an exchange of data between two instead of n parties, hence do not require the application of elaborate SMPC protocols. Instead, the tasks can be solved using the *Private Set Intersection (PSI)* [1] approach.

PSI allows two parties to compare private versions of their sets of data without revealing information to third parties. Hashing is the core of PSI. Many PD tasks exclusively require ascertaining the existence of identical features in documents of two parties. Hence, we consider PSI as promising for developing content-protecting versions of PD methods.

3 Method

In this paper, we consider bibliographic references as the only content to be compared confidentially. The bibliographic coupling algorithm considers the sets of references in the input and comparison document R_d and R'_d to compute the document similarity score bibliographic coupling strength (s_{BC}) as

$$s_{\mathrm{BC}}\left(R_d, R'_d\right) = \frac{|R_d \cap R'_d|}{|R_d \cup R'_d|} = \frac{|R_d \cap R'_d|}{|R_d| + |R_{d'}| - |R_d \cap R'_d|}. \tag{1}$$

Employing simple hashing to protect the confidentiality of bibliographic references is prone to preimage attacks as the number of published papers, and hence the number of possible references is finite. An attacker could acquire the metadata of most or all references, pre-compute their hashes and compare the pre-computed hashes of arbitrary references to the hashes of a protected document to deduce the topical context of the document.

To prevent preimage attacks for our use case, we hash combinations of k references instead of single references and only disclose the resulting set of combined hashes to the detection service. Without loss of generality, we assume that a preprocessing of references has been completed before forming the subsets. We further assume that the preprocessing step i) eliminated any duplicates in the reference lists of individual documents, ii) disambiguated all references in the collection, iii) stored the disambiguated references as a hashable data structure, and iv) excluded documents that contain less than k references from the similarity computation.

We form all k-combinations. For example, if a document contains three references $\{a, b, c\}$ and we seek to form reference subsets of cardinality $k = 2$, we would form, e.g., the subsets $\{a, b\}$, $\{a, c\}$, and $\{b, c\}$ but not $\{a, a\}$. Formally, we form the reference subsets

$$\mathcal{P}_k(R_d) = \left\{ r \subseteq R_d \middle| \, |U| = k \right\} \tag{2}$$

with cardinality $|\mathcal{P}_k(R_d)| = \binom{|R_d|}{k}$. Growing the set of possible hashes by a power of k increases the cost of a preimage attack but also the complexity of the detection process. Moreover, document pairs must contain at least k common references to exhibit a similarity that the content-protecting BC method can detect.

To mask the cleartext of references, we compute the set of hashes

$$H_d = \left\{ H(r) \middle| r \in \mathcal{P}_k(R_d) \right\}. \tag{3}$$

Here $H(r) = \sum_{i=1}^{k} H(r_i)$ denotes the hash function over the subset of references $r \subseteq R_d$. For the case $k = 1$, $H(r)$ yields the hashes of the individual references, i.e., $H_d = \{H(r_1), H(r_2), \ldots, H(r_n)\}$.

The detection service performs a private set intersection of the hashes from the input document and the hashes from previously submitted documents H'_d to compute the private BCS as

$$s_{\mathrm{PBC}}(H_d, H_{d'}) = \frac{|H_d \cap H_{d'}|}{|H_d \cup H_{d'}|}. \tag{4}$$

Similarly, one can derive $s_{\mathrm{BC}}(H_d, H_{d'})$ via

$$s_{\mathrm{BC}}(H_d, H_{d'}) = \frac{\eth_k(H_d \cap H_{d'})}{\eth_k(H_d) + \eth_k(H_{d'}) - \eth_k(|H_d \cap H_{d'})}, \tag{5}$$

where $\eth_k(j)$ is the numeric solution for $j = \binom{\eth_k(j)}{k}$. For example, for $k = 2$ one can derive $\eth_k(j) = \frac{1}{2}(1 + \sqrt{8j+1})$.

After comparing the hashes of an input document to the hashes of the corpus, we retrieve potential sources by ranking all corpus documents in descending order of their maximum s_{PBC} and filter for matches exclusively occurring in one document pair.

4 Experiments

Our experiments analyze the effectiveness, consumption of computational resources, and resistance to preimage attacks for our content-protecting version of the BC algorithm.

