GRAPH STRUCTURES IN PRIVACY-PRESERVING BIOMEDICAL ANALYSES

vom Fachbereich Physik
der Technischen Universität Darmstadt

zur Erlangung des Grades
Doctor rerum naturalium (Dr. rer. nat.)

genehmigte Dissertation von

TOBIAS RAPHAEL KUSSEL

Erstgutachter: Prof. Dr. Kay Hamacher
Zweitgutachter: Prof. Dr. Thomas Schneider

DARMSTADT 2022

This document was typeset with XeTEX and a modified version of Tufte-LaTeX. The bibliography was managed with Zotero and processed by Biblatex. Used fonts were Vollkorn (serif), Gillius ADF No2 (sans serif) and TeX Gyre Cursor (monospace). Figures and images were generated with Matplotlib, PGFPlots, Excalidraw, and Affinity Designer.

Kussel, Tobias: Graph Structures in Privacy-Preserving Biomedical Analyses
Darmstadt, Technische Universität Darmstadt,
Jahr der Veröffentlichung der Dissertation auf TUprints: 2022
URN: urn:nbn:de:tuda-tuprints-218285
Tag der Einreichung: 01.06.2022
Tag der mündlichen Prüfung: 20.07.2022

Abstract

Graph theory is one of the truly interdisciplinary fields of research. Not only does the application of graphs lead to new insights in a variety of disciplines—physics, biology, sociology, mathematics, and computer science—but all of these disciplines, with their different questions and perspectives, actively influence the development and research of graph theory itself. Being able to abstract interconnected and interdependent systems and describe them in a mathematically exact way offers a different way of describing the problem when facing a diverse array of questions. It favors finding certain "topology-based" approaches while simultaneously providing a methodological toolbox; especially statistical physics offers numerous examples for this.

In this thesis, I consider various research questions in the field of distributed biomedical analysis. In this context, the health data used pose a significant risk to the privacy of the individuals involved—still, many conditions are socially stigmatized, so their disclosure could lead to ostracization, occupational disadvantages, and even physical harm. This is reflected in legislation, which requires special protection for health data—and other data that carry the risk of identifying an individual.

In this dissertation, I present the design, implementation, and empirical analysis of methods and algorithms that are not only relevant for obtaining medical knowledge and developing new treatment methods, but at the same time provide an exceedingly high level of protection concerning sensitive data. The developed methods presented hereafter use the cryptographic techniques of *Secure Multi-Party Computation* (MPC), a class of cryptographic protocols that enables the joint computation over distributed datasets while simultaneously providing highest security guarantees regarding the secrecy of the (distributed) input data. Graphs play a central role in this context: the desired functionalities are mapped to Boolean or arithmetic *Directed Acyclic Graphs* (DAGs), where the nodes represent operations or interactive protocols.

However, not only do the methodology and techniques employed in this dissertation make use of approaches from statistical physics and graph theory—many of the medical questions directly relate to graph problems or lead to more efficient solutions when interpreted as graphs. Specifically, a method is developed to efficiently analyze epistasis relationships, i.e., non-linear gene-gene and gene-environment interactions, and the resulting gene regulatory networks in a privacy-preserving manner. To enable the processing of the required amounts of genetic data within feasible time scales, novel generally applicable MPC building blocks were developed. Furthermore, the graph optimization problem of kidney exchanges—the efficient and fair allocation of donors' kidneys for kidney transplantation—is explored. Since sensitive medical data are involved as well, these data protection considerations are of central interest here. In addition to the many boundary conditions to this problem, for example regarding the ro-

bustness of the solutions or the medical compatibility of organ donors and recipients, finding the global solution has been proven to be a problem with superpolynomial complexity. Hence, a locally optimal algorithm to solve this issue is presented in this work. Finally, the problem of probabilistic record linkage, which is highly relevant in medical research, is considered. Based on the personal data such as first name, last name, address, and birthday, the similarity between patients in one or more databases is to be assessed, with the goal of identifying duplicates even in the presence of noisy data, i.e., misspellings and interchanged identifiers. Via the path of graph representation and the associated search for approximated subgraph isomorphisms, record linkage is reduced to a (simpler) database problem with efficiently solvable algorithms. The developed framework *Mainzelliste Secure EpiLinker* (MainSEL) provides a means for secure, privacy-preserving probabilistic record linkage between different medical institutions. However, it is also capable of comparing other data types for their respective similarity. At the end of this dissertation, three promising but not yet extensively explored extensions are presented that allow the matching of pharmacologically relevant small molecule graphs, the search for similar patients in clinical databases for nonspecific diagnoses, or the identification of disrupted biological regulatory pathways by comparing transcriptome networks.

Zusammenfassung

Die Graphtheorie ist eines der wirklich interdisziplinären Forschungsfelder. Nicht nur führt die Anwendung von Graphen zu neuen Erkenntnissen in einer Vielzahl von Disziplinen – Physik, Biologie, Soziologie, Mathematik, Informatik –, sondern all diese Disziplinen wirken mit ihren unterschiedlichen Fragestellungen und Sichtweisen aktiv auf die Entwicklung und Erforschung der Graphtheorie selbst ein. Vernetzte und mit Interdependenzen behaftete Systeme abstrahieren und mathematisch exakt beschreiben zu können, bietet für viele Fragestellungen eine Art der Problembeschreibung, die das Finden gewisser »topologie-basierter« Lösungswege begünstigt und gleichzeitig einen methodischen Werkzeugkoffer zur Verfügung stellt – gerade die statistische Physik bietet zahlreiche Beispiele dafür.

In dieser Arbeit werden verschiedene Fragestellungen aus dem Bereich der verteilten Analysen der Biologie und Medizin betrachtet. Dabei stellen die verwendeten Gesundheitsdaten oft ein erhebliches Risiko für die Privatsphäre der Betroffenen dar – noch immer sind zahlreiche Erkrankungen gesellschaftlich stigmatisiert, sodass ein Bekanntwerden zu Ausgrenzung, beruflichen Nachteilen, bis hin zu körperlichen Gefährdungen führen könnte. Das wird auch von der Gesetzgebung widergespiegelt, die Gesundheitsdaten—und anderen Daten, die das Risiko der Identifikation eines Individuums bergen—einen besonderen Schutzbedarf attestiert.

