Blocking Techniques On Privacy-Preserving Record Linkage

Thomas Meijerman
University of Twente

August 3, 2023

Abstract

Privacy-preserving record linkage solves the problem of matching records over two different datasets without leaking any private information about the subject of those records. Research [8] has shown that a certain standard, IND-S2PC, has to be met to call a two-party protocol privacy-preserving. Furthermore, privacy-preserving blocking techniques were introduced to increase the performance of the privacy-preserving record linkage protocols and preserve the privacy of the relevant subjects of the records. However, many privacy-preserving blocking technique papers do not include their privacy definitions or a privacy analysis of the their work. Here it is shown that many of these privacy-preserving blocking techniques do not satisfy IND-S2PC. To verify privacy-preserving blocking techniques with more participating parties, this research builds on IND-S2PC in the form of IND-S3PC and IND-SMPC. The former is a privacy definition for protocols including two parties and a computing third party. The latter is a privacy definition for protocols with multiple participating parties. Two privacy-preserving blocking techniques are highlighted and it is shown how IND-S3PC can be applied. One of the blocking techniques is proven to be privacy-preserving according to the IND-S3PC definition and the other is proven to be not.

1 Introduction

The protection of data and especially personal data is an area of research that keeps developing. Data has become valuable, making it a target for attackers to steal. This is also the case in the medical world [4]. Medical records contain personal identifiable information (PII) and are stored and protected as is required by e.g. the General Data Protection Regulation (GDPR) [5] for European related records. To access these records outside the jurisdiction of the database owner (e.g. a hospital), permission must always be granted by the subject (e.g. the patient). These measures are all to protect the privacy of the patients. However, there are other parties who want to access medical data for benign ends. Medical research is such a party and, to protect privacy, has to obtain consent from every participant in their study.

For preservation of the subjects' privacy, techniques have been developed to preserve the privacy and still draw meaningful conclusions from personal data for medical research. For example, the medical databases have to be encrypted and there are different techniques to work with this encrypted data. Note that the encrypted databases are still considered to be containing personal data, therefore enforcing all regulations set by the GDPR. The goal here is to preserve the privacy of the subjects.

Different techniques of working with encrypted data include computing statistics under encryption. For example, computing the median like described by Böhler et al. [3]. Furthermore, joining databases is critical for medical research as well as to get a full picture of a subject visiting multiple medical institutions. The joining of multiple databases can be done by, for example, using differential privacy like in the research of Narayan et al. [20].

Research and development for the joining of databases in a secure way is needed to improve the

preservation of privacy of the subjects. For non-encrypted databases there are record linkage techniques that can match records of the same individual from different parties. To preserve the privacy for the subjects in the process of matching records, privacy-preserving record linkage (PPRL) techniques were developed. See Section 2.1 for a variety of such techniques.

While these solutions seem to be the future of data analysis, PPRL techniques struggle to have strong privacy guarantees while being scalable as well. For regular record linkage, so-called blocking techniques were introduced to decrease runtime and therefore increase scalability.

Blocking is a common step in the record linking process as it decreases the complexity by only comparing records that are put in the same blocks from the different databases. Here, blocks are a subset of the data having a similarity in attributes. Only comparing these blocks instead of all the data decreases the runtime of record linkage techniques significantly.

For PPRL however, blocking techniques also need to preserve privacy. Research has delivered privacy-preserving blocking techniques, but there are PPRL techniques that do not utilize those. For instance, Stammler et al. [27] elaborately describe the importance of the privacy guarantees their current work, MainSEL, gives. They argue that blocking techniques would compromise the strong privacy guarantees they now have. For example, the widely used LSH blocking techniques are disproved by He et al. [8] as the record linkage over LSH blocks is not IND-S2PC secure as discussed in Section 4. He et al. also define differential privacy definitions that relate to IND-S2PC. These are used to prove that there are differential blocking techniques that do not satisfy these definitions and therefore also not satisfy IND-S2PC. To preserve the strong privacy guarantees, MainSEL chose to not implement any blocking technique.

Instead of blocking, MainSEL provides PPRL using secure multiparty computation (SMPC). SMPC is a protocol that aims to eliminate a third party (TP) and instead relies on the collaboration of involved parties to conclude desired results, in this case; match records. SMPC ensures that this is done in a secure way, i.e. no information of a party is leaked to another. MainSEL makes use of a Secure EpiLinker which uses SMPC to guarantee privacy and eliminate the TP that was used in Mainzelliste [16], the software MainSEL is based on.

Recently however, Rohde et al. [24] applied blocking techniques on the current Mainzelliste software. They conclude that this appliance resulted in an improvement of runtime "by orders of magnitude". These improvements are more than noteworthy and, seeing as the runtimes of MainSEL [27] are of proportions that are unusable in real use cases, research into privacy-preserving blocking techniques applicable for MainSEL and other PPRL techniques is of great value. For this research, the following general research questions were formulated:

> ***RQ 1*** *Which, if any, blocking techniques adhere to the strong privacy guarantees of privacy-preserving record linkage?*
>
> ***RQ 2*** *What are the strong privacy guarantees of privacy-preserving record linkage?*
>
> ***RQ 3*** *What information can a malicious entity infer or extract from a blocking technique?*

In this work there will be an in-depth look into privacy-preserving blocking techniques and whether they are designed properly according to the strict privacy-preserving definition, IND-S2PC. He et al. verified that LSH-based blocking in general does not satisfy IND-S2PC. In this work it is shown that blocking techniques other than LSH-based ones can be proven to be incompliant with IND-S2PC. Because IND-S2PC is only applicable for two-party settings, we extend the definition of IND-S2PC into IND-S3PC, meaning the indistinguishability in secure three-party computation. With IND-S3PC, blocking techniques that make use of a TP can be tested for compliance. In this work it is shown that an existing privacy-preserving blocking technique with a TP does not comply with IND-S3PC. In another privacy-preserving blocking technique the introduction of a TP provides a solution for the problem privacy-preserving blocking techniques have with achieving IND-S2PC. This is an LSH-based blocking

technique and it is shown to satisfy IND-S3PC. Furthermore, a new definition for IND-SMPC, meaning the indistinguishability in secure multi-party computation, is formulated. However applying this is left up to future work. Summing up, this research includes the following contributions:

- Propose a new privacy definition, IND-S3PC, for privacy-preserving blocking techniques use a third party for certain computations within the protocol.
- Propose a new privacy definition, IND-SMPC, for privacy-preserving blocking techniques between multiple parties.
- Give an overview and insight in existing privacy-preserving blocking techniques from which many intuitively do not satisfy IND-S2PC and IND-S3PC.
- Prove that the LSH-based blocking technique with homomorphic matching by Karapiperis et al. [13] does satisfy IND-S3PC.
- Prove that the canopy-based blocking technique by Shu et al. [26] does not satisfy IND-S3PC.

