applied sciences

MDPI

Article

Lightweight, Secure, Similar-Document Retrieval over Encrypted Data

Mustafa A. Al Sibahee [1,2], Ayad I. Abdulsada [3], Zaid Ameen Abduljabbar [3,4], Junchao Ma [1,*], Vincent Omollo Nyangaresi [5] and Samir M. Umran [6,7]

1 College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China; mustafa@sztu.edu.cn or mustafa.alsibahee@iuc.edu.iq
2 Computer Technology Engineering Department, Iraq University College, Basrah 61004, Iraq
3 Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq; ayad.abdulsada@uobasrah.edu.iq (A.I.A.); zaid.ameen@uobasrah.edu.iq (Z.A.A.)
4 Shenzhen Institute, Huazhong University of Science and Technology, Shenzhen 430074, China
5 Faculty of Biological and Physical Sciences, Tom Mboya University College, Homabay 40300, Kenya; vnyangaresi@tmuc.ac.ke
6 School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; samirhust@gmail.com
7 Iraqi Cement State Company, Ministry of Industry and Minerals, Baghdad 10011, Iraq
* Correspondence: majunchao@sztu.edu.cn; Tel./Fax: +86-186-6620-3163

Abstract: Applications for document similarity detection are widespread in diverse communities, including institutions and corporations. However, currently available detection systems fail to take into account the private nature of material or documents that have been outsourced to remote servers. None of the existing solutions can be described as lightweight techniques that are compatible with lightweight client implementation, and this deficiency can limit the effectiveness of these systems. For instance, the discovery of similarity between two conferences or journals must maintain the privacy of the submitted papers in a lightweight manner to ensure that the security and application requirements for limited-resource devices are fulfilled. This paper considers the problem of lightweight similarity detection between document sets while preserving the privacy of the material. The proposed solution permits documents to be compared without disclosing the content to untrusted servers. The fingerprint set for each document is determined in an efficient manner, also developing an inverted index that uses the whole set of fingerprints. Before being uploaded to the untrusted server, this index is secured by the Paillier cryptosystem. This study develops a secure, yet efficient method for scalable encrypted document comparison. To evaluate the computational performance of this method, this paper carries out several comparative assessments against other major approaches.

Keywords: privacy-preserving similarity detection; document ranking; document fingerprinting; inverted index; Winnowing algorithm; lightweight client

Citation: Al Sibahee, M.A.; Abdulsada, A.I.; Abduljabbar, Z.A.; Ma, J.; Nyangaresi, V.O.; Umran, S.M. Lightweight, Secure, Similar-Document Retrieval over Encrypted Data. *Appl. Sci.* **2021**, *11*, 12040. https://doi.org/10.3390/app112412040

Academic Editor: Arcangelo Castiglione

Received: 20 September 2021
Accepted: 13 December 2021
Published: 17 December 2021

1. Introduction

Given the sheer quantity of content accessible via the World Wide Web [1], it is not difficult to appropriate another's theories or ideas as your own with no acknowledgement of the original author. Morally, it is important to avoid similarity or plagiarism in newly submitted works. This has become a fundamental issue in both the academic and non-academic spheres [2–4]. It is feasible that a plagiarist may amend the writing, theories, or character of another's work in order to present it as their own [5–8].

Document similarity detection (DSD) is present in several functional applications. During the DSD process, several similar files are gathered in a cluster in order to achieve effective file access. Thus, when one document is supplied, all similar documents are also retrieved, and in this way any duplication can be identified, allowing old versions to be marked or deleted. This methodology can be employed to detect whether, for instance,

articles newly submitted to a journal contain plagiarized work or sections of writing substantially similar to that previously published. However, the current DSD systems presume documents to be public and do not take the privacy of documents to be matched into account. This flaw can impede the effectiveness of the technique in many situations.

There are multiple practical circumstances in which it is necessary to detect the similarity measure with a specified questionable document in such a way that the privacy of the content is preserved. For example, to better understand common diseases and epidemics such as COVID-19, as well as the numbers of infected people; different health agencies may wish to jointly verify the similarity of their medical report documents. Privacy dictates that no agency may disclose its report to others. In this instance, it is essential to compare basic reports without breaching privacy guidelines. Furthermore, most journals prohibit double submissions, meaning that the same manuscript may not be submitted to two separate journals at the same time. Thus, the DSD must be able to verify this to be true while still preserving the privacy of each journal. Similarly, conference committees can use privacy-preserving, similarity-detection methods to check the originality of submitted papers.

Encryption is the best way to preserve the privacy of sensitive documents when they are uploaded to an untrusted server or shared in a public environment. However, encryption complicates conventional DSD tasks. Hence, it is imperative that an effective, lightweight, and secure DSD technique is designed that can be used with limited-resource devices to measure the similarity between two stored documents in the encrypted domain without violating one's privacy.

This paper adopts the concept of syntactic similarity. Fundamentally, when employing this concept, two documents with different keywords will not be considered alike, even if they are synonymous. The alternative approach is a semantic concept which uses complex techniques to examine the meaning of a document during the matching process, which results in high computational cost. This latter concept is outside the scope of this paper.

Current DSD methods can essentially be classified into two approaches: hashing [9–15] and vector space [16,17]. In the hash policy, a set of fixed-length substrings is extracted from the document. Then, the hash code is computed for each substring. Finally, a descriptive compressed fingerprint is generated from the hash codes. Two documents are considered similar if they contain a large number of shared fingerprint terms over and above a predefined threshold. This approach is best suited to capturing local similarities, that is, finding any overlapping content between two documents.

The alternative, vector space, approach is to employ information retrieval (IR) concepts to find and measure global similarity information. In the vector space model, each document is represented as a vector for terms or words, and each vector entry indicates the specific frequency information for the corresponding term. In this model, two documents are considered similar if they contain common terms which are repeated. However, the vector space model detects the similarity of entire documents, and can therefore identify two documents as being similar if they contain the same bag of words, even though the content of the documents is dissimilar. Compared with global similarity, local similarity characteristically gains more accurate results for documents to be retrieved. That is why the local similarity is adopted in our work [18,19].

The proposed scheme uses the hashing approach to generate a fingerprint for each document, where the fingerprint is a representative, yet compressed set of numbers. Once the fingerprint set of the entire document is defined, an inverted index structure can be built. The inverted index structure is used extensively and is widely employed in the information retrieval community [20,21] to provide fast data retrieval in a scalable manner for large document collections, as well as supporting fast query processing [22–24]. This study adopted a Winnowing algorithm to extract and select only fundamental properties from a fingerprint set, which not only significantly facilitates the secure search query, but also reduces space complexity, which results in low time-consumption. The inverted index

structure consists of a set of fingerprint terms and their corresponding posting lists. Each list includes a set of document IDs that contain the corresponding fingerprint term.

In order to implement the benefits of the inverted index whilst also safeguarding data, a secure, inverted index is created, and then a secure DSD methodology can be employed over that index. At this stage, a secret key is used to encrypt the index of fingerprints in such a manner that similarity detection can take place without revealing the contents of the underlying data. Only those in possession of the key can generate a valid fingerprint for those documented, which are to be matched.

This paper's contribution can be summarised as follows. First, it will demonstrate how documents can be compared in encrypted domains in a lightweight manner without compromising privacy and revealing the plain data. Second, it develops an efficient and secure solution to compute the common fingerprint terms between the document provided and the entire stored collection; subsequently, the paper will describe how the fingerprint approach can be employed innovatively to generate a secure inverted index, upon which a secure and lightweight privacy-preserving DSD can be constructed. This significantly simplified process results in a low client-side response time. Third, this method is appropriate for lightweight client implementation with limited-resource devices, with low complexity for query generation and a smaller key size for the encryption and decryption process of similar retrieved documents. It also allows for the smallest and most appropriate hash value to be selected by adopting a Winnowing algorithm, incurring low space complexity and low client response times. Fourth, the method proposed in this paper maintains the complexity of linear computation while increasing the scale of document collections. Finally, this study is able to detect duplicates, approximate duplicates, and previous submissions by academic venues privately, as well as to retrieve and rank similar encrypted documents.

The remainder of this paper is organized as follows: a summary of the most important achievements of related research is explored in Section 'Related works'; Section 'Document fingerprinting' illustrates the document fingerprinting technique; Section 'Scheme overview' introduces the problem definition and security requirements; Section 'Proposed scheme details' presents the proposed approach in terms of initialization and private similarity computation and search query phases; Section 'Security analysis' verifies the protection of data privacy; Section 'System evaluation' analyses the performance of the approach before the findings of the paper are summarised in the Section 'Conclusion'.

2. Related Works

The creation of the first method for detecting similarity in documents is credited to Manber [9]. His method uses the concept of hashing to assess the similarities within an extensive collection of documents, whereby a series of fingerprints is obtained from the substrings of a document and then compared with others to detect similarities. Subsequently, Brin et al. [10] applied the concept of the fingerprint to sentences. This procedure has been proven to successfully detect restructuring within phraseology. Velasquez [11] proposed a method of plagiarism detection based on the quantity of words present in both documents within relative sentences in order to assess the level of similarity between the two texts.