Daher werden in dieser Dissertation Methoden und Algorithmen entworfen, implementiert und empirisch untersucht, die nicht nur für die Erlangung medizinischer Erkenntnisse und der Entwicklung neuer Behandlungsmethoden relevant sind, sondern die zeitgleich ein überaus hohes Datenschutzniveau bezüglich der sensiblen Daten aufweisen. Dazu bedienen sich die entwickelten Methoden der Techniken des kryptografischen Feldes der *Secure Multi-Party-Computation* (MPC) – einer Klasse kryptografischer Protokolle, die die gemeinsame Berechnung über verteilte Datenbestände ermöglicht und zeitgleich höchste Sicherheitsgarantien bezüglich der Geheimhaltung der (verteilten) Eingabedaten gibt. Dabei spielen Graphen eine zentrale Rolle: Die abzubildenden Funktionalitäten werden als Boolesche oder arithmetische ungerichtete azyklische Graphen repräsentiert, bei denen die Knoten Operationen oder interaktive Protokolle darstellen.

Doch nicht nur die Methodik und die eingesetzten Techniken in dieser Dissertation bedienen sich der Methoden der statistischen Physik und der Graphtheorie – viele der medizinischen Fragestellungen sind ganz direkt Graphprobleme oder führen über die Betrachtung als Graph zu einer effizienteren Problemdarstellung. So wurde ein Verfahren entwickelt, um Epistasisbeziehungen, also nichtlineare Gen-Gen- und Gen-Umgebungs-Interaktionen, sowie die daraus resultierenden Genregulationsnetzwerke effizient und die Privatsphäre schützend zu analysieren. Um die Verarbeitung der erforderlichen Mengen genetischer Daten in für den Einsatz akzeptablen Zeitskalen zu ermöglichen, wurden neue, all-

gemein verwendbare MPC-Bausteine entwickelt. Des Weiteren wurde das Graphoptimierungsproblem der effizienten und fairen Zuteilung von Spendernieren für Nierentransplantationen bearbeitet. Auch hier sind sensible medizinische Daten betroffen, sodass Datenschutzaspekte eine bedeutende Rolle spielen. Nicht nur bestehen bei dieser Fragestellung viele Randbedingungen, zum Beispiel bezüglich der Robustheit der Lösungen oder der medizinischen Kompatibilität von Organspendern und -empfängern sondern das Finden der globalen Lösung ist ein Problem mit superpolynomialer Komplexität, sodass in dieser Arbeit ein lokal optimaler Algorithmus vorgestellt wird. Schließlich wird das in der medizinischen Forschung hochrelevante Problem des probabilistischen Record-Linkage betrachtet. Dabei soll basierend auf personenbezogenen Daten wie Vor- und Nachname, Adresse und Geburtstag die Ähnlichkeit zwischen Patienten in einer oder mehreren Datenbanken beurteilt werden, mit dem Ziel, selbst bei »verrauschten« Daten, also Schreibfehlern und der Verwechslung von Kennzeichnern, Duplikate zu identifizieren. Über den Weg der Graphrepräsentation und die damit verbundene Suche nach approximierten Subgraph-Isomorphismen wird Record-Linkage auf ein (einfacheres) Datenbank-Problem mit effizient lösbaren Algorithmen zurückgeführt. Das entwickelte Framework *Mainzelliste Secure EpiLinker* (MainSEL) bietet eine Möglichkeit zum sicheren, die Privatsphäre schützenden probabilistischen Record-Linkage zwischen verschiedenen medizinischen Institutionen. Es ermöglicht zudem weitere Datentypen auf Ähnlichkeit zu vergleichen. Am Ende dieser Dissertation werden drei vielversprechende, jedoch noch nicht umfänglich erforschte Erweiterung vorgestellt, die den Abgleich von pharmakologisch relevanten kleinen Molekülgraphen, die Suche nach ähnlichen Patienten in klinischen Datenbanken für unspezifische Diagnosen oder die Identifikation von gestörten biologischen Regulationspfaden durch den Vergleich von Transkriptomnetzwerken erlauben.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1
Preface

Graph structures are present in many areas of scientific research. Ranging from pure mathematical problems over sociological community models and photonic networks to the dynamics of complex systems—for many research questions graph structures and interconnected information flow are useful methodologies (and epistemological models) to formalize interactions and gain insights into the system's behavior. This dissertation explores distributed analyses with biomedical applications and interconnected biological systems—logical candidates for graph formalization.

The representation of a problem not only might lead to "obvious" ways of solving it, changing representations might entirely shape the *complexity* of a possible solution. Hence, while the construction, analysis, and transformation of graph structure are the overarching subjects of this work, in some cases the projection of a graph problem onto a different representation space leads to a more efficiently computable solution (as we will show in Chapter 5).

By its very nature, the field of *Medical Informatics* deals with highly sensitive data—data, whose accidental or malicious disclosure might have serious effects on the social standing, informational autonomy, or even physical security of the individual. This places a moral burden on everyone handling and processing these data, including the researchers developing and implementing analysis algorithms. In this dissertation this obligation is addressed by designing and implementing algorithms with privacy as a primary concern in their design. This does not fulfill the responsibility once and for all, but determines clear "coordinates" in the space between data utility and privacy protection and helps to assess possible risks and their impact. This work strives to limit the need for centralized storage as well as (most of) the transmission of said sensitive data by developing, integrating and analyzing novel cryptographic methods for *Secure Multi-Party Computation* (MPC). This approach allows the joint analysis while providing provable security guarantees under clearly defined assumptions.

The research subject of this dissertation mirrors the diverse roots and applications of graph theory itself, since it is located at the interdisciplinary boundaries of physics, mathematics, cryptography, computer science, medicine, and biology. In the following chapters I demonstrate, that graph theory is a suitable model to understand and analyze the specific needs in the area of medical informatics and provide privacy-preserving and computationally efficient solutions enabling future treatments while respecting personal privacy and informational autonomy.

1.1 Thesis Outline

This dissertation is structured in three major parts. The remainder of the "Preliminaries" Part I—Chapter 2—introduces the context and background knowledge from the various fields included in this thesis; specifically:

- Section 2.1 introduces the fundamentals of graph theory,
- Section 2.2 discusses the relevant legal and technical aspects when it comes to data protection in the (bio-)medical realm, and
- Section 2.3 gives a concise primer of concepts, techniques, and primitives used for multi-party computation.
- Section 2.4 introduces the general notation used throughout this dissertation.

Part II presents the main research questions and results of this dissertation.