The layout of this work is as follows. First general knowledge about the main components for the research, PPRL, SMPC and blocking will be elaborated upon. Section 4 describes the privacy definition IND-S2PC and shows with an example how existing privacy-preserving blocking techniques do not satisfy this definition. This is followed by this research' contribution, IND-S3PC and IND-SMPC, in Section 5. The new IND-S3PC is applied on two blocking techniques that are said to be privacy-preserving in Section 6.1 and 6.2. It is shown in Section 6.1 that this is true according to the IND-S3PC definition, and false for the blocking technique in Section 6.2. Finally, the results and future work are discussed in Section 8.

2 Background

Before diving into privacy-preserving blocking, an understanding of PPRL is needed. The privacy guarantees that PPRL gives needs to be ensured in the blocking, so it is important to keep PPRL in mind. Next SMPC is explained as it is used in many PPRL and blocking techniques. Furthermore, it can also serve as substitute for a TP. This will be important in the discussions later. Last will be an overview of a broad variety of (privacy-preserving) blocking techniques. An introduction is given by explaining standard blocking and its various features. Anonymity within blocking is explained with a blocking technique of Han et al. in Section 2.3.2. LSH-based blocking is explained in Section 2.3.3. LSH is later used as an example for the problem this work is trying to solve. This section will conclude with a summary of some other interesting privacy-preserving blocking techniques.

2.1 Privacy-Preserving Record Linkage

Record linkage [6] is the process of matching different records of the same individual. Most of the time this is done across different databases where identification for the same person differs. Record linkage relies on PII to match records. For example, two records containing the same date of birth and address are most likely to be from the same individual. There are many record linkage techniques that are state of the art, examples and comparisons are described by Karr et al. [14].

To protect the PII and thereby the privacy of the subjects, privacy enhancing techniques have been developed to protect the databases as well as their accompanying actions. As record linkage uses PII, multiple PPRL techniques have emerged which guarantee the privacy of subjects in multiple databases whilst matching records.

There exist various takes on how PPRL can be implemented, all with their respective privacy guarantees. The matching of personal identifiers using HMAC [1] is regarded as the simplest solution. Two parties would submit their personal identifiers from their database, hashed by HMAC, to a TP which would link the encrypted identifiers and send the result back. This is very simple, however, requires no faulty data and therefore, does not work in practice. More advanced methods like bloom filters (BF), the

Dice-coefficient and locality sensitive hashing (LSH) are commonly used in PPRL [25] [18] [13], all which incorporate fault tolerance techniques and offer scalability and privacy guarantees. However, in these cases there is always a trade-off between privacy and scalability, meaning that better privacy guarantees usually means higher computational costs as well. Especially in cases where the TP is omitted, it is costly to preserve the privacy between the participating parties. Furthermore, most privacy-preserving blocking papers do tend to skip defining privacy formally and therefore miss definite proof of why their protocol is privacy preserving, making them less credible. A further discussion of this can be found in Section 8.

2.2 Secure Multi-Party Computation

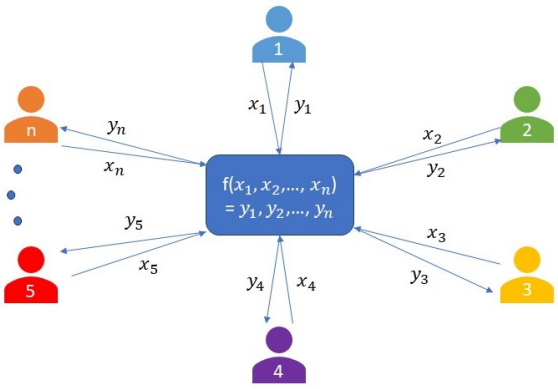

Figure 1: General visualization of secure multiparty computation, image from [19].

SMPC is a vital component in many of the PPRL and privacy-preserving blocking techniques. It originated from the work of Yao [30] in the 80s and has since developed. It has been applied in various appliances solving privacy issues, including record linkage and blocking. For example, Laud et al. [17] used SMPC to match records for healthcare analysis improvements.

SMPC addresses the problem of needing a TP for computation purposes. Many privacy-preserving techniques rely on a TP to do the computations and send results to the relevant parties. One can argue that such a TP may not be trustworthy at all or is another possible liability in preserving privacy. Instead, SMPC relies on the fact that no information can be inferred from the parties involved who, only by working together, can compute results among themselves. Therefore, SMPC allows for participating parties to compute results amongst themselves without the need of an external entity.

Visualized in Figure 1, SMPC generally works as follows; there is a need for some form of computation on multiple inputs owned by different parties. These parties want to keep their own information private but still wish to compute a result together. SMPC allows these parties to submit their input and compute their output without inferring any information about the other parties' input. There are also cases where SMPC restricts the result finding to be only possible if at least a certain number of the participating parties, not all, respond to the query.

2.3 Blocking Techniques

This section will describe a wide range of approaches of blocking techniques. Before the privacy-preserving blocking techniques are explained, the standard blocking techniques will be laid out. Standard blocking techniques substantiate the work of many privacy-preserving blocking techniques and are therefore a good introduction to further understand blocking and the challenges when introducing privacy.

2.3.1 Standard Blocking

The problem blocking addresses is the scalability of record linkage. Matching records between databases normally requires all records of all databases to be compared, which is impractical. Blocking provides a way for record linkage to only compare potential matches that the blocking technique deemed similar in some way. Blocking divides the database in blocks where each block consists of records with a matching property. For example, blocks are made based on the first letter of the last name resulting in 26 blocks. Matching records means that record

linkage only looks at blocks with the same property, e.g. blocks with letter 'a' would only be compared with each other, reducing the number of comparisons record linkage makes.

This basic technique has been improved in various ways. Steorts et al [28] and O'Hare et al. [21] describe basic traditional blocking techniques and more recent blocking techniques that are based on clustering algorithms such as the k-nearest neighbour algorithm. The algorithm is based on clustering records based on distances and set threshold. This will result in blocks where the distance between records is smaller than in other blocks therefore making it probable that matching records reside within similar blocks.