Schleimer et al. [12] developed a systematic algorithm for confined fingerprinting, known as the Winnowing algorithm, which extracts a fundamental property from a fingerprint set which is guaranteed to detect documents with similarities and decreases the time complexity of the search query. Sorokina et al. [13] employed this algorithm in order to develop a methodology for detecting intellectual theft among collections of textual work. Wang and Chang [25] latterly propounded a form of concise document rendition, which could be employed to effectively discriminate duplicate and almost duplicate documents from result lists.

Another system for uncovering intellectual theft has been suggested by Paul and Jamal [26]. Their concept consists of five principle elements, which are: pre-processing,

candidate retrieval, phraseology gradation, semantic role labelling, and comparability detection. The first element, pre-processing, is composed primarily of sentence partition, which separates the language into multiple phrases and then eliminates stop words, which are the most commonly used words in the language.

The next stage, candidate retrieval, involves the collection of a selection of texts comparable to the one being queried, utilizing n-grams and the Jaccard coefficient to gather the documents. After this, the sentence-ranking step generates a hierarchy of phraseology in the document being investigated in order to delineate original and questionable sentence pairings, while the resultant similarity level is determined via the principle of cosine similarity. Subsequently, the Semantic Role Labelling phase seeks to identify, based on the semantic association of its parts, the function in terms of the meaning of each section of a sentence, such as, for instance, the phrase's subject and object.

This mode of discernment targets shared meanings. The final stage, similarity detection, examines the ordered suspected and original phrases for comparability. Sections of other texts are placed against each section of the examinee text to produce results. Assessment has demonstrated that the pre-generated hierarchy of phrases using the sentence ranking step may enhance the efficacy of the system, yet it is costly in terms of the computation needed.

There is little extant research on the subject of secure DSD over encrypted data. The EsPRESSo protocol, proposed by Blundo et al. [14], implemented private set intersection cardinality (PSI-CA) and Jaccard similarity based on N-grams to ensure a privacy-preserving DSD. While EsPRESSo adopts MinHash to increase time efficiency, it unfortunately reduces document retrieval accuracy [27]. The work of Blundo et al. is far from being scalable, especially if large data or big data are adopted [28], while the authors do not take into account the scope of the compromised threat model. Instead, Blundo et al. took a pragmatic approach that emphasises ensuring that an adversary claiming to be either the server or the client gets the minimum amount of information by engaging in the scheme [29]. Thus, there can be no guarantee that parties with malicious behaviour intent can retain the same input through multiple interactions, so it remains an open and interesting problem.

The main difficulty when applying privacy-preserving DSD approaches to the comparing of scalable documents is the high cost of computation. To tackle the limitations of scalability and efficiency, this paper adopts an inverted index, since it is highly efficient for comparing multiple large document collections and providing fast retrieval of similar documents [20,21]. Moreover, this paper's method employs a Winnowing algorithm that candidates the smallest and most appropriate hash value while ensuring a reasonable retrieval accuracy, which is practical for real-world applications. Both the inverted index and Winnowing algorithm result in a lowered computational complexity of secure search queries, with only a slight effect on retrieval accuracy.

Dong et al. [15] proposed a new scheme to enhance computational efficiency based on Garbled Bloom Filters (GBFs) and an efficient (PSI-CA) protocol. Oblivious transfer (OT) is adopted to compute the similarity between two documents, which requires additional time and computational cost. This new work is called the oblivious Bloom intersection and the protocol requires expensive computation to construct GBFs. Specifically, where λ is a security parameter; λ public key operations are required for the server side, while at the client side, two λ public key operations are required for each Naor-Pinkas OT operation.

Murugesan et al. [17] used the vector space model to represent documents, and adopted cosine similarity to measure the global similarity between document pairs maintained by a third-party server. The main disadvantage of this approach is that it defines the global similarity, but not the local similarity. Unfortunately, global similarity often suffers characteristically less accurate document retrieval results than local similarity [18,19]. Jiang and Samanthula [30] used the k-gram technique to uncover local similarities on third-party servers. They used the random share method to compute the distance between the k-

gram sets in a privacy-preserving manner. However, such a method requires extensive computational and storage costs. Moreover, such studies lack experimental results.

Under both Refs. [17,30], it was said that the server should not know its stored data and the use of a third-party service, as the connection means that the privacy of the matched values is unlikely to be preserved. The technique proposed by our work avoids leaking any information to third parties or untrusted servers in the name of security. In addition, this method uses fingerprint selection to create an inverted index to achieve accurate matching identification and low computational overheads, which makes it compatible with lightweight client implementation.

Unger et al. [31] suggested a privacy-preserving plagiarism detection protocol that would be implemented by both the server and the client. Initially, both parties exchange a number of sentences. Then, in a certain order, they list pairs of sentences. After that, each pair of sentences is taken as an input. For such a pair, a private and secure multiparty computation protocol is implemented to generate the cosine and Sorensen-Dice metrics values. The thresholds are compared with these values in order to detect similarities between each pair of sentences. Since this process occurs for each pair of sentences between the two parties, it requires several rounds of communication, making this especially impractical for devices with limited resources. Additionally, these operations make high computational demands, since two sets of similarity metrics are applied. The method proposed by this paper requires only one round of communication between the two parties, whilst simultaneously providing a secure inverted index as well as Winnowing algorithm, resulting in efficiency savings in terms of both computational costs and time.

Abidin and Mitrokotsa's [32] study proves that Yasuda et al.'s [33] method is unsafe, because it adopts the properties of the somewhat homomorphic to protect the feature vectors of the biometrics for building privacy-preserving biometric authentication and matching.

3. Document Fingerprinting

In the detection of totally and partially similar or copied texts, fingerprinting can be effectively manipulated through the use of hash codes. Primarily, the text is split into a set of k-grams in order to obtain the fingerprint. This k-gram set is then hashed and compressed in the process of selecting a subset of these codes to form a fingerprint. The principal challenge posed by this technique is the selection of the most appropriate and smallest hash value from which to constitute the fingerprint. Thus, choosing the most appropriate technique has a great effect on reducing the size of the fingerprint set and facilitates the secure search query document. This can considerably reduce the encrypted comparison operations to retrieve similar documents, which plays an effective role in facilitating scalable documents comparing and reducing client response times.

The present study has selected the Winnowing algorithm [12], which nominates the most effective and smallest hash value from k-gram window slides. Particular boundaries exist for detecting similarities among duplicated and original texts when employing such hashed fingerprints. In the subsequent section, a concise explanation of the function of this algorithm is given.

Given the string St as a sequence of n characters, the k-gram is a substring of length k. For example, the 4 grams of the string St = 'to be or not to be' is {'tobe', 'obeo', 'beor', 'eorn', 'orno', 'rnot', 'nott', 'otto', 'ttob', 'tobe'} of length $n - k + 1$. Hashing the k-gram set is achieved by Karp-Rabin's algorithm [34]. This algorithm allows the hash of the $i + 1$st k-gram to be computed efficiently from the ith k-gram. Suppose that the first k-gram is a set of $t_1, \ldots, t_k$ numbers in the based b. Then, we can hash these numbers as:

$$F_1 = (t_1 \times b^{k-1} + t_2 \times b^{k-2} + \cdots + t_{k-1} \times b + t_k) \mod M, \quad (1)$$

where M is a constant defined by the user. The second k-gram of $t_2, \ldots, t_{k+1}$ numbers is computed efficiently as follows:

$$F_2 = (((F_1 - t_1 \times b^k) + t_{k+1}) \times b) \mod M. \quad (2)$$

Generally, the ith k-gram is computed as:

$$F_i = (((F_{i-1} - t_1 \times b^k) + t_{k+i-1}) \times b) \mod M, \forall i = 2 \ldots n. \quad (3)$$

Now the Winnowing algorithm selects the fingerprint among the $F_1, \ldots, F_{n-k+1}$ hash codes. Given any two documents, this algorithm guarantees finding substring matches between the documents which satisfy the following criteria:

1. The length of the matched strings is not less than the guarantee threshold, T.
2. The length of the matched substrings does not exceed the noise threshold, k.

Note that both $T > k$ and k are user-defined values. Choosing a large value of k prevents coincidental matching between two documents, however, it does limit the sensitivity to reordered document contents, since it cannot detect the relocation of any substring of a length less than k.

Therefore, it is necessary to choose a minimum value of k such that coincidental matches will be negligible. The algorithm defines a window size as: $w = T - k + 1$.

Each position in the sequence $1 \leq i \leq (n - k + 1) - w + 1$ defines a window of hashes $F_i, \ldots, F_{i+w-1}$. In each window, the minimum hash value is selected. Should there be more than one hash with the minimum value in the same window, only the rightmost occurrence must be selected. The same hash value among successive windows will not be inserted in the fingerprint. All selected hashes are considered to be the fingerprint of document. Consequently, it plays an essential role to select the much fewer numbers of actual hash values, incurring a low computational overhead.