- Chapter 3 describes the design, implementation, and experimental evaluation of protocols for privacy-preserving analysis of non-linear interactions in complex genetic regulation networks to find relevant gene–gene interactions driving the expression of certain diseases. For the efficient design of the protocols, novel cryptographic building blocks are introduced.
- Chapter 4 discusses a secure protocol for solving the *Kidney Exchange Problem* (KEP), a graph optimization problem for optimally distributing kidney donor organs to treat patients with chronic kidney failure. As the calculation of the globally optimal solution to this problem is not tractable, we design a locally optimal heuristic generating and operating on graph structures, which offers the flexibly to be modified by medical experts.
- Chapter 5 introduces a ubiquitous problem in medical analyses, the problem of performing patient de-duplication while dealing with noisy, uncertain, and possibly missing information (known as *probabilistic record linkage* in medical informatics parlance). The solution's algorithmic design requires the definition and formalization of novel (mathematical) similarity order and tie-solving strategies. As an extension, three still ongoing applications involving the embedding of graph structures into binary fields are described: (1) the privacy-preserving calculation of a chemically meaningful similarity of small molecules, applicable for *virtual screening* procedures for de novo drug development, (2) a structural patient similarity measure to find "unspecifically" similar patients—a problem occurring for example during molecular tumor boards, and lastly (3) the identification of defects in transcriptome pathways. These extensions are still under research and pose a promising outlook regarding the applicability of the introduced solution outside its intended core domain.

Finally, Part III concludes this dissertation, summarizing the achieved insights, and presenting some promising future directions for continuing research.

1.2 Publications

The following thesis draws on the ideas and writing of the following published or submitted research articles. They are the result of my own research, performed jointly with several coworkers and collaborators, during my doctoral studies:

Stammler, S., **Kussel**, **T.**, Schoppmann, P., Stampe, F., Tremper, G., Katzenbeisser, S., Hamacher, K., and Lablans, M. (2020)
Mainzelliste SecureEpiLinker (MainSEL): Privacy-Preserving Record Linkage Using Secure Multi-Party Computation. *Bioinformatics* 38.6, pp. 1657–1668. DOI: 10.1093/bioinformatics/btaa764.

Hamacher, K., Katzenbeisser, S., **Kussel**, **T.**, and Stammler, S. (2020)
Genomische Daten und der Datenschutz. *Datenschutz und Datensicherheit (DuD)* 44.2, pp. 87–93. DOI: 10.1007/s11623-020-1229-9.

Wirth, F., **Kussel**, **T.**, Hamacher, K., and Prasser, F. (2021)
A Simple but Powerful No-Code Approach to Practical Secure Multiparty Computing in Medical Research: Development Study. *BMC Bioinformatics*. In Review.

Birka, T., **Kussel**, **T.**, Möllering, H., and Schneider, T. (2021)
"An Efficient and Practical Privacy-Preserving Kidney Exchange Problem Protocol". *33. Kryptotag (crypto day matters)*. Gesellschaft für Informatik e.V. / FG KRYPTO. DOI: 10.18420/cdm-2021-33-31.

Hamacher, K., **Kussel**, **T.**, Landesberger, T. von, Baumgartl, T., Höhn, M., Scheithauer, S., Marschollek, M., and Wulff, A. (2022)
Fallzahlen, Re-Identifikation und der technische Datenschutz. *Datenschutz und Datensicherheit (DuD)* 46.3, pp. 143–148. DOI: 10.1007/s11623-022-1579-6.

Kussel, **T.**, Brenner, T., Tremper, G., Schepers, J., Lablans, M. *, and Hamacher, K. * (2022)
Record Linkage-based Patient Intersection Cardinality for Rare Disease Studies using Mainzelliste and Secure Multi-Party Computation. *BMC Journal of Translational Medicine*. In Review, Pre-Print: https://www.researchsquare.com/article/rs-1486673/v1. DOI: 10.21203/rs.3.rs-1486673/v1.

Birka, T., Hamacher, K., **Kussel**, **T.**, Möllering, H., and Schneider, T. (2022)
SPIKE: Secure and Private Investigation of the Kidney Exchange problem. *BMC Medical Informatics and Decision Making*. In Review, Pre-Print: https://arxiv.org/abs/2204.09937.

Wettstein, R. *, **Kussel**, **T.** *, Hund, H., Fegeler, C., Dugas, M., and Hamacher, K. (2022)
"Secure Multi-Party Computation Based Distributed Feasibility Queries - A HiGHmed Use Case". *64. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e. V. (GMDS)*. Accepted.

Hamacher, K., **Kussel**, **T.** *, Schneider, T., and Tkachenko, O. * (2022)
"PEA: Practical Private Epistasis Analysis using MPC". *European Symposium on Research in Computer Security (ESORICS)*. Accepted.

* indicates equal contribution.

Chapter 2
Preliminaries

By definition, interdisciplinary research draws from ideas of many different fields. In this chapter we will describe the context in which the research of this thesis can be placed, as well as a brief introduction to the field specific background knowledge required for this thesis.

Some chapters require additional background knowledge only relevant for that specific chapter. This information will be presented in the respective chapters, as needed.

2.1 Graph Theory

Graph theory entered the stage of mathematics with Euler's pioneering solution of the "Bridges of Königsberg" problem in 1735[1]. At that time it was a favorite pastime of the burghers of Königsberg (whose two islands in the Preugel river were connected with each other and the main land via seven bridges) to muse, whether it was possible to cross all bridges exactly once. Not only did Euler answer this question in his work "Solutio Problematis Geometriam Situs Pertinentis", he abstracted it to find an intuitively correct solution for *any* number of islands and *any* number of bridges. By counting the bridges in and out of an area he explained, that only two areas—one if the path must end at the starting point—are allowed to have an uneven number of connections, as all areas beside the start- and endpoint must be entered and left again. In today's graph terms, he viewed the city of Königsberg as a (planar) graph with the landmasses being *vertices*, the bridges *edges* and found a beautiful proof by counting the vertex *degrees*. All those terms will be formally introduced in the following sections.

Some citizens of Königsberg
Were walking on the strand
Beside the river Pregel
With its seven bridges spanned

"O Euler, come and walk with us,"
Those burghers did beseech.
"We'll roam the seven bridges o'er,
And pass but once by each."

"It can't be done," thus Euler cried.
"Here comes the Q.E.D.
Your islands are but vertices
And four have odd degree."