There is also a difference between supervised and unsupervised blocking. Blocking techniques require some form of ruling or featuring on which the blocking can be based, e.g. the first letter of the last name. Choosing the optimal ruling is a challenge on its own and requires knowledge about the blocking technique used and the data itself. Supervised blocking requires such expertise. Unsupervised blocking however can decide on the ruling itself, making it attractive for easy deployment as no expertise on both the type of data and the blocking technique is needed. O'Hare et al [21] also review some unsupervised blocking techniques while also introducing their own in [22] which does not require labeled data and manual fine-tuning and still outperforms other techniques.

2.3.2 K-anonymity Blocking

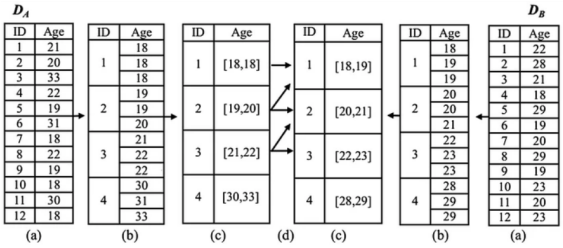

Figure 2: Example of K-anonymity blocking by Han et al. [7].

Han et al. [7] introduced a novel blocking technique where the blocks are k-anonymous, meaning that every record in the block has k-1 with the same blocking key value. They solely use numerical attributes from the PII for blocking. A figure of an example from their paper is shown in Figure 2. Depicted is the process of blocking database A (D_A) from left to right, and database B (D_B) from right to left. Here age is chosen as identifier (named blocking key by Han et al.) and the sorted values are put into blocks. The blocks in the figure are k-anonymous with k=3. Next, representative values describing the different blocks are made like such; [x,y] with x and y representing the lower and upper bound of the block's domain. For age it would be [18,19] for one block, containing the records with age 18 till 19, like in Figure 2c of D_B. To safely check for similarity, the representative values are encrypted using Paillier. Because of Paillier's homomorphic property, subtractions under encryption are possible. Therefore it is possible to compute similarity securely according to the similarity method explained by Han et al. [7]. Simply, these are computed by subtracting two values and checked on whether the result is positive. The so-called Decision Unit, which distributes public keys and has the private key, will decrypt these encrypted results of subtractions, check whether the result is positive and give back whether blocks match or not.

The privacy guarantees are based on HBC parties. Paillier cryptosystem is semantically secure and even when information about the plaintext is revealed, it is k-anonymous. However, the Decision Unit is a TP and introduces the discussion whether a TP is desired in such a protocol. This discussion will be elaborated upon later as the two main privacy-preserving blocking techniques discussed in Section 6.1 and 6.2 also include a TP.

2.3.3 Locality Sensitive Hashing

LSH-based blocking is currently one of the most common blocking technique used when applying blocking in a PPRL protocol. Standard LSH-based blocking puts the hashes of similar data into the same blocks (or bins or buckets), thereby significantly reducing the dimensions of the data. Note that all data of the records is used when hashing, not just the significant PII attributes. Karapiperis et al. [13] state that there

are three main LSH families which are used to create hashes. These are Jacard, Euclidean and Hamming LSH. All create the hash families differently, but the result will always be blocks containing hashes of similar data.

Similar data is hashed in the same block because LSH is designed to result in collisions of hashes. That is, the smaller the distance between two records, the higher the probability that the resulting hash by a certain hash family is the same. Because these records are hashed, this is as secure as the used hash family. LSH families are not known to be cryptographically secure, e.g., they are designed to be not collision resistant which is typically desired of a hash family. However, this property is essential for LSH to work like it does for blocking, collisions are the actual blocks.

The hashed records of the same blocks over different databases still have to be compared. Representing the records as Bloom filters and computing the distances based on the hash family gives a high probability of matching records. The computation itself can be done either by a TP or by applying SMPC. For the latter the distances can then be computed using the homomorphic property of e.g. Paillier cryptosystem. Other methods include k-means LSH, nearest-neighbour lookup on LSH blocks, transitive LSH, which all utilize the LSH blocks to compute high probability matching record pairs.

2.3.4 Other techniques

Benkhaled et al. [2] propose their novel K-Modes algorithm as an extension upon clustering technique K-Means. Instead of needing numerical data, like the K-Means algorithm, K-Modes can handle categorical data. This seems promising for blocking, however they do not consider privacy in their approach.

Karakasidis et al. [11] propose a secure record linkage technique consisting of three steps. The data is first encoded into phonetics using multiple phonetic algorithms to decrease missing matches. Fake phonetics are injected into the data and finally all are securely hashed. The preparation of using phonetics seems interesting and is maybe a viable addition to other private blocking techniques.

There are also techniques that improve upon existing blocking techniques. For example, Multi-Sampling Transitive Closure for Encrypted Fields (MS-TCEF) applies scalability to blocking techniques [10]. They state that linear complexity is achievable by using MS-TCEF and their own Sorted Neighbourhood blocking [12]. They also state that another blocking technique can be used instead of the Sorted Neighbourhood blocking, MS-TCEF will remove redundancy in blocks and improve the fault tolerance for compatible blocking techniques.

Finally, Vatsalan et al. [29] created a blocking technique based on signatures and phonetics, however, it is less accurate than LSH-based blocking. Kuzu et al. [15] describe their differential blocking technique with controlled data leakage. Ranbaduge et al. [23] present their hashing-based blocking technique stating its guarantee for scalability and privacy.

3 Notations

Table 1 explains all notations used in proofs hereafter. All parties are considered to be honest but curious (HBC) unless said otherwise. HBC means the parties follow the protocol truthfully but want to collect any information about that can be inferred from other parties.