For example, suppose we have the following hash codes: 77 72 42 17 98 50 17 98 8 88 67 39 77 72 42 17 98, and suppose that the window size is w = 4. All of the windows will be (77, 72, 42, **17**) (72, 42, 17, 98) (42, 17, 98, 50) (17, 98, 50, **17**) (98, 50, 17, 98) (50, 17, 98, **8**) (17, 98, 8, 88) (98, 8, 88, 67) (8, 88, 67, 39) (88, 67, **39**, 77) (67, 39, 77, 72) (39, 77, 72, 42) (77, 72, 42, **17**) (72, 42, 17, 98). According to the Winnowing algorithm [12], the fingerprint will be: 17 17 8 39 17, as illustrated by the bold font. The pseudocode of the Winnowing algorithm can be found in the Ref. [12].

4. Scheme Overview

4.1. Problem Definition

Consider a data owner *Bob* to have a private document collection $D = \{d_1, d_2, \ldots, d_m\}$ of size m. *Bob* outsources the storage and computing of his collection to a remote server S, such as a cloud server, to enjoy high-quality services at a reasonable and efficient cost. However, the server S is not trusted to see the contents of the stored collection, and as such, *Bob* has to encrypt his collection before outsourcing it.

To enable efficient retrieval, *Bob* also builds a secure inverted index from the private document collection and uploads it together with the documents to the non-secure server S, such as a cloud server. The smart device user, *Alice*, has a document Q which she wants to test for similarity against all the m documents that *Bob* has stored in S without either revealing Q to S or revealing D to *Alice*. In order to achieve this, *Alice* first extracts the fingerprint of her document, encrypts it, and then sends the result to the server S.

Upon receiving the secure fingerprint of the document Q from *Alice*, the server S matches the provided fingerprint against its secure index and then responds to *Alice* by the scores which represent the matching score for all the stored m documents. Finally, *Alice* downloads the top-h documents. Figure 1 shows the structure of the proposed work.

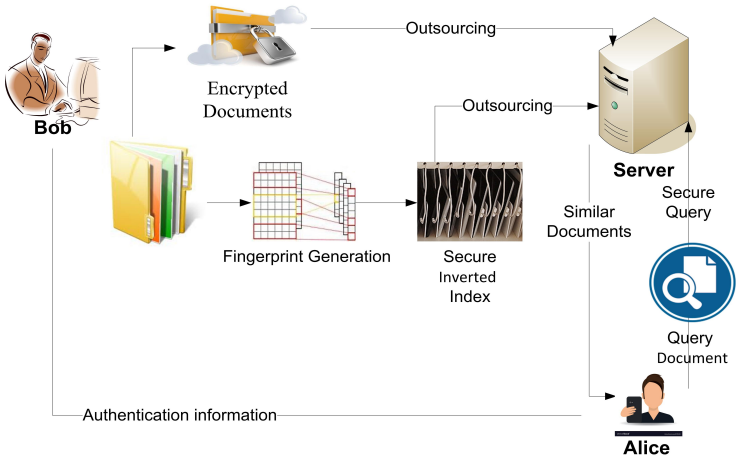

Figure 1. The proposed privacy-preserving DSD.

4.2. Security Requirements

This paper assumes that the server S is an honest but curious party, whose protocol can be trusted, but, at the same time, who will attempt to obtain as much information as possible from the stored data. The security requirements are as follows:

1. Document collection: The server S is not allowed to know the contents of *Bob*'s collection.
2. Query document: Query document for the authorized user *Alice* should not be revealed to server S, and the server S cannot generate a valid fingerprint query without knowing the secret key. A provided query should not leak any information about its underlying fingerprint. However, fingerprints are generated deterministically, so the server must not know whether the same query has been presented before; such leakage is known as query patterns.
3. Index security: The secure index does not leak any information about its contents.
4. The server must not know the search results for multiple searches or it may learn whether multiple searches have the same fingerprint; this is known as access patterns leakage.

5. Proposed Scheme Details

The following sections describe our suggested scheme. To enable effective document similarity detection, *Bob* creates a secure inverted index and outsources it to the remote server along with the encrypted documents. The server performs an index comparison according to data users' queries with no information about the data other than that which *Bob* has made accessible.

The secure DSD mechanism over encrypted data consists of two stages; namely, initialization, followed by private similarity computation and search queries. The "Initialization stage" subsection covers the configuration phase which includes index-generating, secure, searchable inverted index building and document encryption. The second subsection, "Private similarity computation and search queries", describes the encrypted document search and retrieval which is built on top of an inverted index. Figure 1 shows the mechanism of the secure and private DSD.

5.1. Initialization Stage

The data owner *DO Bob* initiates the process of the privacy-preserving DSD mechanism by generating the secret keys $K_1 = (b, M), kpub, kpriv$, and $Kcoll$, where b is the base parameter and M is a constant. Then the *DO* shares the following information with other authorized data users to launch a secure DSD.

- $Kcoll$: secret key of document collection encryption/decryption.
- K_1: secret keys of index construction.
- $kpub$: the public key of the Paillier encryption function.

- *kpriv*: the private key of the Paillier decryption function.

5.1.1. Secure Inverted Index.

Our proposed secure DSD is based on the inverted index structure to privately retrieve the documents with the top-*h* scores. The secure inverted index is constructed using the following two steps.

1. Fingerprint generation: Given the document collection $D = \{d_1, d_2, \ldots, d_m\}$. The data owner *Bob* generates the fingerprint for each document d_i, as explained in the "Document fingerprinting" section. Recall that the fingerprint is a set of integer numbers (terms). We define $F = \{f_1, f_2, \ldots, f_l\}$ to be the union set of all the fingerprint terms of the document collection. The inverted index I includes a set of fingerprint items f_j and their corresponding posting lists P_j i.e., $I = \{(f_j, P_j), j = 1, \ldots, l\}$.
   The posting list refers to the set of document IDs that contains the term f_j. The posting list P_j is simply a bit vector of size m, where m is the total number of *Bob*'s collection D. Given that $ID(d_i)$ is the identifier of the document d_i, the element vector $P_j[ID(d_i)] = 1$ if, and only if d_i includes the fingerprint term f_j. Table 1 shows a simple index of five fingerprint terms that were derived from 14 documents. For example, the term 500 appears in five documents (d_1, d_2, d_4, d_6, and d_{14}).
2. Inverted index encryption: In this stage, the inverted index is transformed into a secure index by encrypting the terms of fingerprints and their relevant posting vectors. Fingerprint terms (numbers) should be encrypted such that only the authorized users can generate valid queries. Otherwise, the third-party server could learn the fingerprint terms of a given document. Similarly, the posting vectors should be encrypted to hide the number of documents in a given fingerprint item, of which information may be used to conduct a frequency attack.
   Our solution to protect the fingerprint terms is to consider the parameters b and M of Equations (1) and (3) as a secret key $K_1 = (b, M)$, such that only the authorized users who have the secret key K_1 can generate a valid fingerprint.
   Encrypting the posting vectors is more difficult. This is because such vectors have to be encrypted, while preserving their ability to rank the retrieved documents. In this paper, we utilize the attractive feature of the Paillier cryptosystem [35] to alleviate this challenge.
   Paillier is a secure semantic and additive homomorphic asymmetric encryption scheme. The semantically secure feature ensures that repeatedly encrypting the same number (0 and 1 in our case) will generate different ciphers. Let Enc_{kpub} and Dec_{kpriv} be Paillier encryption and decryption functions with the public and private keys *kpub* and *kpriv*, respectively. Thus, if $m_1 = m_2$ are equal messages, then $Enc_{kpub}(m_1) \neq Enc_{kpub}(m_2)$. However, $Dec_{kpriv}(m_1) = Dec_{kpriv}(m_2)$. The additive homomorphic property means that $Enc_{kpub}(m_1 + m_2) = Enc_{kpub}(m_1) \times Enc_{kpub}(m_2)$. Given a constant c and a ciphertext $Enc_{kpub}(m)$, the multiplicative property can be defined as: $Enc_{kpub}(m)^c = Enc_{kpub}(c \times m)$.
   We use the Paillier cryptosystem to encrypt each bit of the posting vectors, such that if $P_j[ID(d_i)] = 1$, then we store $Enc_{kpub}(1)$. Otherwise, we store $Enc_{kpub}(0)$. Algorithm 1 shows the generation for the secure searchable inverted index built by the proposed method.
   It must be acknowledged that, since Paillier is a secured encryption method with regard to semantics, every encrypted zero and one value is distinct. As soon as the encryption process is complete, falsified data must be inserted as noise into the inventory in order to obscure the quantity of fingerprint items within the collection.
   REMARK: For more clarity that the inverted index is secure, it consists mainly of a set of fingerprint terms and the corresponding posting lists. Initially, Karp-Rabin's algorithm was used to hash the K-gram set in a secure and efficient manner. Then, the Winnowing algorithm selects the most important fingerprint among hash codes. For further security, these hash codes are protected using the parameters b and M of

Equations (1) and (3) as a secret key $K_1 = (b, M)$, whereas the Paillier cryptosystem is adopted to encrypt the posting lists.