4.1 Experimental Setup

We conducted our experiments on a dataset of 105,120 arXiv documents, into which we embedded ten confirmed cases of plagiarism, each consisting of the plagiarized document and one source document. We used the same dataset in previous work [15]. We excluded documents without processable reference data and documents with more than 150 references. The final dataset contained 92,082 documents and 1,726,359 unique bibliographic references.

In a preprocessing step, we used the open-source software GROBID[1] to convert all documents into the uniform TEI-format[2]. TEI employs XML to structure the documents' content and allows for easy extraction of the bibliographic references.

Initial tests showed that the title field has the highest probability of being present in the reference string. Therefore, we used the normalized title field for the hashing. To decide on a hash function to use, we counted the number of hash collisions resulting from applying Adler32, SHA1, and SHA256 for hashing

[1] `https://github.com/kermitt2/grobid`
[2] `https://tei-c.org/`

Table 1: Hash generation for 92,082 documents.

k-Tuple	Hashes	Time in sec	Size in GB
k = 1	1,726,359	21	0.185
k = 2	45,951,328	31	2.5
k = 3	1,848,313,500	258	126

all reference subsets of size 3 in 5,000 documents. Only Adler32 yielded hash collisions (1,320), i.e., mappings of different inputs to the same hash, due to the comparably smaller size of its hashes (32-bits). We thus chose SHA1 as it offers sufficient collision resistance (2^{80}) and is faster to compute than SHA256. Collisions would cause false positives and, thus, an unnecessary effort for human reviewers.

4.2 Results

Effectiveness. To compare the effectiveness of PBC to the original BC method, we computed s_{PBC} and s_{BC} for all ten test cases in our dataset using subset sizes of 1, 2, and 3, respectively. We found that for both $k = 2$ and $k = 3$, the similarity scores computed by PBC and BC were equal for all test cases. This result shows that PBC detects identical references equally well as BC.

Resource Consumption. To assess the computational effort of our content-protecting PBC method, we analyzed the computation time and storage required for computing s_{PBC} depending on the size of k. We divided the analysis into two steps.

In the first step, we assessed the time and storage required for computing and storing the hashed reference subsets. **??** shows the results for analyzing the entire dataset of 92,082 documents using different sizes of k. The results show that the exponential growth of the space required for storing hashed reference subsets is the limiting factor for using larger subset sizes.

In the second step, we assessed the time required for performing the private set intersection of the hashed reference subsets depending on k. For this analysis, we only used 1,000 documents from our dataset. **??** shows the number of all hashes and the fraction of those hashes that occur in one, two, and three documents, respectively. The table also shows the time required for applying PBC to compute the bibliographic coupling strength of the input document with a comparison document. For increasing values of k, the number of hashes occurring in more than one document decays rapidly. For $k = 1$, 86% of the references occur in one document only. Combinations of three references are unique in 99.3% of the cases. The increase in required computation time is almost constant in the number of hashes due to the use of indexes.

Resistance to Preimage Attacks. To motivate the resistance of PBC to preimage attacks, we estimate the computation time required to perform such an attack, using computer science publications as an example. As explained in **??**, a preimage attack requires knowing the possible hash inputs, i.e., in our

Table 2: Detection against 1,000 documents

k-Tuple	Hashes	Ratio in 1/2/3 docs	Time in ms
k = 1	22,658	.86/.07/.03	98
k = 2	357,765	.98/.02/.001	103
k = 3	5,250,076	.99/.01/.0	118

case, the possible references in academic documents. To estimate the number of possible references for computer science, we use the dblp bibliography[3], which is the most comprehensive collection of bibliographic records for journal articles, conference papers, and monographs in this field. As of May 2020, dblp contains 5.05 million records.

For $k = 1$, a preimage attack on a single computer science document requires computing $\binom{5.05\times10^6}{1} = 5.05 \times 10^6$ hashes, i.e., a time complexity of $\mathcal{O}(n^k)$. Assuming a computing time of 1ms per hash would result in a single-threaded runtime of $5.05 \times 10^3 s \approx 1.40$h. Analogously, for $k = 2$, computing $\binom{5.05\times10^6}{2} \approx 12.75 \times 10^{12}$ hashes requires a single-threaded runtime of more than 404 years and more than 680 million years for $k = 3$. We see these numbers as conservative lower-bound estimates of the effort required because our calculation ignores i) interdisciplinary citations, ii) citations to sources not formally published, such as websites, code libraries, and datasets, and iii) the inevitable incompleteness of dblp[4].