Table 1: Notations

A	party A or Alice
B	party B or Bob
C	party C, TP or Charlie
$\mathcal{N}$	set of all participating parties
$H^j(x)$	record x hashed by a composite hash function j consisting of base hash functions of certain family H
D_A	dataset from Alice
$B_i(D_A)$	blocks generated from D_A with i denoting the block
B^S	a blocking strategy
T_A^j	blocking group of Alice consisting of $H^j(x)$ with the same hash result
S_A	subset id-pairs for Alice
I_A	$\{\text{IDs} \in D_A \wedge \text{IDs} \in S_B\}$
$\tilde{I_A}$	homomorphically encrypted records referenced by I_A
$\tilde{a}, \tilde{b}, \tilde{c}$	homomorphically encrypted intermediate value
M	all possible record pairs
$\|x\|$	size of any variable x
π_1	LSH blocking with homomorphic SMPC matching protocol
π_2	Canopy-based blocking protocol
k	security parameter used in definitions, parameterizes sizes, outputs and views, which are all polynomial in k
$negl(k)$	Any function over security parameter k that is negligible

4 IND-S2PC Definition And Application

Following is the privacy definition of IND-S2PC. This definition will be applied on LSH-based blocking to introduce the problem privacy-preserving blocking techniques have satisfying IND-S2PC. This is based on the work by He et al. [8]. First let's define IND-S2PC.

Consider two HBC holders of datasets, Alice and Bob, who want to match records without leaking any information other than the cardinality of their dataset and their matched records. The protocol π to match the records satisfies IND-S2PC when it is a two-party protocol that computes function f $f(D_A, D_B) = f(D_A, D'_B)$ for any D_A and D_B D'_B pair. This means given D_A, Alice cannot distinguish D_B from D'_B and therefore cannot obtain any information about the datasets from Bob.

Definition 4.1. Indistinguishability over the two-party setting
The following definition is for an adversary A, without loss of generality as A and B are interchangeable. For any probabilistic polynomial adversary T the following definition holds:

$$\begin{aligned} &Pr[T(view_A^\pi(D_A, D_B)) = 1] \\ \leq &Pr[T(view_A^\pi(D_A, D'_B)) = 1] + negl(k) \end{aligned}$$

According to He et al.[8], this defines that an adversary A has no advantage in guessing which dataset is which, D_B or D'_B, in other words, they are computationally indistinguishable. This assumes that any function over k, $negl(k)$, is computationally bound and therefore does not improve the chances of adversary A.

4.1 IND-S2PC On LSH-Based Blocking

Why LSH-based blocking techniques are proven not to satisfy IND-S2PC is explained. Again consider D_A and D_B, D'_B pair where $f(D_A, D_B) = f(D_A, D'_B)$. Consider the difference between D_B and D'_B to be one record, let record b in D_B and record b' in D'_B and let the distance between the records b and b' be big enough to have their hashes be different (with high probability). Therefore, b and b' are put into different blocks. The cardinality of the blocks that b and b' are put in will both equal to 1 as the rest of the dataset remains the same, so $|B_b(D_B)| - |B_b(D'_B)| = 1$, which is the cardinality of block B_b in D_B - cardinality of block B_b in D'_B, and $|B_{b'}(D'_B)| - |B_{b'}(D_B)| = 1$, which is the cardinality of block $B_{b'}$ in D'_B - cardinality of block $B_{b'}$ in D_B. Alice as adversary can now set D_A such that her blocks B_b and $B_{b'}$ in D_A differ in cardinalities and Equation 1 can be used accordingly in Equation 2. Equation 1 is a calculation

of the cost of matching blocks. I.e. the number of potential matches found when comparing all blocks.

$$cost_{B^S}(D_A, D_B) = \sum_{i,j \in B^S} |B_i(D_A)||B_j(D_B)| \quad (1)$$

The equation is filled in below according to the explanation above.

$$\begin{aligned}
&cost_{B^S}(D_A, D_B) - cost_{B^S}(D_A, D'_B) \\
=&|B_b(D_A)||B_b(D_B)| + |B_{b'}(D_A)||B_{b'}(D_B)| - \\
&|B_b(D_A)||B_b(D'_B)| + |B_{b'}(D_A)||B_{b'}(D'_B)| \\
=&|B_b(D_A)| \cdot (|B_b(D_B)| - |B_b(D'_B)|) + \\
&|B_{b'}(D_A)| \cdot (|B_{b'}(D_B)| - |B_{b'}(D'_B)|) \\
=&|B_b(D_A)| + |B_{b'}(D_A)| \quad (2)
\end{aligned}$$

because $|B_b(D_B)| - |B_b(D'_B)| = 1$
and $|B_{b'}(D'_B)| - |B_{b'}(D_B)| = 1$

therefore: $|B_b(D_A)| - |B_{b'}(D_A)| \neq 0$

Alice set D_A such that her own blocks differ in cardinality. Furthermore, the cost (see Equation 1) now completely depends on the dataset of Alice, as can be seen in Equation 2 ($|B_b(D_A)| + |B_{b'}(D_A)|$). Because Alice's block cardinalities differ, the last inequality in Equation 2 holds and Alice can distinguish D_B and D'_B. This proves that the blocking strategy using LSH blocking does not satisfy IND-S2PC. Other blocking techniques can also be shown to not satisfy IND-S2PC in a similar way. That is, if the block cardinality is revealed, it is likely that the protocol does not satisfy IND-S2PC.

5 IND-S3PC And IND-SMPC Definitions

Following are the new privacy definitions, IND-S3PC and IND-SMPC. These are part of the contributions of this research. They build on the IND-S2PC definition for IND-S3PC to be applied in a three-party setting and IND-SMPC to be applied in a multi-party setting.

5.1 IND-S3PC

There are many blocking techniques in which the protocol makes use of a TP. To verify that these are privacy-preserving, a new definition is needed. Extending the IND-S2PC definition, IND-S3PC (indistinguishability over secure three-party computation) can be defined. This definition is for protocols involving two parties that use a TP, a semi-trusted helper party, solely for computing purposes, like distribution parameters required for the protocol or the computation of distances between potential record matches. The TP is considered semi-trusted to be not colluding with either Alice or Bob.

It then follows that for two HBC parties Alice and Bob with respective datasets have to be indistinguishable in each simulated outcome of the blocking protocol like Definition 4.1. The TP may not infer any information about the contents of the dataset, blocks and records. Only the cardinalities of the blocks and dataset may be known by the TP. Kamara et al. [9] formalize a definition of a non-colluding party (the TP) together with HBC parties (Alice and Bob) in their work in Section 4.1.