The inverted index structure is often used to achieve fast and efficient data retrieval, and facilitates quick access to files in file management systems [20,21]. This paper implemented a Winnowing algorithm to reduce the fingerprint terms of the document collection, facilitating their management and maintenance. Practically, the Winnowing algorithm allows a query document search using only a portion of the fingerprint set, depending on the terms of the query document.

Table 1. Inverted index example.

Term	Vector													
500	1	1	0	1	0	1	0	0	0	0	0	0	0	1
520	0	0	0	1	0	1	0	0	1	0	0	1	0	0
600	1	0	0	0	0	0	0	1	0	0	0	0	0	0
680	0	0	1	0	0	0	1	0	0	0	0	0	1	0
710	1	0	1	0	0	0	0	0	0	1	0	0	0	1

Algorithm 1 Shows the building of the secure searchable inverted index.

Input: the document collection $D = \{d_1, d_2, \ldots, d_m\}$, Paillier encryption and decryption functions Enc_{kpub} and Dec_{kpriv}.
Output: the secure index I.
1: Start;
2: $F = \varnothing$
3: **For** each $d_i \in D$
4: Use equations 1 and 3 to generate the fingerprint
5: $df_i = \{f_{i1}, f_{i2}, \ldots, f_{ia}\}$;
6: $F = \cup df_i$;
7: **End for**
8: **For** each $f_i \in F$
9: **For** $i = 1 \ldots m$
10: If $f_i \in df_i$
11: Set $P_j(ID(d_i)) = Enc_{kpub}(1)$;
12: Else
13: Set $P_j(ID(d_i)) = Enc_{kpub}(0)$;
14: End if
15: **End for**
16: Store (f_j, P_j) in I;
17: **End for**
18: Add some fake records in I to hide the number of fingerprint items;
19: End;

5.1.2. Index Construction and Document Encryption

1. Index construction: *Bob* uses the secret keys K_1 and *kpub* to build the inverted index from the collection D as in Algorithm 1.
2. Document encryption: *Bob* encrypts his collection D with the secret key *Kcoll*. He then sends the encrypted collection to the server S. Thus, the privacy of the documents therein is protected. Once the data are outsourced, a data user should be able to match their documents with those held on the remote server. The AES encryption/decryption algorithm is commonly efficiently combined into popular and widespread lightweight devices, such as smart devices and IoTs, by means of hardware [36,37]. For further security, the counter mode CTR [38] is utilized with AES. For instance, the encryption process is $Enc_{kcoll}(.)$, where *kcoll* is the length secret key (128-bit). Note that the key is a smaller size, which significantly speeds up the encryption and decryption process.

For more protection, the document file is randomly mapped with fake data unrelated to the owner's genuine data to mask its real size. This random mapping is done after the secure searchable inverted index has been built to avoid padding the index with random data.

5.2. Private Similarity Computation and Search Queries Stage

Once the secure document set and the secure searchable inverted index have been outsourced to the remote server, authorized users can retrieve similar documents. These users can remotely access these documents via the server. In this subsection, the details of the proposed privacy-preserving DSD programme are explained. Algorithm 2 shows the private and secure DSD proposal.

Algorithm 2 Private and Secure DSD.

Input: the document Q.
Output: the Scores.
{*Alice*:}
start;
Use equations 1 and 3 to generate the fingerprint $df_i = \{f_{i1}, \ldots, f_{ia}\}$;
Send Qf to server S;
{*Server S*:}
For $i = 1 \ldots m$
  $Score(ID(id_i)) = Enc_{kpub}(0)$
End for
For $j = 1 \ldots c$
  If $gf_i \in I$
    For $i = 1 \ldots m$
      $Score(ID(id_i)) = Score(ID(id_i)) + Enc_{kpub}(e_{ij})$
    End for
  End If
End for
{*Alice*}
For $i = 1 \ldots m$
  $\propto_i = Dec_{krpiv}(Score(ID(id_i))$
End for
end;

1. Fingerprint construction: Suppose that *Alice* wants to compare her document Q with the collection D. She first uses the secret key K_1 to generate the secure fingerprint items $Qf = \{qf_1, qf_2, \ldots, qf_c\}$, as in Equations (1) and (3). Once completed, she sends Qf as a secure search query to the remote server S.
   REMARK: The fingerprint items are double-secured, first securely by hashing the K-gram set using Karp-Rabin's algorithm. Then, the Winnowing algorithm selects the fundamental property among hash codes. Second, these codes are securely protected by using the parameters b and M of Equations (1) and (3) as a secret key $K_1 = (b, M)$.
2. Search: Given the secure fingerprint set (search query) Qf, the server S searches its secure inverted index to find matched terms. For each matching item, S retrieves the corresponding posting vector, that is, retrieves $P_j = [e_{1j}, \ldots, e_{mj}]$ such that $(f_j, P_j) \in I$ and $f_j = qf_j$ for all $j = 1 \ldots c$, where e_{ij} is the encrypted bit of the document i corresponding to the fingerprint term j. Once the documents are retrieved, S calculates the score for each document d_i. To do so, S uses the additive property of the Paillier cryptosystem to get the score for each document as follows: $\propto (d_i) = e_{i1} + e_{i2} + \cdots + e_{ih}$, where h is the number of matched fingerprint items. Finally, the server S sends the scores $\propto (d_1), \propto (d_2), \ldots, \propto (d_m)$ to *Alice*. Most of the expensive processing is done on the server-side; specifically, the processes of matching terms and calculating the score are affordable for the server-side execution time.

3. Score decryption: On the client-side, *Alice* uses the secret key *Kpriv* to decrypt the received score values $\propto (d_i)$ for all $i = 1 \ldots m$, showing her which documents are similar to her own document Q. The score values are small in size and limited in number, so they do not invoke a significant computational and communication cost for client implementation.
4. Document retrieval: *Alice* asks the server S to retrieve the documents with the top-h scores. After retrieving the most similar documents, she decrypts them using an efficient symmetric cipher AES with the *Kcoll* 128-bit length secret key; because this uses a much shorter key, it reduces lightweight client computation requirements. *Alice* can then manually decide which of the top-h retrieved documents are the most suitable for her request.

Continuing with the index illustrated in Table 1, suppose that the terms of the query document are: 400, 500, 600, 710, and 800. Thus, the scores of the 14 documents are calculated as in Table 2. We see that Document 1 is the most similar with three scores, followed by Document 14 with two scores, and so on.

Table 2. Score calculation.

Term	**Vector**													
500	1	1	0	1	0	1	0	0	0	0	0	0	1	1
600	1	0	0	0	0	0	0	1	0	0	0	0	0	0
710	1	0	1	0	0	0	0	0	0	1	0	0	0	1
Score	**3**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**2**

6. Security Analysis

6.1. Security of Encryption

It is assumed here that the encrypted document database, secure searchable inverted index, and encrypted query will not disclose information to the non-secure remote server. It is reasonable to assert that the document database will be made secure through AES with CTR mode encryption. Nevertheless, certain problems remain that must be solved.

The encrypted document files may disclose vital information to a remote server due to their actual size. To address this issue, it is recommended that stored, encrypted documents are processed with additional random data padding so that their actual sizes are not easily detected. To prevent random values from being added to the secure searchable index, the padding process should take place after the safe inverted index has been created.

This paper believes that the schemes proposed in related works do not offer sufficient protection to the access pattern since private access processes all documents in the entire document set. This can happen by the server and any unauthorized party. As for *Alice*, she is an authorized user; however, she cannot obtain any information except what she obtains through the search query document because there is no link between the repeated search query documents in our work as it will be explained in the next subsection. This paper's researchers suggest that protecting the search pattern is not sufficient without protecting the access pattern, attributing this to the fact that the results of repeated searches will enable the attacker to link search queries immediately. Therefore, this paper's proposed design protects the access pattern since the remote server stores encrypted identifier IDs for the documents and no unsecured plain text.

To further increase security, the proposed design uses a cryptographic method to maintain a secure, searchable inverted index. More clearly, the homomorphic property was used to calculate all similarity scores on the encrypted queries on the server, which were then sent to the user side. The server's computing capabilities mean that this is not considered expensive on the server-side; while the similarity scores sent to the user is a small integers in practice. This limited number of similarity scores is because the documents of highest similarity are always performed, which not only reduces the cost

of communication but also reduces the cost of decryption on the client-side, facilitating efficient implementation.

To prevent attempted statistical attacks, it is advised that fake hash values are added to hide the actual number of values stored in the search index. The encrypted query is protected by similar means to those used in the secure searchable index.

To measure the amount of information that is disclosed to the remote server from the encrypted documents, the dependency between these encrypted documents and their corresponding plain text is quantified. The rationale behind this method is that low dependency translates into low information leakage. The mutual information (MI) entropy [38,39] is used to quantify the dependency between two entities. During the experiments conducted for this paper, safe documents yielded a low MI value, equal to 0.0014.