- William T. Tutte

[1] Alexanderson (2006)

Since then, graph theory found many applications in different fields of science, as graphs turned out to be a very useful epistemological model of interconnected entities, abstracting problems from social sciences, e.g., friendship networks and community development, medicine, e.g., epidemic spread models, biology, e.g, food and pollination networks, modelling of ion channel dynamics, to engineering science, e.g., traffic congestion analysis, and informatics, e.g., computer networks. All those fields can be analyzed using methodology deeply rooted in statistical physics and discrete mathematics.

Graph structures are the primary way of problem description throughout this dissertation. All research projects can be understood in terms of graph theory and graph dynamics. The following sections give an overview of some basic graph properties.

2.1.1 Formal Definition of Graphs

Formally defined, a Graph is a pair of sets $\mathcal{G} = (V, E)$, where V is the set of *vertices* (also called *nodes*), and E the set of *edges* $e = (u, v) : u, v \in V$ connecting the vertices of the graph. If the direction of the edges are considered—in a directed graph, or *digraph*—the pairs of vertices (u, v) describing the edges are considered ordered, that is $(u, v) \neq (v, u)$. The starting vertex of an edge in a digraph is called *source*, the endpoint *sink*, invoking the image of flows.

For some applications multiple "types" of vertices exists and edges are only allowed between different vertex types. An example are pollination networks describing the relationship between multiple pollinating insects and plants in an ecosystem. The example can be represented as a *bipartite* graph $\mathcal{G} = (U, V, E)$ with edge pairs consisting of exactly one vertex in U and one in V. In the same way *tripartite*, etc. graphs can be constructed.

In many applications it is useful to denote the "importance" of an edge by associating a scalar value—a *weight*—with each edge. Especially while analyzing the dynamics of graph systems weighted graphs contain important information, e.g., regarding flows or capacities on edges. Figure 2.1 shows an example for a weighted digraph, the thickness of the arrows indicate the edge's weight.

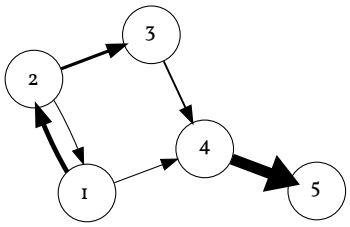


Figure 2.1: Example of a weighted digraph with five vertices.

Further generalizations of graphs include *multigraphs*, where multiple edges between the same pair of vertices can co-exist and "self-edges" are allowed, or *hypergraphs*, where an edge can connect more than two vertices. Hypergraphs are commonly used in cooperative game theory and social choice theory. Meta networks—networks between networks—can be described as *multilayer graphs*. Many more extensions are defined for special applications, e.g. *dipole graphs*, *colored graphs*, and *ancestral graphs*. While all those types of graphs extend the original definition, *dynamic graphs* add another dimension, by introducing time dependency in the graph structure, that is $\mathcal{G}(t) = (V(t), E(t))$, allowing vertices or edges to appear or disappear dynamically. Dynamic graphs play an important role for example in the analysis of infection models[2].

[2] Peixoto and Gauvin (2018)

A sequence of vertices consecutively connected by edges form a *walk*, if the walk is self-avoiding—that means it does not intersect itself—it is called a *path*. The length of a walk is often defined as its *geodesic distance*—the number of edges in the walk. A closed path starting and ending at the same vertex forms a *loop* or *cycle*. A *subgraph* H of a graph G is fully contained in G: $V_H \subseteq V_G, E_H \subseteq E_G$ and the sinks of all edges in E_H are in V_H.

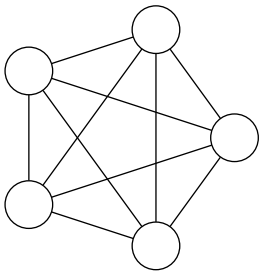


Figure 2.2: The fully connected K_5 graph

One prerequisite for many theorems in graph theory is that the graph is *planar*, that means that it can be embedded in a two-dimensional plane without crossing edges. KURATOWSKI's theorem shows, that a graph is only planar, if and only if (iff) it does not contain the fully connected K_5 graph (Figure 2.2) or the fully connected bipartite graph $K_{3,3}$ (Figure 2.3) as a *minor*—a minor being a graph obtained by removing vertices or edges, or by contracting edges and merging their endpoints.

2.1.2 Representation of Graphs

The definition of graphs in terms of sets allows for easy formalization and extension, however, the mathematical manipulation of specific graphs is unwieldy.

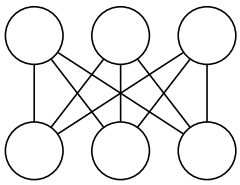


Figure 2.3: The fully connected bipartite $K_{3,3}$ graph

The most common representation of graphs as a mathematical structure is the *adjacency matrix*. It's a $n \times n$ matrix, where $n = |V|$ is the number of vertices in the graph. In its most basic form the adjacency matrix entries are:

$$a_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

In multigraphs other integer values can occur, denoting the number of edges between two vertices. Weighted graphs encode the edge weights as real values in the adjacency matrix ($A_{ij} = 0$ still means, that *no* edge exists between the two vertices). Adjacency matrices of undirected graphs are by construction symmetric, that is $A = A^T$. This is not necessarily the case for digraphs[3]. As the order of vertices in a graph is arbitrary[4], different permutations of the vertex order lead to different adjacency matrices describing the same graph.

The adjacency matrix is by no means the only possible representation of a graph utilizing structures from linear algebra, for example the *incidence matrix* is a $n \times m$ matrix, with $n = |V|, m = |E|$. The matrix entry B_{ij} indicates, whether vertex i is part of edge j. In directed graphs, the starting vertex is marked by a negative entry, so the semantics of outgoing and ingoing flows seems obvious[5]. Incidence matrices are useful representations for performing projections of graphs.

From a computational perspective, the graph property of *sparsity / density* is of importance. By normalizing the number of edges $m = |E|$ in a graph with the number of edges in a complete graph—a graph where every possible edge is present—we define the density of a graph as $d = \frac{2m}{n(n-1)}$, with n being the number of vertices $n = |V|$. A graph with $d < 1/2$ is called *sparse*, otherwise *dense*. The adjacency matrix for sparse graphs saves many zero entries, decreasing memory efficiency as well as algorithmic performance. For these cases the representation as an *adjacency list* might be beneficial, saving the neighboring vertices for every vertex as (linked-)list of lists (see, Figure 2.4).