Because the TP knows the cardinalities of the blocks, the problem of Section 4.1 persists if the TP is the adversary. IND-S3PC defines that this is considered acceptable. In other words, IND-S2PC implies IND-S3PC but IND-S3PC does not necessarily imply IND-S2PC. IND-S3PC is therefore a weaker privacy definition than IND-S2PC. However, by defining IND-S3PC this way, the issues laid out in Section 4.1 are prevented for the participating parties with the actual datasets. Section 7 further discusses the pros and cons of using the TP like this, for now IND-S3PC is considered privacy-preserving as the views of Alice and Bob after executing the protocol satisfy IND-S2PC.

5.2 IND-SMPC

For a multi-party setting, where one party wants to link records with multiple other datasets, the privacy-preserving blocking technique should satisfy IND-SMPC.

Definition 5.1. Indistinguishability over the multi-party setting
Given $\mathcal{N}$, the set of all the participating parties, IND-SMPC implies that the view yielded from executing protocol π over datasets D_i and D_j is indistinguishable from the view yielded from executing protocol π over datasets D_i and D_l. Note that party $i \in \mathcal{N}$ is fixed without loss of generality and it holds for all distinct $j, l \in \mathcal{N}$ where $i \neq j \neq l$. Naturally, π should also satisfy IND-S2PC. For any probabilistic polynomial adversary T the following definition holds:

$$\begin{aligned} &Pr[T(view_i^{\pi}(D_i, D_j)) = 1] \\ \leq &Pr[T(view_i^{\pi}(D_i, D_l)) = 1] + negl(k) \end{aligned}$$

Note that this definition is not further applied in this work. It is left up to future work to verify privacy-preserving multi-party blocking satisfy IND-SMPC.

6 IND-S3PC On Existing Blocking Techniques

Now that IND-S3PC is defined, two privacy-preserving blocking techniques are evaluated on whether they satisfy IND-S3PC. The first is an LSH-based blocking protocol with homomorphic SMPC matching, the second a canopy-based protocol. Both make use of a TP, be it in very different ways. It is shown that the LSH-based blocking technique, which was previously deemed not to satisfy IND-S2PC, does satisfy IND-S3PC. It is also shown that the canopy-based blocking technique does not satisfy IND-S3PC.

6.1 LSH Blocking With Homomorphic SMPC Matching

An LSH-based blocking protocol from the work of Karapiperis et al. [13]. The protocol has the option for regular two-party matching or matching through SMPC involving a TP. They evaluate upon three matching techniques, or distance metrics, which in turn are applicable to three types of hash families. The Min-Hash family is sensitive to the Jaccard metric, Hamming family the Hamming distance and the p-stable distributions-based family the Euclidean metric. The conclusion indicates that the Hamming family outperforms the other two. Therefore we will consider this method from now on.

Consider LSH hash family H where H^H is sensitive to the Hamming distance. Because of the properties of LSH, it allows for hashes of similar records to be categorized, thus creating blocks. A block consists of the IDs and hashed records. The latter can be compared to another party's records from the same block after which ID pairs can be found representing the matched records.

The two-party setting has already proven to be not secure as described in Section 4. However, Karapiperis et al. also introduce a three-party setting involving a TP. The order of messages and views of Alice, Bob and Charlie can be seen in Figure 3 and Equations 3, 4.

$$\begin{aligned} view_A^{\pi_1} : &① \; C \to A : H^j \\ &② \; A \to C : T_A^j \\ &③ \; C \to A : S_A, I_A \\ &④ \; A \to B : \tilde{I_A} \\ &④ \; B \to A : \tilde{I_B} \\ &⑤ \; A \to C : \tilde{a_A}, \tilde{b_A}, \tilde{c_A} \\ &⑥ \; C \to A : M \end{aligned} \tag{3}$$

$$\begin{aligned} view_C^{\pi_1} : &① \; C \to A, B : H^j \\ &② \; A, B \to C : T_A^j, T_B^j \\ &③ \; C \to A, B : S_A, I_A, S_B, I_B \\ &⑤ \; A, B \to C : \tilde{a_A}, \tilde{b_A}, \tilde{c_A}, \tilde{a_B}, \tilde{b_B}, \tilde{c_B} \\ &⑥ \; C \to A, B : M \end{aligned} \tag{4}$$

The circled numbers refer to the numbers used in Figure 3. The protocol is as follows, Alice and Bob block their dataset with λ composite hash functions from LSH family, so each record is hashed λ times and put into the respective blocking group T^j with j representing which composite hash function was used to create that blocking group. They now have j blocking groups consisting of blocks represented by the hashes

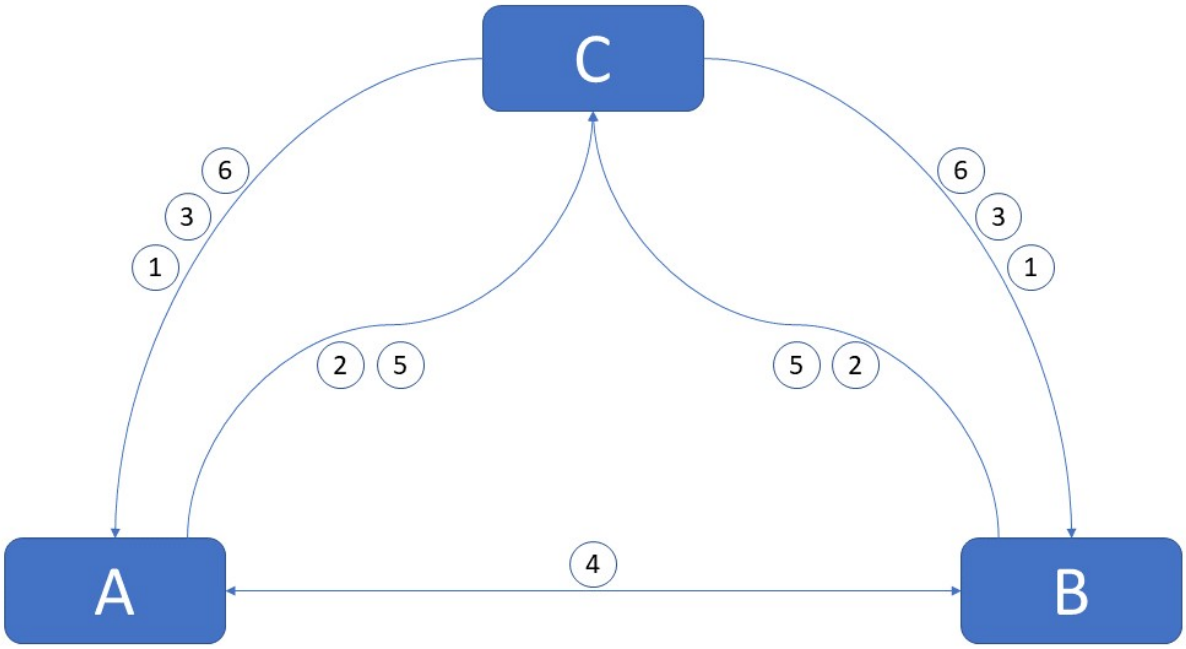

Figure 3: Homomorphic LSH blocking protocol [13].