6.2. Search Anonymity

Each search query document has a different encryption key in the proposed model. Thus, two search queries have a different encrypted query even if they are from the same query document, which ensures that different search queries cannot be linked. By disabling linking queries, this scheme can provide search pattern confidentiality and privacy.

7. System Evaluation

7.1. Setup of Experiment

In this section, the experimental evaluation of the proposed scheme is presented. One thousand text documents were downloaded from the real dataset *Request For Comments* (RFC) [40]. During the fingerprint generation, 5-g were used, and the guarantee threshold T set to 10. The results are the average of 100 runs. The document collection was encrypted by an AES algorithm and the counter mode CTR with 128-bit key. The experiments were conducted on a 2.53 GHz Intel i5 M380 CPU, Windows 7 OS 64-bit, with 3 GB RAM. The experiments were conducted using MATLAB R2008a. The Paillier cryptosystem was implemented using Java class. The inverted index was stored in a hash table to provide $O(1)$ access time. The performance of the implementation was measured by searching the whole database through an inverted index with each potentially suspicious document to retrieve similar documents.

7.2. Retrieval Evaluation

In this experiment, the effectiveness of the proposed scheme to retrieve similar documents from the non-secure server's encrypted database was tested. This was accomplished by using the precision evaluation metric.

One hundred random, suspicious documents were selected as the query set. The comparison was assessed between this work and the resemblance R measure for computing the similarity between two documents. The R measure is defined as [41]:

$$R = |\text{Inter}|/|\text{Union}|, \tag{4}$$

where the Inter set represents the intersection set of two fingerprint sets, while the Union represents the union set. The ranking method used only employs the intersection set without using the union set to measure resemblance.

The precision P of the query document q is defined as:

$$P(q) = |R \cap A|/|A|, \tag{5}$$

where R is the retrieved documents in this study, while A is the set of retrieved documents under Equation (4). Where $Q = \{q_1, q_2, \ldots, q_x\}$ of x queries, the average precision can be

$$avg\_press = \frac{\sum_{i=1}^{x} P(q_i)}{x} \tag{6}$$

Figure 2 shows that the average precision is lower as the top-*h* documents increases. As expected, increasing the *h* value results in dissimilar documents being retrieved.

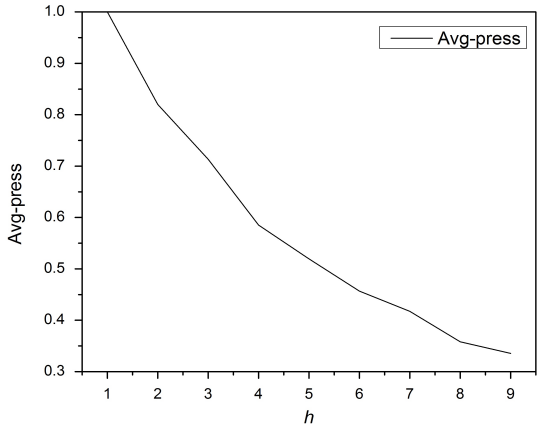

Figure 2. Retrieval evaluation.

7.3. Effectiveness

In this experiment, the goal was to use differential privacy, which works by adding or removing some noisy items into the query document without scarifying the accuracy of queries [42]. The efficacy of this work was evaluated to recover the inserted noise in the provided query. During the test, different amounts of noise were inserted into the query, ranging from 2% to 9% of the length of the query document.

Noise that represents the falsified data was inserted into the query document at random locations to hide the number of fingerprints. Details on the example datasets are provided below. Figure 3 illustrates that precision decreases as the amount of the inserted noise increases. The noise value of 0.6 was adopted in our proposed model, as it does not affect the retrieval accuracy at all.

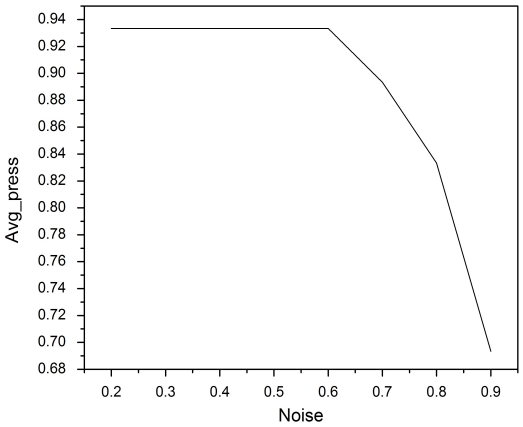

Figure 3. Effectiveness of the proposed scheme.

7.4. Index Building

The construction of the secret inverted index by the data owner, *Bob*, included generating the fingerprints and encrypting the posting vectors. In this experiment, the effect of the collection size, *m*, and the *k* value for the *k*-gram on the secure inverted index building time and the total number of individual fingerprint terms is illustrated. Figure 4 shows that increasing *k* when computing the *k*-gram sets provokes an increase in the fingerprint terms. Similarly, increasing *k* requires more time to compute the fingerprint terms, as explained in Figure 5.

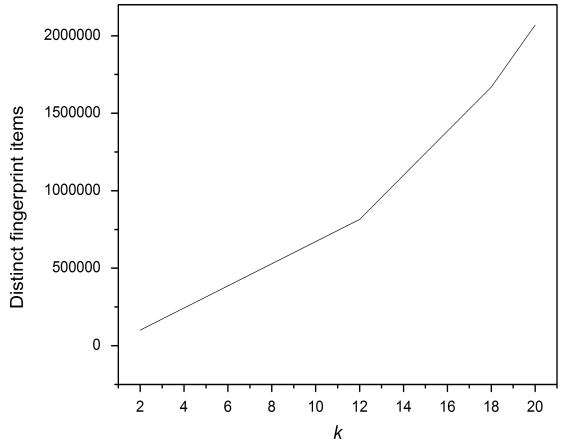

Figure 4. *K* effect on the number of fingerprint items.

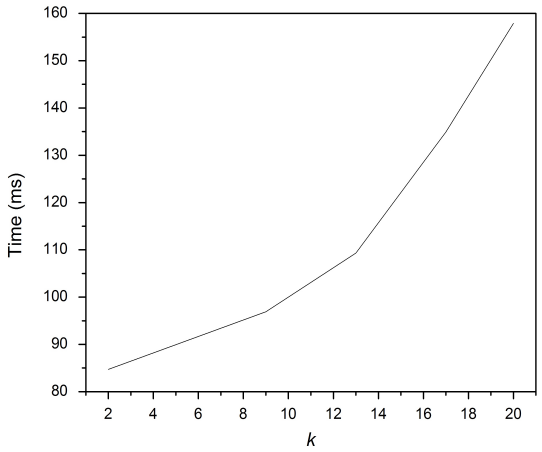

Figure 5. *K* effect on the time to generate the fingerprint items.

Similarly, Figures 6 and 7 show the effect of increasing the document collection size on the total number of fingerprint terms and indexing time, respectively. Again, increasing the collection size leads to increases in the two latter terms.

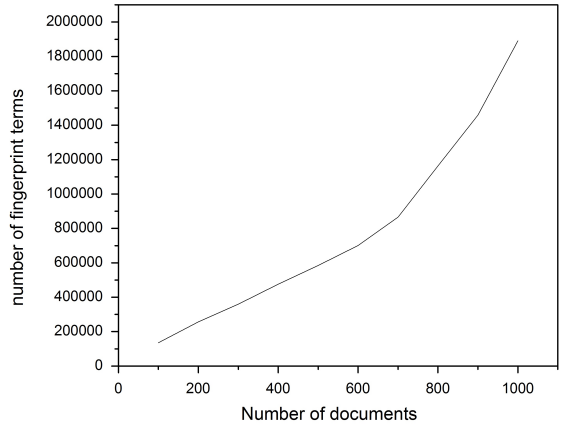

Figure 6. Effect of collection size on fingerprint size.

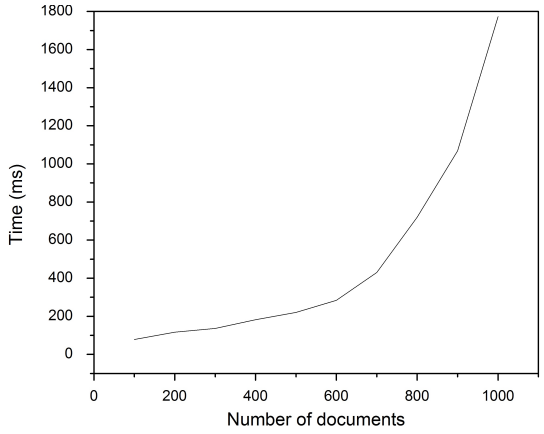

Figure 7. Effect of collection size on indexing time.

7.5. Ranking Time

The principal purpose of the ranking technique is to speed up the process of retrieving similar documents in an orderly manner. This process sorts the similarity scores of all the encrypted storage fingerprints in descending order to obtain the sorting result. Based on the ranking result, the target comparison document set is formed from each of the encrypted stored fingerprints with the highest score extracted according to the number of retrievals.