The time complexity of $\mathcal{O}(n^k)$ for performing a preimage attack on a single document makes this attack too costly for $k \geq 2$ if the attacker cannot limit the possible references significantly, e.g., to a narrow research field. For example, for $k = 2$ and $x = 30{,}000$ possible references, the attack requires a single-threaded runtime of approx. 125h for one document. For $k = 3$, the required single-threaded runtime exceeds 100h per document for $x \geq 1{,}300$. We hypothesize that being able to restrict the possible references that sharply reduces the novel, i.e., unexpected information an attacker could obtain, hence reduces the benefit of a preimage attack greatly.

5 Conclusion and Future Work

We proposed PBC - a method that computes the bibliographic coupling strength of documents without revealing the bibliographic references involved in the computation. To realize PBC, we invented a new PSI approach that uses hashed feature subsets to prevent preimage and dictionary attacks. The security of the approach is adjustable to the computing resources that might be spent on the specific problem by increasing the number of features included in a subset, and by using hash functions with higher bit-lengths.

[3] `https://dblp.uni-trier.de`

[4] `https://dblp.uni-trier.de/faq/23593238.html`

We showed that PBC using SHA1 capably identifies similar documents in a large corpus without causing hash collisions. We demonstrated that a subset size of $k = 2$ achieves the best trade-off between computation time, required storage, and attack resistance. A subset size of $k > 2$ causes a steep rise in computation time and is therefore limited to documents with small numbers of references.

Hashed document feature subsets show promise for building a future decentralized PD service since accuracy would only decrease if a document pair does not contain at least two matching references. As shown in our prior work [6], an overlap of two references generally does not constitute a similarity that is significant enough to identify a document as suspicious of plagiarism.

In the future, we will allow encrypting the document IDs of unpublished documents. By doing so, the PDS can still detect that an input document overlaps with already existing unpublished documents in the distributed reference database. However, the PDS can no longer determine the number of documents with which the input document shares content. By using incentive mechanisms, the owner of the unpublished document is motivated to share the document with the PDS privately. To avoid false positives, authors can, e.g., submit previous versions of rejected grant proposals to prove that they were the authors of the earlier document.

In this initial study, we focused entirely on Bibliographic Coupling and excluded more sophisticated citation-based plagiarism detection methods like Greedy Citation Tiling and Longest Common Citation Sequences [5]. In the future, we will devise content protection methods that support pattern-based PD approaches that use mathematical features [15] and images [13]. We will analyze these detection methods and examine which features need to be masked and which features can be shared openly during the detection process without revealing any semantic information.

In summary, we successfully conducted the first step towards our vision of a decentralized content protecting plagiarism detection system [8]. The findings of our initial study confirm our research direction of using hashed document features, such as references, in-text citations, images, and formulae, to devise such a service.

To ensure the reproducibility of our experiments, our data and code are available at *https://github.com/ag-gipp/20CppdData*

References

[1] H. Chen, K. Laine, and P. Rindal. "Fast Private Set Intersection from Homomorphic Encryption". In: *Proceedings ACM Conference on Computer and Communications Security*. 2017, pp. 1243–1255. DOI: 10.1145/3133956.3134061.

[2] L. Chen et al. *Report on Post-Quantum Cryptography*. en. Tech. rep. 8105. NIST, Apr. 2016. DOI: 10.6028/NIST.IR.8105.

[3] G. Cormode and S. Muthukrishnan. "An improved data stream summary: the count-min sketch and its applications". In: *Journal of Algorithms* 55.1 (2005), pp. 58–75. DOI: https://doi.org/10.1016/j.jalgor.2003.12.001.

[4] T. Foltýnek, N. Meuschke, and B. Gipp. "Academic Plagiarism Detection: A Systematic Literature Review". In: *ACM Computing Surveys* 52.6 (Oct. 2019), 112:1–112:42. DOI: 10.1145/3345317.