computed. Note that because of the properties of LSH, similar data is hashed into the same value with high probability. They send their blocking groups to Charlie (2). Charlie will form candidate pairs of possible matching records by matching the same blocks, removing duplicate candidate pairs, and sends a subset of those to Alice and the remaining to Bob, S_A and S_B. Both will also receive a list with IDs that reference records in their own dataset as well as being part of an ID pair in the subset of the other, I_A and I_B (3).

$$S_A = \text{list of ID pairs sent to Alice}$$
$$I_A = \{IDs \in D_A \wedge IDs \in S_B\}$$

Both parties will encrypt their own records referenced by I following the Paillier encryption method, into BFs, and send them to the other party (4). Because of the homomorphic property, both parties can compute intermediate distance values between their own records and the received encrypted records in S. Intermediate values a, b, c have: a: number of bit positions in the BFs where both are 1, b: number of bit positions in the BFs where in BF_B it is 1 and in BF_A it is 0, c: number of bit positions in the BFs where in BF_A it is 1 and in BF_B it is 0, or number of bit positions in BF_A it is 1 minus a.

$$BF_A : [1,1,0,1,0,1,1]$$
$$\tilde{BF_A} : Enc([1,1,0,1,0,1,1])$$
$$\tilde{BF_B} : Enc([1,0,1,1,0,1,0])$$
$$\tilde{a} : 1+0+1+1+0 = 3$$
$$\tilde{b} : 1+0 = 1$$
$$\tilde{c} : 5 + a * -1 = 2$$

The results, also encrypted, will be sent to Charlie (5) who can decrypt and compute the actual distance between the record pairs. The Hamming distance is $b + c$, the difference in bits, and is therefore $1 + 2 = 3$ in the example above. Only Charlie can see this end result as the encrypted result can only

be decrypted by the key held by Charlie. If this distance is smaller than a certain threshold, Charlie will deem them matched. This is done for every potential match and Charlie will give back the actual matches to the dataset owners (6) for them to continue with the PPRL protocol.

6.1.1 Privacy Analysis

Regarding the leakage described by He et al. which said that LSH is not IND-S2PC secure because of the leakage of cardinalities of blocks; this is not directly applicable in the three-party setting. However, considering that Charlie and Alice are colluding, the same leakage can be proven again. Charlie still receives all blocks from both parties, therefore, the cardinalities are known to Charlie. With the help of Alice, one can apply the same method as described in Section 4.1 and distinguish D_B from D'_B. This however, is only possible when the TP colludes with another party as other leaks are actually prevented by this protocol.

The main reason for the leakage of LSH blocking protocols is because the cardinalities of the blocks give away a distinction between two datasets. If the three parties are considered to be HBC, Alice and Bob should not be able to infer anything about the cardinalities of the other party's blocks. Furthermore, Charlie should not be able to infer anything about the contents of the blocks.

Karapiperis et al. constructed the protocol to have the third party distribute the potential matches across the two data owners evenly. These potential matches are drawn from all the blocking groups. Because Alice and Bob have no idea how the potential matches are distributed, they have no way of determining the other one's blocks' cardinalities. Because of the redundancy, records should be hashed λ number of times in λ blocking groups. A record can end up in the same block across numerous blocking groups. This does not necessarily have to be the case, a record can also be hashed in all different blocks across the blocking groups. However, LSH makes this scenario less feasible. In both cases there can be no information gained about the other blocks' cardinalities. Both parties get a list of half the potential matches unknowing which potential match originated from which block or blocking group.

This specific step of distributing potential matches instead of giving back the matched blocks makes it IND-S3PC secure. The randomness in distribution and redundancy in blocks ensures IND-S3PC because the TP infers nothing about the datasets except for block cardinalities, and both parties infer no information except for matched record IDs of the other dataset.

6.2 Canopy-based Blocking

The canopy-based blocking technique makes use of canopy clusters from which 1 or more clusters form a block. The order of messages and views of Alice, Bob and Charlie can be seen in Figure 4 and Equations 5, 6. The circled numbers compare to the numbers used in Figure 4. Consider the two parties Alice and Bob. They first decide on the parameters that will be used and some public reference dataset (1). By using the same parameters and dataset to form canopies (2), they will end up with the same canopies. These are now individually filled with the data of their own dataset (3). This is done by measuring the Jaccard distance between the record's blocking attribute values and the canopy centre's reference attribute values. The latter is a result of steps (1) (2) and therefore the same for both parties. Depending on the distance and two thresholds, T_1 and T_2, the record is assigned to a or multiple canopies. Records from each canopy form a block which can be identified by the canopy ID.

$$\begin{aligned} view_A^{\pi_2} : & (5)\ A \rightarrow C : B_i(D_A) \\ & (7)\ C \rightarrow A : M \end{aligned} \tag{5}$$

$$\begin{aligned} view_C^{\pi_2} : & (5)\ A, B \rightarrow C : B_i(D_A), B_i(D_B) \\ & (7)\ C \rightarrow A, B : M \end{aligned} \tag{6}$$

For further redundancy in blocks, the blocks are merged when they are too small (4). This is determined by k, here the minimum block cardinality. IDs representing the blocks will be merged as well such