During the ranking time, the study sums up the posting vectors of the matched fingerprint terms. All vectors are of a fixed length equal to the collection size m. Figure 8 shows that more matched fingerprint terms require a longer ranking time. The document collection size for this experiment was fixed at 1000. Meanwhile, Figure 9 illustrates that increasing the posting vector requires more ranking time since longer vectors require more time to be added. From the foregoing, the time taken to rank files no longer affects the file retrieval process, as it is of intangible value (0.00019 ms) for the overall performance of the scheme and the client response time.

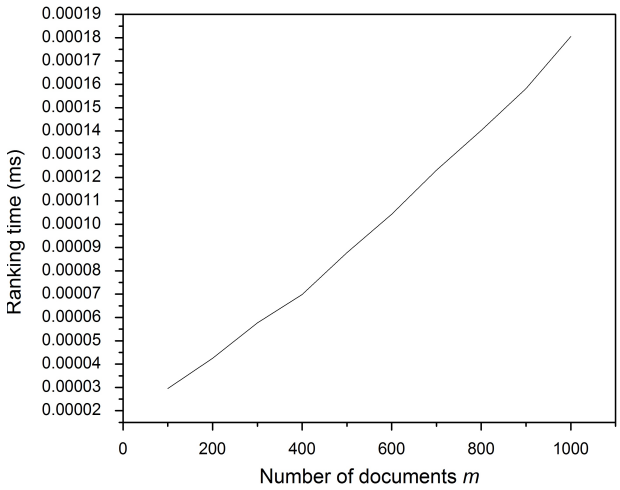

Figure 8. Ranking time: increasing the fingerprint items.

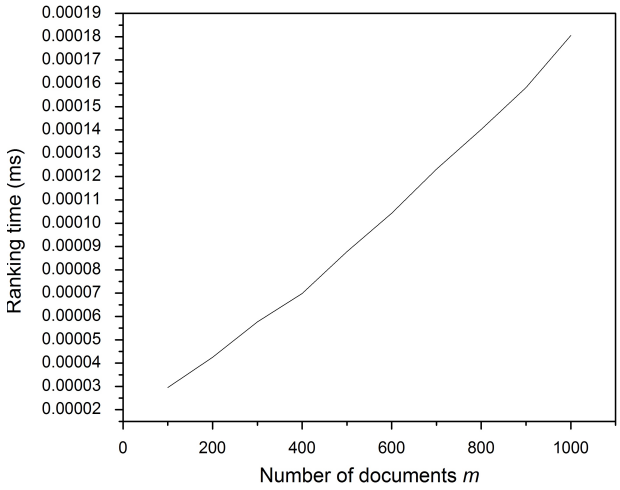

Figure 9. Ranking time: increasing the document collection.

7.6. Fingerprint Security

The fingerprint terms are protected by the secret key $K_1 = (b, M)$, where b is the base number used in Equations (1) and (3). Without knowing b, no party can generate valid fingerprint terms. In this experiment, the resemblance was computed as seen in Equation (4) between a fingerprint set generated by the valid secret key, $b = 50$, and 1000 fingerprints generated by 1000 different b values. One of these values is equal to the original. Figure 10 shows that only the valid b value has the maximum resemblance, although the values aligned to the valid key are 90%, 80% and 70%. This experiment verifies that fingerprint security is validated for the application of privacy-preserving document retrieval privacy.

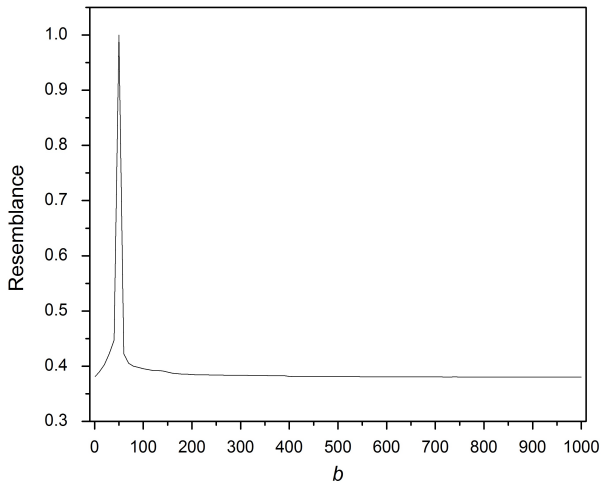

Figure 10. Effect of b on security.

7.7. Comparative Performance Analysis

The researchers understand that two protocols in the literature, namely EsPRESSo [14] and GBF [15], are most closely related to the work proposed in this paper. Therefore, to prove the efficiency of the proposed work, the results were compared against EsPRESSo and GBF, while a fair comparison was ensured by adopting 80-bit security parameters across all works. First, the researchers analysed the complexity of the three protocols in terms of the computational and communication costs, the results of which can be seen in Table 3; the computational and communication costs of EsPRESSo and this work are *O(m)* while the GBF costs are $O(m^2)$. Table 3 shows that the methods proposed in this study require the lowest computational costs compared with the EsPRESSo and GBF methods.

Table 3. Comparison of computational and communication costs (h: time required to implement hash; pk: time required to implement public and private key operations; e: time required to implement exponentiations; m: the number of documents.)

Work	Computation Cost	Communication (Bit)
Our work	h: $1.2 \times 10^2 \times m$, pk: 3.2×10^2	$1.2 \times 10^3 \times m$
EsPRESSo	h: $2.3 \times 10^2 \times m$, e: $3.9 \times 10^2 \times m$	$1.8 \times 10^2 \times m$
GBF	h: $3.0 \times 10^4 \times m$, pk: $4.4 \times 10^2 \times m^2$	$2 \times 10^4 \times m^2$

Furthermore, this study demonstrates a significant performance improvement over the alternative methods, thanks to the generation of the inverted index [20,21] and the use of a hash table for storage, which provides $O(1)$ access time. Moreover, this paper's approach uses a Winnowing algorithm to extract only the effective property and exclude others with less effect when retrieving similar documents, which reduces the computational complexity space. The communication costs of this study are higher than the EsPRESSo method since the latter uses Min-Hash, which generates lower communication costs, but it has disadvantages of reducing the accuracy of the similar documents retrieved [27], as it will be explained later in detail in Figure 11. Min-Hash measures approximate similarity, while the approach adopted by this study is designed to measure exact and approximate similarity for DSD. GBF has the highest communication cost because it considers the full set to represent each document.

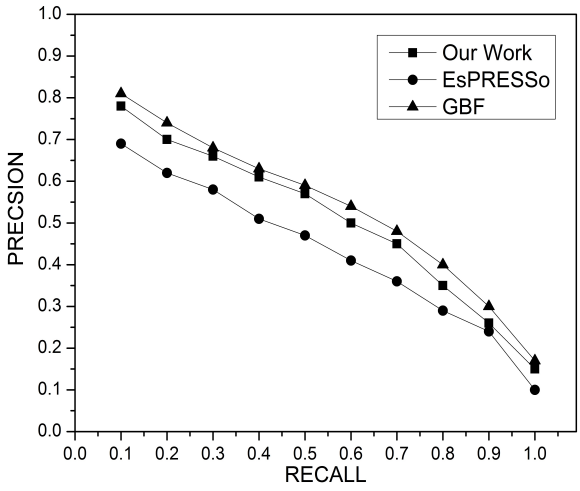

Figure 11. Performance of precision and recall.

Figure 12 shows the privacy-preserving DSD performance. A single document was compared with the document collection consisting of m documents. The results of the comparison show that all methods have m linear complexity. Of the three methods trialled, this study returned the best performance; the method proposed in this study improves the cost speed of document retrieval with $O(1)$ access time based on the inverted index [20,21], which includes a set of fingerprint items and their corresponding posting lists. While EsPRESSo performs slightly less efficiently in this regard as long as it uses a Min-Hash function, it is inefficient in retrieving scalable documents, as noted later. GBF returned the worst performance, since it is modelled for computation of big set intersections and requires an oblivious transfer (OT) [43] to compute the similarity between two documents.

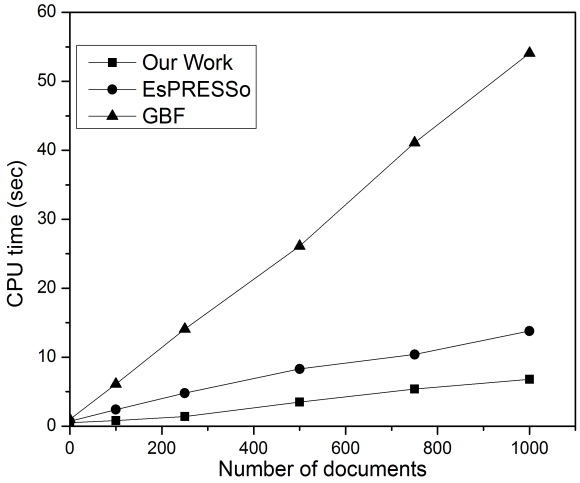

Figure 12. Performance of private and secure DSD ($1 \times m$ document).