2.1.3 Properties of Graphs

Graphs provide a framework to analyze various properties. In this section, some basic properties of graphs, vertices, and edges are shown.

One property of vertices already introduced in the "Bridges of Königsberg" problem is the vertex *degree*, the number of edges attached to this vertex. In an unweighted, undirected graph the degree k_i of vertex i is easily calculable using the adjacency matrix:

$$k_i = \sum_{j=1}^{n} A_{ij}.$$

[3] The following adjacency matrix describes the example graph in Figure 2.1:

$$A = \begin{pmatrix} 0 & 3 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

[4] Although, an appropriate enumeration of vertices may be important to efficiently compute the solution to some graph problems.

[5] The following incidence matrix describes the example graph in Figure 2.1:

$$B = \begin{pmatrix} -3 & 1 & 0 & 1 & 0 & -1 \\ 3 & -1 & -2 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1.2 & 0 & 0 \\ 0 & 0 & 0 & 1.2 & -7 & 1 \\ 0 & 0 & 0 & 0 & 7 & 0 \end{pmatrix}$$

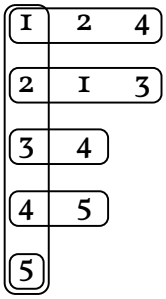


Figure 2.4: Adjacency list for the example graph in Figure 2.1. The weights are omitted for brevity and can be included by saving vertex–weight pairs.

In directed graphs, the degree is divided into *out-degree* and *in-degree*:

$$k_i^{\text{in}} = \sum_{j=1}^{n} A_{ij},$$
$$k_j^{\text{out}} = \sum_{i=1}^{n} A_{ij}.$$

Using the adjacency matrix representation, the number of paths of a certain length L between two vertices i and j is easy to calculate. By raising the (unweighted) adjacency matrix to the L-th power, the matrix elements give the number of paths: $\#p(i,j,L) = \left[A^L\right]_{ij}$. Of special importance are the diagonal elements giving the number of paths starting and ending at the same vertex—cycles. Note, however, that cycles are invariant under cyclic shift—a cycle $1 \to 2 \to 3 \to 1$ is identical to the cycle $2 \to 3 \to 1 \to 2$—leading to duplicates in the count.

The *diameter* of a graph is the length of the "longest shortest path."[6]—in formal terms $d = \max_{i,j \in V} \arg\min_r [A^r]_{i,j} > 0$, where $\arg\min_r [A^r]_{i,j} > 0$ gives the length of the shortest path between vertices i and j.

[6] Newman (2018)

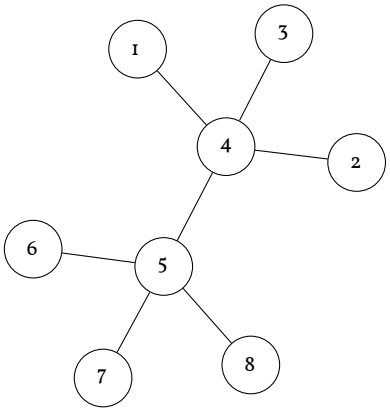


Figure 2.5: A "barbell" graph with one edge connecting two components.

By looking at the graph in Figure 2.5 and envisioning a some kind of "flow" between the nodes, it becomes obvious, that not all vertices are equally "important" in a topological sense. All flows between the two components must flow over the bridging edge between vertices 4 and 5. Removing one of those two vertices would stop all cross-component communication, while the removal of one "leaf" node would only decrease the amount of flow. The *centrality* property tries to measure the "importance" of a vertex or edge with regard to a specific property. A simple centrality measure is *degree centrality*, measuring the importance of a vertex by its number of connections. Related but more intricate is the *eigenvector centrality*, weighting a node's importance not only based on the number of neighbors, but *their* importance as well. The eigenvector centrality can be found by solving the eigenwert problem $A\vec{x} = \kappa\vec{x}$, where $\vec{x}$ is the eigenvector giving the centrality scores and κ is the largest eigenvector[7]. A modified form of eigenvector centrality is the *PageRank* centrality, famous for being the original ranking algorithm in the internet search engine "Google"[8]. Going back to the "flow" idea in Figure 2.5, the importance of nodes 4 and 5 is not well captured using the introduced centrality measures. *Betweenness centrality* however, directly captures the idea: it measures what fraction of the shortest paths between every vertex in the graph includes the respective vertex: $x_i := \sum_{k,\ell \in V} \frac{n^i_{k\ell}}{g_{k\ell}}$, where $n^i_{k\ell}$ is the number of the shortest paths between k and ℓ using vertex i and $g_{k\ell}$ the total number of the shortest paths between k and ℓ. This shows, that the appropriate metrics are highly dependent on the underlying research question and problem model.

[7] This is only true for connected graphs with positive edge weights. In this case the PERRON-FROBENIUS theorem assures, that all components of the eigenvector to (only) the largest eigenvalue are positive.

[8] Bryan and Leise (2006)

One important structural property of graph systems is the number of existing *components*. A component in an undirected graph is a subset of vertices, such that there exists a path between every member of this subset. Unconnected vertices form a component of size one by themself. If only one component exists, the graph is a *connected* graph. Graphs with more than one component can be represented—by permuting the vertex order—in block diagonal form[9].

[9] Adjacency matrix of a disconnected, undirected graph with three components:

$$A = \begin{pmatrix} \boxed{B_1} & 0 & 0 \\ 0 & \boxed{B_2} & 0 \\ 0 & 0 & \boxed{B_3} \end{pmatrix}$$

In directed graphs, a distinction between *weakly* and *strongly* connected components is made, where weak connectivity is achieved, if a path between all vertices exists when ignoring the edge directions. If all vertices are reachable from any other vertex even when taking edge direction into account, the graph is strongly connected. *Cliques* on the other hand, are small groups of vertices that are *fully* connected to each other.

2.1.4 Common Graph Topologies

Some graph topologies are prevalent enough to warrant an own name. This includes fairly simple topologies, like *lattices* (Figure 2.6) or *rings* (Figure 2.7). Lattices play an important role in statistical physics, e.g., for Ising spin glasses[10], as well as in cryptography, where the computational hardness comes from lattice-based graph problems[11]. Other important "simple" graph topologies are *trees* (Figure 2.8). Trees are acyclic, undirected, connected networks. The vertices with degree $k = 1$ are called *leafs*. Often trees are visualized with a dedicated *root* vertex as the base, however, in principle every vertex, even leaf vertices, can function as a root. An important subtype often encountered in computer science are *binary trees*, where every vertex "splits into two", i.e., the root vertex has degree $k_0 = 2$, every interior vertex has degree $k_i = 3$, and the leaf vertices have degree $k_l = 1$. The example graph in Figure 2.8 shows a binary tree.