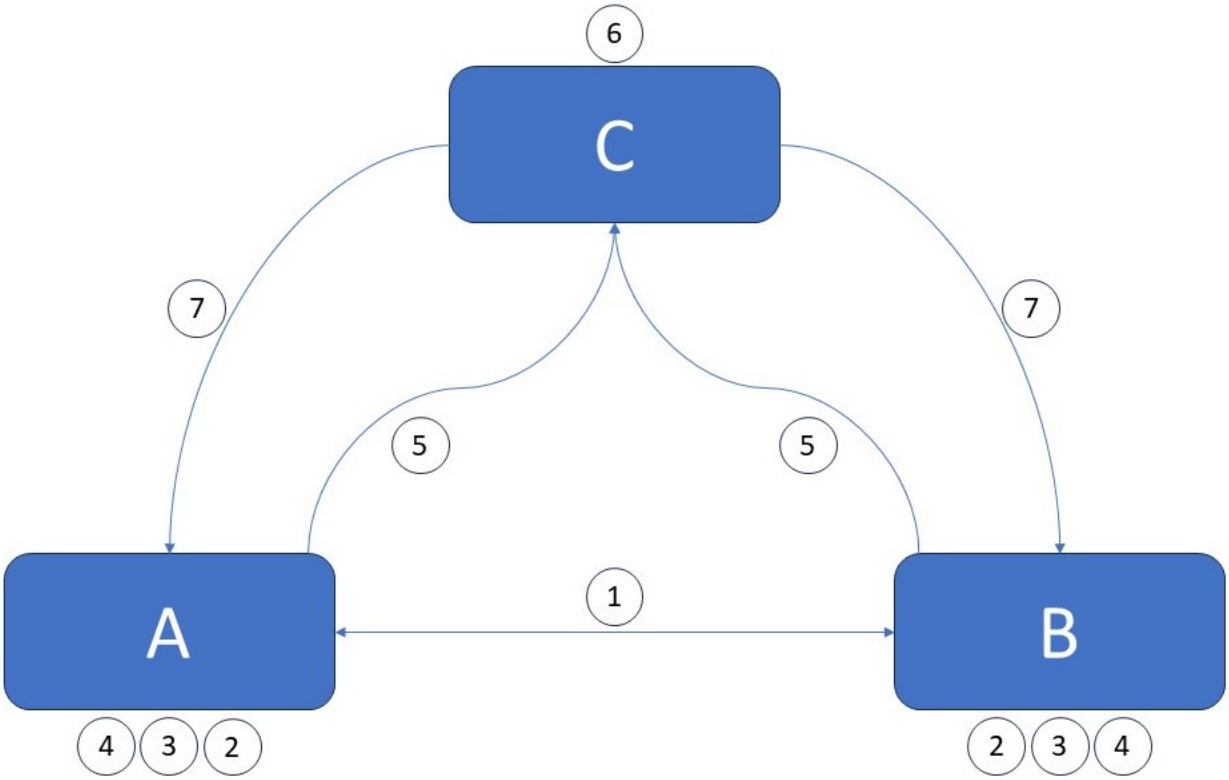

Figure 4: Canopy-based blocking protocol [26].

that the resulting block is represented by all initial block IDs. For this, the Jaccard distance is again used to compute distances between blocks based on their q-gram representations. The number of q-grams is the average number of q-grams among the blocking attribute values. If the frequency of a q-gram is the same as another it is ordered alphabetically. After merging, the blocks are sent to the Charlie (5). Charlie will generate ID pairs out of both Alice's and Bob's merged blocks (6). These are sent back to Alice and Bob (7) and they can then use PPRL only on the necessary candidate pairs.

6.2.1 Privacy Analysis

The protocol seems to be secure, but the block cardinalities are leaked to both Alice and Bob. Consider D_A and D_B, D'_B where $f(D_A, D_B) = f(D_A, D'_B)$ and let the distance between the differences b and b' of D_B and D'_B be big enough to have them be placed in different canopies, or blocks. Thus $|B_b(D_B)| - |B_b(D'_B)| = 1$ and $|B_{b'}(D'_B)| - |B_{b'}(D_B)| = 1$. Alice as adversary can now set D_A such that her own blocks differ in cardinality, just as long as it is larger than the predetermined minimum block cardinality k. Charlie will generate record pairs and send those to Alice and Bob. Alice can now derive the costs of linking each block and conclude that the cost of linking with D_B and linking with D'_B is different due to the 1 difference in $B_b(D_A)$ and $B_{b'}(D_A)$. Alice can therefore distinguish between D_B and D'_B, hence the canopy-based blocking technique does not satisfy IND-S3PC.

Merging the blocks helps the protocol achieve a

higher performance on matching records. It also triggers a potential solution for achieving IND-S3PC. Now the protocol has a minimum block cardinality, but consider that the blocks have a fixed block cardinality. If every block is padded with some random data from the rest of the dataset such that every block has the same cardinality, there is no information to be gained by the adversary. However, the protocol would lose some performance. This intuition is left up for future work.

7 Discussion

Following is a summary of the findings and contributions of this work as well as limitations, discussion of trade-offs and possible future work.

The study on privacy-preserving blocking techniques and their respective privacy guarantees yielded some negative results. Proofs and validation of whether the technique was privacy-preserving, was mostly insufficient. Privacy definitions were missing and most papers only claim that the blocking technique is privacy-preserving without giving the 'why'. Furthermore, the issue of unintentionally revealing the cardinality of a block to other parties remains in many "privacy-preserving" blocking techniques, thereby failing to meet the requirements of IND-S2PC and our contribution, IND-S3PC.

To verify blocking techniques with a TP on their privacy preservation, we defined IND-S3PC. The new definition states that the cardinalities of the block may only be known by a TP. IND-S3PC is applied on two different privacy-preserving blocking techniques. The canopy-based blocking technique is shown to not satisfy IND-S3PC, but the LSH-based blocking technique with homomorphic matching is shown to do satisfy IND-S3PC.

This proves that LSH blocking protocols, which were initially deemed not privacy-preserving in Section 4, can still be utilised in a blocking protocol in combination with a computing TP. This triggers the question whether other existing blocking techniques can benefit of using a TP in their protocol or if existing techniques could be changed to satisfy IND-S3PC. For example, the discussed canopy-based blocking technique could be adapted to have a fixed block cardinality. This would make the revealed block cardinalities redundant for an adversary. Solutions to avoid the block cardinality leakage or to make the block cardinality redundant is left up to future work. For these it can be interesting to include an impact analysis on the efficiency of the protocol.

Open for discussion is whether a TP is an accepted entity in blocking protocols. The question arises whether a TP is trusted and, if it is, why does it not execute the entire blocking protocol by itself and just return the record pairs to the dataset owners. IND-S3PC defines it such that only the block cardinalities are entrusted to the TP, but one could argue more information may be inferred or given to the TP if it is trusted anyway. It is up to the user to make this decision.