Figure 13 shows the processing time consumed when detecting similarity between each pair of documents for both collections of size m. Recall that GBF's complexity scale is $O(m^2)$ times, as long as it requires (m^2) complexity to compare encrypted documents. It is clear that this paper's model and the EsPRESSo protocol have linear complexity with increased document size, but the proposed model has less processing time than EsPRESSo, which can be jointly attributed to the winnowing algorithm and the inverted index. The Winnowing algorithm focuses on ensuring that only the outstanding properties are extracted from the fingerprint set, making them as few as possible, while the access time of an inverted index is very short, which is $O(1)$.

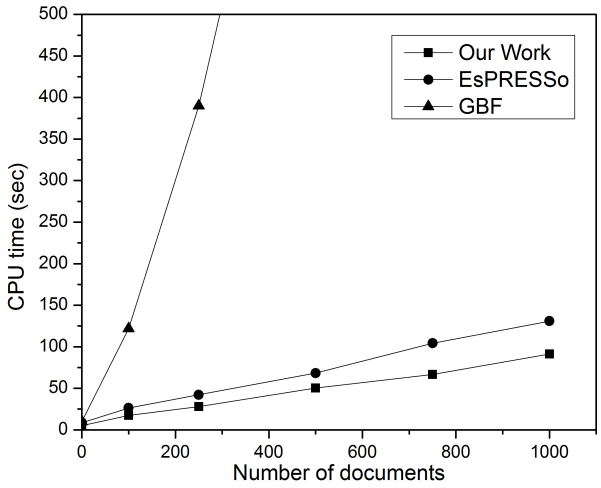

Figure 13. Performance of private and secure DSD ($m \times m$ documents).

To prove the efficiency and scalability of this method for privacy-preserving DSD between big collections, the same settings as the above experiments were adopted with a set of 5000 documents randomly downloaded from the real dataset *Request For Comments* (RFC) [40]. Figure 14 shows the processing time that this study's protocol and EsPRESSo take to compare two large sets of documents, each containing 5000 documents. It can be seen from Figure 14 that when the number of documents is small, EsPRESSo has a computational cost that is close to or slightly higher than this method; however, when the number of documents is large, the computational cost of this paper's method has a smaller growth rate and less processing time compared to EsPRESSo. This experiment therefore proves the claim that EsPRESSo is far from scalable when applied to large data [28]. Therefore, the work proposed in this study is more efficient for privacy-preserving DSD between big document collections, as well as for scalable documents.

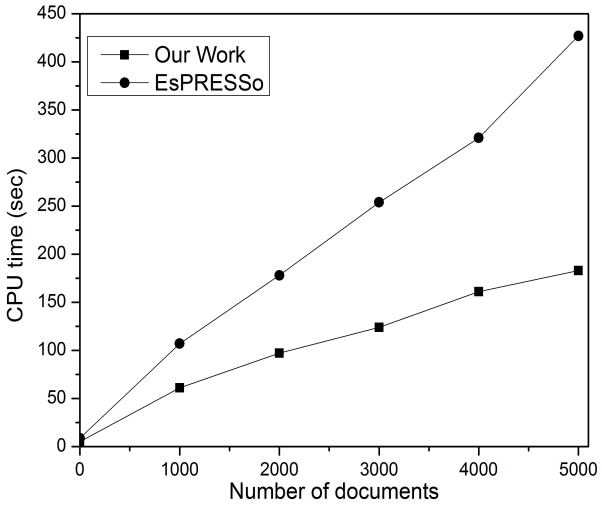

Figure 14. Performance of private and secure DSD (5000 × 5000 documents).

The following experiments compare the performance of the proposed work with AEsPRESSo and GBF in terms of precision and recall, as shown in Figure 11. The precision and accuracy are formulated as in Equations (4) and (5). GBF has the best overall effectiveness because it employs the full set to represent each document; however, this results in a very expensive computational cost, as noted in previous experiments and shown in Figures 12–14. The effectiveness of this paper's model is slightly lower than GBF since it uses a Winnowing algorithm that extracts and selects the fundamental property from a fingerprint set to reduce terms that represent documents. The selection of only the most important and effective fingerprint has led to a reduction in computational complexity. The AEsPRESSo model has the least effectiveness for precision and recall due to the precision loss in the Min-Hash that is used to represent and measure the similarity between two sets. This paper concludes that the accuracy of precision is highly sensitive to the way documents are represented.

The consuming time for query generation was estimated for this work, as well as the EsPRESSo and GBF protocols for sets of different sizes. Figure 15 shows that EsPRESSo is comparable to this protocol. This is because the EsPRESSo protocol applies the Min-Hash technique, which incurs significantly lower computation overheads; while this paper's scheme applies the Winnowing algorithm on the hash values set as a filtering mechanism to select only the fundamental property.

Thus, the consuming time of the query generation is highly sensitive to the number of hash values. Moreover, the slight difference in computational overheads occurred as a result of stronger security claims for query generation in the proposed protocol's security requirement. In any case, this difference is considered simple and does not have a tangible effect during implementation by the client, thus enabling the conclusion that EsPRESSo and the present study's method can each be applied with lightweight client implementation. It should also be noted that this paper's method has a higher performance in terms of document scalability and document retrieval accuracy, as shown previously. For query generation time, the GBF method has lower performance, indicating that it cannot be used in devices with limited resources. Recall at GBF required 2 $\nwarrow$ public key operations in the client side for each Naor-Pinkas OT operation in order to meet the stronger privacy claims in the GBF proposal.

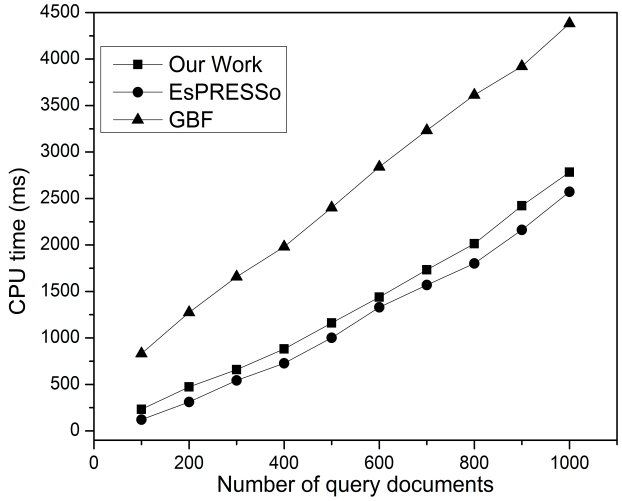

Figure 15. Performance query generation.

8. Conclusions

There is a real need for a methodical and functional method of retrieving and ranking private similar documents, especially in large text libraries. This work has tackled the issue of preserving privacy while detecting document similarity as well as retrieving and ranking these documents over encrypted data. In other words, the importance of the proposed work stems from its ability to compare encrypted documents in a way that reduces complexity while preserving privacy. The Winnowing algorithm was employed in this method to select the most important and effective fingerprint for each document in an efficient manner, which led to low space complexity and low response times for the client. Fingerprints were compressed versions of *k*-gram sets, capable of detecting both full and partial copies. Thus, an inverted index was developed based on the fingerprint set, permitting accurate, efficient, fast identification of matching documents, which is a greatly simplified process that results in low response times on the client side. This was then secured by utilizing a Paillier homomorphic encryption system. This cryptosystem allows the posting vectors to be summed without decryption. In order to demonstrate the functional benefits of this methodology, a number of experiments were conducted. These experiments prove that this method is more efficient than other major methods of privacy-preserving, secure, and similar document retrieval over encrypted data for lightweight client implementation with resource-limited devices. In addition, the linear computational complexity of our proposed work remains stable with the increase in the size of document sets. However, the communication cost of the proposed method is a limiting factor.

Future research will focus on developing this system to include the capability to detect plagiarism and distinguish inherent plagiarism without a reference corpus. There is also scope to improve the proposed system by attaching further functions to the service, as well as by utilizing cluster-based inverted indexing and parallel programming to lessen the runtime. Furthermore, the possibility exists to refine the methodology with an additional stage which will retrieve a text from the gathered group which is more similar to the document being queried. The addition of this measure might be proven to be influential in the overall performance of the proposed procedure. However, the study's limitations and implications for future research may remain affected by the communication cost.