[5] B. Gipp and N. Meuschke. "Citation Pattern Matching Algorithms for Citation-based Plagiarism Detection: Greedy Citation Tiling, Citation Chunking and Longest Common Citation Sequence". In: *Proceedings ACM Symposium on Document Engineering*. Sept. 2011, pp. 249–258. DOI: 10.1145/2034691.2034741.

[6] B. Gipp, N. Meuschke, and C. Breitinger. "Citation-based Plagiarism Detection: Practicability on a Large-Scale Scientific Corpus". In: *JASIST* 65.8 (Aug. 2014), pp. 1527–1540. DOI: 10.1002/asi.23228.

[7] O. Goldreich, S. Micali, and A. Wigderson. "How to Play ANY Mental Game". In: *Proceedings ACM Symposium on Theory of Computing*. 1987, pp. 218–229.

[8] C. Ihle. "A Privacy-Preserving and Decentralized Approach for Plagiarism Detection". en. In: *Proceedings of the Doctoral Consortium at the ACM/IEEE Joint Conference on Digital Libraries*. June 2019.

[10] M. M. Kessler. "Bibliographic coupling between scientific papers". In: *American Documentation* 14.1 (1963), pp. 10–25. DOI: 10.1002/asi.5090140103.

[11] M. Marlinspike. *The Difficulty Of Private Contact Discovery*. Jan. 2014.

[12] T. C. Maxino and P. J. Koopman. "The Effectiveness of Checksums for Embedded Control Networks". In: *IEEE Transactions on Dependable and Secure Computing* 6.1 (Jan. 2009), pp. 59–72. DOI: 10.1109/TDSC.2007.70216.

[13] N. Meuschke et al. "An Adaptive Image-based Plagiarism Detection Approach". In: *Proceedings ACM/IEEE Joint Conference on Digital Libraries*. June 2018. DOI: 10.1145/3197026.3197042.

[14] N. Meuschke et al. "HyPlag: A Hybrid Approach to Academic Plagiarism Detection". In: *Proceedings ACM SIGIR Conference*. 2018. DOI: 10.1145/3209978.3210177.

[15] N. Meuschke et al. "Improving Academic Plagiarism Detection for STEM Documents by Analyzing Mathematical Content and Citations". In: *Proceedings ACM/IEEE Joint Conference on Digital Libraries*. 2019. DOI: 10.1109/JCDL.2019.00026.

[16] M. Murugesan et al. "Efficient privacy-preserving similar document detection". In: *The VLDB Journal* 19.4 (Aug. 2010), pp. 457–475. DOI: 10.1007/s00778-009-0175-9.

[17] S. Riazi et al. "MPCircuits: Optimized Circuit Generation for Secure Multi-Party Computation". In: *Proceedings IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 2019, pp. 198–207. DOI: 10.1109/HST.2019.8740831.

[18] P. Rogaway and T. Shrimpton. "Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance". In: *Proceedings International Workshop on Fast Software Encryption (FSE)*. 2004.

[19] N. Unger, S. Thandra, and I. Goldberg. "Elxa: Scalable Privacy-Preserving Plagiarism Detection". en. In: *Proceedings ACM Workshop on Privacy in the Electronic Society*. 2016, pp. 153–164. DOI: 10.1145/2994620.2994633.

[20] D. Weber-Wulff. *False Feathers: A Perspective on Academic Plagiarism*. Springer, 2014. DOI: 10.1007/978-3-642-39961-9.

[21] D. Weber-Wulff. "Plagiarism Detectors Are a Crutch, and a Problem". In: *Nature* 567.7749 (2019), pp. 435–435. DOI: 10.1038/d41586-019-00893-5.

Listing 1: Use the following BibTeX code to cite this article

```
@InProceedings{Ihle2020,
  title = {A {First} {Step} {Towards} {Content} {Protecting}
      {Plagiarism} {Detection}},
  booktitle = {Proceedings of the {ACM}/{IEEE} {Joint} {
     Conference} on {Digital} {Libraries} ({JCDL})},
  author = {Ihle, Cornelius and Schubotz, Moritz and
     Meuschke, Norman and Gipp, Bela},
  year = {2020},
  month = {Aug.},
  topic = {pd},
  doi = {10.1145/3383583.3398620},
}
```