If a TP is deemed not desirable, SMPC can be used to replace a TP. That is, blocking protocols could compute the parts in the protocol that leak the cardinality of blocks by SMPC thus not leaking the cardinalities of the blocks and therefore circumventing the problem these protocols have with satisfying IND-S2PC.

SMPC avoids the use of a TP but consequently triggers another trade-off. SMPC is often costly in computing. Research can be done on whether the use of SMPC within the blocking protocol is practically applicable. Especially the impact on the efficiency of the protocol.

8 Conclusion

In conclusion, this work builds on the two-party privacy definition IND-S2PC. We propose IND-S3PC for a three-party protocol where the third party is a computing party. We also propose IND-SMPC for a multi-party protocol.

We show that there are privacy-preserving blocking techniques that do not satisfy IND-S3PC and IND-S2PC. Nevertheless, one blocking technique is shown to be privacy-preserving according to the definition of IND-S3PC. Furthermore, privacy-preserving blocking techniques in the multi-party setting can be verified to satisfy IND-SMPC.

References

[1] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Annual international cryptology conference*, pages 1–15. Springer, 1996.

[2] H. N. Benkhaled, D. Berrabah, and F. Boufares. A novel approach to improve the record linkage process. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1504–1509. IEEE, 2019.

[3] J. Böhler and F. Kerschbaum. Secure multiparty computation of differentially private median. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 2147–2164, 2020.

[4] Breach Portal. `https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf`. Accessed: 2023-06-16.

[5] GDPR. `https://gdpr-info.eu/`. Accessed: 2023-06-16.

[6] R. Hall and S. E. Fienberg. Privacy-preserving record linkage. In J. Domingo-Ferrer and E. Magkos, editors, *Privacy in Statistical Databases*, pages 269–283, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[7] S. Han, D. Shen, T. Nie, Y. Kou, and G. Yu. Private blocking technique for multi-party privacy-preserving record linkage. *Data Science and Engineering*, 2(2):187–196, 2017.

[8] X. He, A. Machanavajjhala, C. Flynn, and D. Srivastava. Composing differential privacy and secure computation: A case study on scaling private record linkage. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1389–1406, 2017.

[9] S. Kamara, P. Mohassel, and M. Raykova. Outsourcing multi-party computation. *Cryptology ePrint Archive*, 2011.

[10] A. Karakasidis, G. Koloniari, and V. S. Verykios. Scalable blocking for privacy preserving record linkage. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 527–536, 2015.

[11] A. Karakasidis and V. S. Verykios. Secure blocking+ secure matching= secure record linkage. *Journal of Computing Science and Engineering*, 5(3):223–235, 2011.

[12] A. Karakasidis and V. S. Verykios. A sorted neighborhood approach to multidimensional privacy preserving blocking. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 937–944. IEEE, 2012.

[13] D. Karapiperis and V. S. Verykios. An lsh-based blocking approach with a homomorphic matching technique for privacy-preserving record linkage. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):909–921, 2014.

[14] A. F. Karr, M. T. Taylor, S. L. West, S. Setoguchi, T. D. Kou, T. Gerhard, and D. B. Horton. Comparing record linkage software programs and algorithms using real-world data. *PloS one*, 14(9), 2019.

[15] M. Kuzu, M. Kantarcioglu, A. Inan, E. Bertino, E. Durham, and B. Malin. Efficient privacy-aware record integration. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 167–178, 2013.

[16] M. Lablans, A. Borg, and F. Ückert. A restful interface to pseudonymization services in modern web applications. *BMC medical informatics and decision making*, 2015.

[17] P. Laud and A. Pankova. Privacy-preserving record linkage in large databases using secure multiparty computation. *BMC medical genomics*, 11(4):33–46, 2018.

[18] I. Lazrig, T. C. Ong, I. Ray, I. Ray, X. Jiang, and J. Vaidya. Privacy preserving probabilistic record linkage without trusted third party. In

2018 16th Annual Conference on Privacy, Security and Trust (PST), pages 1–10. IEEE, 2018.

[19] M. Lemus, M. F. Ramos, P. Yadav, N. A. Silva, N. J. Muga, A. Souto, N. Paunković, P. Mateus, and A. N. Pinto. Generation and distribution of quantum oblivious keys for secure multiparty computation. *Applied Sciences*, 10(12):4080, 2020.

[20] A. Narayan and A. Haeberlen. Djoin: Differentially private join queries over distributed databases. In *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, pages 149–162, 2012.

[21] K. O'Hare, A. Jurek-Loughrey, and C. de Campos. A review of unsupervised and semi-supervised blocking methods for record linkage. *Linking and Mining Heterogeneous and Multi-view Data*, pages 79–105, 2019.

[22] K. O'Hare, A. Jurek-Loughrey, and C. de Campos. An unsupervised blocking technique for more efficient record linkage. *Data & Knowledge Engineering*, 122:181–195, 2019.

[23] T. Ranbaduge, D. Vatsalan, P. Christen, and V. Verykios. Hashing-based distributed multi-party blocking for privacy-preserving record linkage. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 415–427. Springer, 2016.

[24] F. Rohde, M. Franke, Z. Sehili, M. Lablans, and E. Rahm. Optimization of the mainzelliste software for fast privacy-preserving record linkage. *Journal of Translational Medicine*, 19(1):1–12, 2021.

[25] R. Schnell, T. Bachteler, and J. Reiher. Privacy-preserving record linkage using bloom filters. *BMC medical informatics and decision making*, 9(1):1–11, 2009.

[26] Y. Shu, S. Hardy, and B. Thorne. Canopy-based private blocking. In *Australasian Conference on Data Mining*, pages 203–215. Springer, 2018.

[27] S. Stammler, T. Kussel, P. Schoppmann, F. Stampe, G. Tremper, S. Katzenbeisser, K. Hamacher, and M. Lablans. Mainzelliste secureepilinker (mainsel): Privacy-preserving record linkage using secure multi-party computation. *Bioinformatics*, 2020.

[28] R. C. Steorts, S. L. Ventura, M. Sadinle, and S. E. Fienberg. A comparison of blocking methods for record linkage. In J. Domingo-Ferrer, editor, *Privacy in Statistical Databases*, pages 253–268, Cham, 2014. Springer International Publishing.

[29] D. Vatsalan, J. Yu, B. Thorne, and W. Henecka. P-signature-based blocking to improve the scalability of privacy-preserving record linkage. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 35–51. Springer, 2020.

[30] A. C. Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.