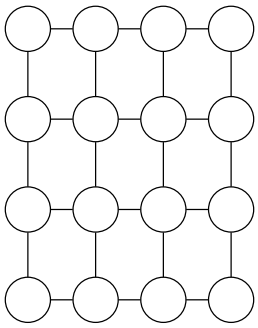


Figure 2.6: Example of a 4×4 lattice graph.

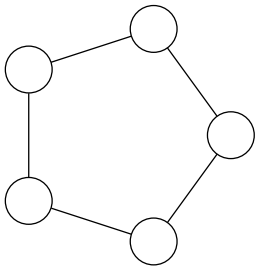


Figure 2.7: Example of a ring graph.

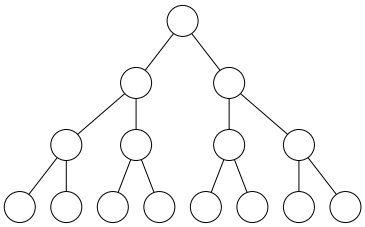


Figure 2.8: Example of a binary tree graph.

Other graph topologies are defined by their generation algorithm. This is especially true for the various *random graphs*, whose topology is randomly generated according to a generative algorithm. Examples are *Erdős-Rényi* graphs[12]—generated by creating N vertices with M randomly placed edges—*Watts-Strogatz* graphs[13]—generated by starting with a ring of N vertices that are regularly connected to K other vertices in the ring and then rewiring existing edges based on a random process—or *Barabási-Albert* graphs[14], where the generation starts with a small connected network and adds vertices one by one, each newly added vertex randomly connected to the already present vertices with a probability that is proportional to the (current) vertex degree. These random graphs are valuable to model real systems or study the properties of graphs themselves, as they all exhibit specific, but different properties. For example, Erdős–Rényi graphs are simple models for systems exhibiting percolation phase transitions allowing the analytic description of many properties, Watts-Strogatz graphs all exhibit *small-world* characteristics—that is the average distance between two randomly chosen vertices scales with the logarithm of the number of vertices—important for the study of social graphs or computer chip architectures, and Barabási-Albert graphs are *scale free*, that is exhibiting a power-law degree distribution—as for example found in internet topologies. Furthermore, random graphs connect the study of graph systems, via an adjacency matrix representation, to the rich body of work in the field of random matrix theory[15].

[10] Baxter 2016, pp. 88.
[11] Micciancio and Goldwasser (2002)
[12] Erdös and Rényi (1959)
[13] Watts and Strogatz (1998)
[14] Albert and Barabási (2002)
[15] Akemann, Baik, and Di Francesco (2011)

2.2 Methods for Data Protection

While being subject to societal changes[16]—as most social values—the right for privacy and informational autonomy are strongly embedded in European laws. With the ability to quickly "percolate" information on a global scale, paired with a new kind of "permanence" of data, the introduction of (quantitative) information technologies in every area of our lives lead to privacy concerns on a new scale. This is especially true concerning personal medical data[17].

[16] Bezanson (1992); Jones (2003)

[17] Goldsmith (2000)

Going back to telemedicine ideas in the early 1900s[18], telematics, as well as patient and inmate administration during world war two[19] introduced computers and automated data processing into the medical realm and fundamentally changed the field. Ultimately it led to the emergence of the field of medical informatics in the 60th and 70th, and the ubiquity of medical data in present days[20]. This kind of data has to be considered especially sensitive, as the disclosure could seriously affect the involved individuals regarding social standing and even physical well-being. The primary European governing framework for data protection, the *General Data Protection Regulation* (GDPR) acknowledges the sensitive nature of health data and contains special restrictions to the transfer and processing of genetic, biometric, and health related data[21].

[18] Bashshur and Shannon (2009)

[19] Black (2001); Aly and K. H. Roth (2018)

[20] Ambinder (2005)

[21] §4 1. GDPR

Research endeavors, on the other hand, rely on the availability and processing of data. Many medical and genomic studies require a vast information base to reach statistically significant conclusions. In this area of tension, a plethora of laws and regulations—ranging from the GDPR to national, e.g., the German Federal Data Protection Act (*Bundesdatenschutzgesetz*, BDSG) or even regional laws, e.g., the German State Hospital Act (*Landeskrankenhausgesetz*, LKHG)—regulate privacy risk assessments, methods, and causes that allow the access of protected data.

This section of the dissertation draws upon work published in Hamacher, K., Katzenbeisser, S., Kussel, T., Stammler, S. (2020) "Genomische Daten und der Datenschutz". *Datenschutz und Datensicherheit (DuD)* and Hamacher, K., Kussel, T., von Landesberger, T., Baumgartl, T., Höhn, M., Scheithauer, S., Marschollek, M., Wulff, A. (2022) "Fallzahlen, Re-Identifikation und der technische Datenschutz". *Datenschutz und Datensicherheit (DuD)*. The author was deeply involved in both publications contributing sections concerning technical data protection methods and GDPR related implications to the first and being the primary author of the latter.

Five Safes Framework

While this dissertation concentrates on algorithms for technical data protection, the secure analysis is only one step of providing data privacy and minimizing individuals' risks. A widespread and more "holistic" framework providing a frame of reference for all data access and data privacy considerations, is the *Five Safes* framework[22]. The framework distinguishes five, mostly orthogonal dimensions of data access (taken from T. Desai, F. Ritchie, and Welpton (2006)):

[22] T. Desai, F. Ritchie, and Welpton (2006)

- **Safe Projects:** Is the use of the data appropriate?
- **Safe People:** Can the researchers be trusted to use the data in an appropriate manner?
- **Safe Data:** Is there a disclosure risk in the data itself?
- **Safe Settings:** Does the access facility limit unauthorized use?
- **Safe Outputs:** Are the statistical results non-disclosive?

It stresses, that security is not a binary state, but a measure with many intermediate values on multiple axes.