Author Contributions: Conceptualization, M.A.A.S.; Funding acquisition, J.M.; Resources, V.O.N. and Z.A.A.; Software, V.O.N. and Z.A.A.; Supervision, J.M.; Visualization, M.A.A.S., S.M.U. and A.I.A.; Writing-original draft, M.A.A.S., A.I.A. and Z.A.A.; Writing-review and editing, M.A.A.S., A.I.A. and J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Natural Science Foundation of Top Talent of SZTU (grant No. 20211061010016).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sibahee, M.A.A.; Lu, S.; Abduljabbar, Z.A.; Liu, E.X.; Ran, Y.; Al-ashoor, A.A.J.; Hussain, M.A.; Hussien, Z.A. Promising Bio-Authentication Scheme to Protect Documents for E2E S2S in IoT-Cloud. In Proceedings of the 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Macau, China, 21–24 August 2020; pp. 1–6.
2. Tian, Z.; Wang, Q.; Gao, C.; Chen, L.; Wu, D. Plagiarism Detection of Multi-Threaded Programs via Siamese Neural Networks. *IEEE Access* **2020**, *8*, 160802–160814. [CrossRef]
3. AlSallal, M.; Iqbal, R.; Palade, V.; Amin, S.; Chang, V. An integrated approach for intrinsic plagiarism detection. *Future Gener. Comput. Syst.* **2019**, *96*, 700–712. [CrossRef]
4. V., G.P.; Velasquez, J.D. Docode 5: Building a real-world plagiarism detection system. *Eng. Appl. Artif. Intell.* **2017**, *64*, 261–271.
5. El-Alfy, E.S.M.; Abdel-Aal, R.E.; Al-Khatib, W.G.; Alvi, F. Boosting paraphrase detection through textual similarity metrics with abductive networks. *Appl. Soft Comput.* **2015**, *26*, 444–453. [CrossRef]
6. Sanchez-Vega, F.; Villatoro-Tello, E.; y Gomez, M.M.; Villasenor-Pineda, L.; Rosso, P. Determining and characterizing the reused text for plagiarism detection. *Expert Syst. Appl.* **2013**, *40*, 1804–1813. [CrossRef]
7. K, V.; Gupta, D. Text plagiarism classification using syntax based linguistic features. *Expert Syst. Appl.* **2017**, *88*, 448–464. [CrossRef]
8. Oberreuter, G.; Velasquez, J.D. Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Syst. Appl.* **2013**, *40*, 3756–3763. [CrossRef]
9. Manber, U. Finding Similar Files in a Large File System. In *USENIX Winter 1994 Tecnichal Conference*; USENIX: Berkeley, CA, USA, 1994; pp. 1–10.
10. Brin, S.; Davis, J.; Garcia-Molina, H. Copy Detection Mechanisms for Digital Documents. *SIGMOD Rec.* **1995**, *24*, 398–409. [CrossRef]
11. Velasquez, J.D.; Covacevich, Y.; Molina, F.; Marrese-Taylor, E.; Rodriguez, C.; Bravo-Marquez, F. DOCODE 3.0 (DOcument) COpy DEtector): A system for plagiarism detection by applying an information fusion process from multiple documental data sources. *Inf. Fusion* **2016**, *27*, 64–75. [CrossRef]
12. Schleimer, S.; Wilkerson, D.S.; Aiken, A. Winnowing: Local Algorithms for Document Fingerprinting. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD03), San Diego, CA, USA, 9–12 June 2003; pp. 76–85.
13. Sorokina, D.; Gehrke, J.; Warner, S.; Ginsparg, P. Plagiarism Detection in arXiv. *arXiv* **2007**, arXiv:cs/0702012.
14. Blundo, C.; De Cristofaro, E.G.P. Espresso: Efficient privacy-preserving evaluation of sample set. In *Data Privacy Management and Autonomous Spontaneous Security*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7731, pp. 89–103.
15. Dong, C.; Chen, L.; Wen, Z. When private set intersection meets big data: An efficient and scalable protocol. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS13), Berlin, Germany, 4–8 November 2013; pp. 789–800.
16. Shivakumar, N.; Garcia-Molina, H. Building a Scalable and Accurate Copy Detection Mechanism. In Proceedings of the First ACM International Conference on Digital Libraries (DL96), Bethesda, MD, USA, 20–23 March 1996; pp. 160–168.
17. Murugesan, M.; Jiang, W.; Clifton, C.; Si, L.; Vaidya, J. Efficient Privacy-preserving Similar Document Detection. *VLDB J.* **2010**, *19*, 457–475. [CrossRef]
18. Christian, K.; Matthias, G.; Petra, K.G.; Matthias, G. Methods for a similarity measure for clinical attributes based on survival data analysis. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 457–475.
19. Hua, Y.; Wang, S.; Liu, S.; Cai, A.; Huang, Q. Cross-Modal Correlation Learning by Adaptive Hierarchical Semantic Aggregation. *IEEE Trans. Multimed.* **2016**, *18*, 1201–1216. [CrossRef]
20. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
21. Jiang, D.; Leung, K.W.T.; Yang, L.; Ng, W. TEII: Topic enhanced inverted index for top-k document retrieval. *Knowl.-Based Syst.* **2015**, *89*, 346–358. [CrossRef]
22. Ding, S.; Attenberg, J.; Suel, T. Scalable Techniques for Document Identifier Assignment in Inverted Indexes. In Proceedings of the 19th International World Wide Web Conference, Raleigh, NC, USA, 26–30 April 2010; pp. 311–320.
23. Figueroa, K.; Camarena-Ibarrola, A.; Reyes, N. Shortening the Candidate List for Similarity Searching Using Inverted Index. In *Pattern Recognition*; Roman-Rangel, E., Kuri-Morales, Á.F., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Olvera-López, J.A., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 89–97.
24. Chandwani, G.; Ahlawat, A.; Dubey, G. An approach for document retrieval using cluster-based inverted indexing. *J. Inf. Sci.* **2021**. [CrossRef]
25. Wang, J.H.; Chang, H.C. Exploiting Sentence-Level Features for Near-Duplicate Document Detection. In Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology (AIRS09), Sapporo, Japan, 21–23 October 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 205–217.
26. Paul, M.; Jamal, S. An Improved SRL Based Plagiarism Detection Technique Using Sentence Ranking. *Procedia Comput. Sci.* **2015**, *46*, 223–230. [CrossRef]
27. Lee, H.; Wu, Z.H.; Zhang, Z. Dimension Reduction on Open Data Using Variational Autoencoder. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 1080–1085.

28. Xu, R.; Morozov, K.; Basu, A.; Rahman, M.S.; Kiyomoto, S. Security Analysis of a Verifiable Server-Aided Approximate Similarity Computation. In *Advances in Information and Computer Security*; Obana, S., Chida, K., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 159–178.
29. Simhadri, S.; Steel, J.; Fuller, B. Cryptographic Authentication from the Iris. In *Information Security*; Lin, Z., Papamanthou, C., Polychronakis, M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 465–485.
30. Jiang, W.; Samanthula, B.K. N-Gram Based Secure Similar Document Detection. In *IFIP Annual Conference on Data and Applications Security and Privacy*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 239–246.
31. Unger, N.; Thandra, S.; Goldberg, I. *Elxa: Scalable Privacy-Preserving Plagiarism Detection*; WPES '16; Association for Computing Machinery: New York, NY, USA, 2016; pp. 153–164.
32. Abidin, A.; Mitrokotsa, A. Security aspects of privacy-preserving biometric authentication based on ideal lattices and ring-LWE. In Proceedings of the 2014 IEEE International Workshop on Information Forensics and Security (WIFS), Atlanta, GA, USA, 3–5 December 2014; pp. 60–65.
33. Yasuda, M.; Shimoyama, T.; Kogure, J.; Yokoyama, K.; Koshiba, T. Practical Packing Method in Somewhat Homomorphic Encryption. In *Data Privacy Management and Autonomous Spontaneous Security*; Garcia-Alfaro, J., Lioudakis, G., Cuppens-Boulahia, N., Foley, S., Fitzgerald, W.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 34–50.
34. Karp, R.M.; Rabin, M.O. Efficient Randomized Pattern-matching Algorithms. *IBM J. Res. Dev.* **1987**, *31*, 249–260. [CrossRef]
35. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
36. Shafagh, H.; Hithnawi, A.; Droescher, A.; Duquennoy, S.; Hu, W. Talos: Encrypted Query Processing for the Internet of Things. In *SenSys 15 13th ACM Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2015; pp. 197–210.
37. Abduljabbar, Z.A.; Jin, H.; Ibrahim, A.; Hussien, Z.A.; Hussain, M.A.; Abbdal, S.H.; Zou, D. Privacy-Preserving Image Retrieval in IoT-Cloud. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; pp. 799–806.
38. Bakhache, B.; Ghazal, J.M.; Assad, S.E. Improvement of the Security of ZigBee by a New Chaotic Algorithm. *IEEE Syst. J.* **2014**, *8*, 1024–1033. [CrossRef]
39. Venkatasubramanian, S. Measures of Anonymity. *Priv.-Preserv. Data Min. Model. Algorithms* **2008**, *34*, 81–103.
40. RFC. Request For Comments Database. Available online: http://www.ietf.org/rfc.html (accessed on 13 December 2021).
41. Broder, A. On the Resemblance and Containment of Documents. In Proceedings of the Compression and Complexity of Sequences (SEQUENCES97), Salerno, Italy, 13 June 1997; pp. 21–30.
42. Li, Y.; Yang, D.; Hu, X. A differential privacy-based privacy-preserving data publishing algorithm for transit smart card data. *Transp. Res. Part C Emerg. Technol.* **2020**, *115*, 102634. [CrossRef]
43. Rabin, M.O. How to exchange secrets with oblivious transfer. *IACR Cryptol. ePrint Arch.* **2005**, *34*, 187.