Most of those dimensions have been accounted for in medicinal research for a long time, for example, data usage for research purposes must be cleared by the *Institutional Review Board* (IRB) and an *Use and Access Committee* (UAC) of the data owning party, casting an ethics vote with regard to the appropriate data usage (Save Projects). Additionally, most data access request procedures mandate user training concerning data protection (Save People). As a last example, data protection regulation such as *Health Insurance Portability and Accountability Act* (HIPAA) and the GDPR require some kind of re-identification protection for the use of cohort data, such as k-anonymity, introduced in the next (sub)section (Save Outputs).

The algorithms and methods introduced in this dissertation are concerned with providing a very high security level in the areas of "Safe Data" and "Safe Settings".

2.2.1 Pseudonymization and Anonymization

The storage of *Identifying Personal Data* (IDAT) together with *Medical Data* (MDAT) is generally only permitted in an explicit medical care context. For the use of clinical data for research purposes, public health, and epidemiology, the German Federal Data Protection Act (*Bundesdatenschutzgesetz*, BDSG) requires the pseudonymization or anonymization[23] of the data. Both concepts pursue the same goal: the utilization of the sensitive data while taking the data protection of the individual entries into account, but under different "boundary conditions".

[23] §§48, 50 BDSG

When data records are *pseudonymized*, the fields that are considered to be personally identifying are removed from the data record and a pseudonym is included as an identifier. That way no conclusions can be drawn about the identity of the individual "without the use of additional information"[24]. The pseudonymized dataset can be analyzed and, if necessary, individual persons can be re-identified by back-translating the pseudonyms—provided the access to the IDAT–pseudonym mappings. This back-translation is subject to severe legal restrictions. These pseudonyms are referred to as *first-order pseudonyms*.

[24] §46 5. BDSG, translation by the author

To further protect the identity of the individual, e.g., in specific cohort studies, the first-order pseudonym is subsequently replaced by a different pseudonym, unique to this study—the use-case specific *second-order pseudonym*. This step increases the difficulty to correlate the datasets between independent studies, thus

allowing the re-identification of a common subset of patients—a vector exploited in so-called *linkage attacks*. This additional pseudonymization is usually performed by fiduciary-operated and -certified pseudonymization services.

Data records are considered *anonymized* if they are modified in such a way that the "individual information regarding personal or factual circumstances"[25] cannot be assigned to an individual—or only with disproportionate effort. This definition includes so-called *quasi-identifiers*, i.e., data that is not considered identifying per se, but is in combination with other quasi-identifiers unique enough to allow the re-identification of a person. Hence, anonymization guarantees a stronger form of privacy than pseudonymization, as the possibility to link pseudonyms back to IDAT is not allowed under the definition of anonymization[26].

[25] §3 (1) BDSG, 2009 Version, translation by the author

[26] §46 5. BDSG, §4 5. GDPR

One central area of tension in the legal domain is whether a relative or an absolute interpretation of personal reference (German term of art: "relativer oder absoluter Personenbezug") is required for anonymization[27]. These positions differ in the assessment of what external knowledge an adversary might use with the absolute position being an extreme position of assuming that an adversary might have *all* external knowledge, including illegally acquired information and full access to all computer systems. This position would not count securely encrypted data as anonymized, as long as a decryption key is known to *anyone*, including authorized parties. The relative position excludes illegal means—such as hacking—from the "reasonable" ways for an adversary to acquire external information. Both extreme positions—and intermediate positions—are valid interpretations of legal texts and currently no resolving precedence exists[28].

[27] Klar and Kühling (2018)

[28] Klar and Kühling (2018)

In an American study[29] it was shown that almost 90 % of the records of the 1990 U.S. Census can be linked to an individual person using the characteristics "zip code", "gender," and "date of birth". Using the characteristics "county", "sex," and "date of birth", still nearly one-fifth of U.S. citizens can be uniquely identified.

[29] Sweeney (2000)

To characterize the extent of anonymization, different measures are used. The most relevant measures for this *Statistical Disclosure Control* (SDC) in medical practice are k-*anonymity*, ℓ-*diversity*, and t-*closeness*.

k-anonymity

A DATASET SATISFIES k-*anonymity*[30] if all combinations of attributes are satisfied by at least k entries in the dataset. Depending on the privacy requirements of the application, a specific value for k is chosen—in medical practice values of $k = 5$, in rarer cases $k = 3$ or $k = 11$, are deemed sufficient[31]. To achieve k-anonymity, datasets are "binned" into *equivalence classes*, such that each class consists of at least k entries. For example, an equivalence class "A" could include all male patients between 30 and 40 of age in the 64XXX zip code area.

[30] Sweeney (2002)

[31] European Medicines Agency (2017); Oswald (2013)

The measure of k-anonymity is easy to attack if the sensitive characteristic is distributed homogeneously within an equivalence class (e.g., if almost all the patients in equivalence class "A" suffer from COVID-19). That way the medical condition of a specific person could be inferred with high likelihood—even if only a few, coarsely binned quasi-identifiers are known to the attacker. The largest

attack vector, however, consists of associating the k-anonymous dataset to additional, external information[32].

[32] See the discussion of absolute and relative interpretation of personal reference above.

To counteract the described risks, the *ℓ-diversity* measure[33] requires that the sensitive feature within each equivalence class has some variability. The models for measuring this variability range from very simple (e.g., "there must be at least ℓ different diagnoses be included") to mathematically more complex, e.g., entropy-based methods. The ℓ-diversity can be further generalized to the more advanced measure of *t-closeness*[34] in which the frequency distribution of the sensitive features in each equivalence class must match the frequency distribution of the Features in the entire dataset, up to a threshold t.

ℓ-diversity

[33] Machanavajjhala et al. (2007); Stammler, Katzenbeisser, and Hamacher (2016)

t-closeness

[34] N. Li, T. Li, and Venkatasubramanian (2007)

2.2.2 Utility and Re-identification

The anonymity measures presented above are conceptually simple to understand and easy to use, unfortunately, all suffer—albeit to varying degrees—from vulnerabilities that allow the re-identification of individuals. Furthermore, the choice of appropriate parameters is difficult, as the data loses *utility* for parameter choices with higher anonymization. For example, higher values of the k-parameter for k-anonymity, group the data into coarser and coarser classes to ensure the higher anonymity, thus removing information. Finally, all of the above measures are difficult to apply if not only one, but multiple sensitive features, like a primary *and* a secondary diagnosis, have to be protected.

In the following, we present two different attacks on anonymized datasets that are by no means theoretical in nature, but their attack vectors are actively exploited and can be used against anonymized datasets of real production systems: *linkage attacks* and *tracker attacks*.

Linkage attacks

A *linkage attack* is an attempt at re-identification in which the pseudonymized or anonymized data are correlated with other information known to the attacker.

The canonical example of a linkage attack is the successful re-identification of U.S. Gov. William Weld in 1997. According to the narrative Weld's medical records in a pseudonymized insurance dataset were correlated with the public voter registration records of Cambridge, Massachusetts, thereby re-identifying Weld. Systematic research[35] indicates, however, that this sequence of events is probably more of a myth. Much more likely, the linking of the insurance data with the public hospitalization of the governor led to the re-identification—a different story, yet a successful linkage attack.

[35] Barth-Jones (2012)

Another, more recent example is the re-identification of users of the Netflix Prize dataset. In October 2006, the streaming company issued a prize to improve the system for suggesting new movies based on previous viewing habits. Included in the call for entries was an anonymized dataset containing the movie ratings of half a million users over a period of five years. Using the film ratings from the "Internet Movie Database" (IMDB)[36] as an additional data source, two researchers

[36] https://imdb.com

from the University of Texas were able to identify users from the anonymized database[37] —including strong indications of political and religious beliefs, as well as sexual preferences.

[37] Narayanan and Shmatikov (2007)

These two examples illustrate a fundamental aspect that is exploited in linkage attacks: a data owner can only control the pseudonymization and anonymization of its *own* data. Even if a data owner anonymizes the data considering additional datasets known at the time of publication, it cannot be ruled out that this anonymized dataset will be completely de-anonymized by new available datasets at some point in the future.

Tracker attacks

Datasets that are not published in full, but rather allow interactive queries, for example the query of a hospital information system for the blood pressure of all patients between 30 and 40 with the diagnosis "liver abscess," are threatened by so-called *tracker attacks*[38]. In this attack, the query results are stored by a *tracker*, thus systematically expanding knowledge of the database over the course of multiple queries. The individual queries can be designed in such a way that they fulfill the usual anonymization measures but, for example, their intersection or difference sets reveal sensitive information. For example, the query of *all* diagnoses made would presumably be k-anonymous with a high value for k. The problem arises with a second query like "The diagnoses of all patients not named John Doe", which—k anonymous in itself as well—allows leakage by comparing both results.

[38] D. E. Denning, P. J. Denning, and Schwartz (1979)

Even if databases are carefully pseudonymized, sensitive information can be extracted or patients can be re-identified using trackers. Two pragmatic mitigation measures are:

1. Query rate limiting: A user of the database can only make a limited number of queries in a period of time. This prevents automated and fast executions of complex attacks, which require a high number of queries.
2. Preservation of an audit history: The queries of all users are stored, so that in case of misuse of the database the misbehaving user can be traced.

These countermeasures are either reactive or prevent only certain attacks. More advanced procedures that analyze the request history of users and thus prevent trackers are complicated, aimed against known attacks, and for large databases computationally and storage-wise expensive.

2.2.3 Differential Privacy

One *Statistical Disclosure Control* (SDC) technique defeating not only the described attacks, but protecting the data against *any* correlations or multiple queries, is *Differential Privacy* (DP)[39]. Albeit related to the previously discussed anonymity measures, DP allows for mathematical exactness in the risk assessment. It is based on the statistical perturbation of the data with the goal that the same query result must be obtained *regardless of the presence or absence of a specific dataset*. This re-

[39] Dwork, McSherry, et al. (2006)

quirement was formalized into a rigorous mathematical framework by Dwork, McSherry, et al.[40] in 2006 in which the controlled addition of stochastic noise is used to not only hide the *exact* result but to prevent the inference of individual datasets, even with additional sources.

[40] Dwork, McSherry, et al. (2006)

This method is comparable to a technique used for sociological surveys for the investigation of, for example, illegal or socially unacceptable behavior: *plausible deniability*. The following protocol for yes/no questions shows an intuitive example:

The respondent tosses a coin. If the result is "heads", he answers the question truthfully. If the result is "tails," a second coin is tossed. If the second toss result in "heads," the answer is "yes"; if the result is "tails," the respondent answers "no".

A respondent can justify his answer with the random process at any time. As the inserted *noise* follows a known statistical distribution, the "true" distribution can be inferred from a sufficiently large number of samples.

The perturbation of the results follows a strict *privacy budget* defined by the parameters ϵ and δ, both quantifying the level of anonymization and the loss of utility. Two formal definitions form the basis of DP. Firstly[41], a mechanism is ϵ-indistinguishable if for all pairs $\mathbf{x}, \mathbf{x}'$ which differ in only one entry, for all adversaries $\mathcal{A}$, and for all transcripts t[42]:

[41] Dwork, McSherry, et al. 2006, p. 270.

[42] A transcript is the result of an interaction between user and a privacy mechanism, such as the result of a single query function. The definitions presented generalize to more abstract notions of transcripts.

$$\left| \ln \left(\frac{\Pr[\mathcal{T}_{\mathcal{A}}(\mathbf{x}) = t]}{\Pr[\mathcal{T}_{\mathcal{A}}(\mathbf{x}') = t]} \right) \right| \leq \epsilon.$$

As ϵ is small, a Taylor approximation results in the roughly equivalent formulation $\frac{\Pr[\mathcal{T}_{\mathcal{A}}(\mathbf{x})=t]}{\Pr[\mathcal{T}_{\mathcal{A}}(\mathbf{x}')=t]} \in 1 \pm \epsilon$, meaning that the probability of an attacker receiving the same transcript from querying either the database or one differing in exactly one entry is arbitrarily close to one.

Secondly[43], a randomized algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private, if for all $\mathcal{S} \subseteq Range(\mathcal{M})$ and for all databases $\mathbf{x}, \mathbf{x}'$ such that $||\mathbf{x} - \mathbf{x}'||_1 \leq 1$:

[43] Dwork and A. Roth 2013, p. 17.

$$\Pr[\mathcal{M}(\mathbf{x}) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(\mathbf{x}') \in \mathcal{S}] + \delta,$$

where $||x-y||_1$ is defined as the ℓ_1 distance between databases, giving the number of differing entries.

This second definition allows the precise control over the amount of noise to add (most often sampled from a *Laplace distribution*—A symmetric version of the exponential distribution with *Probability Density Function* (PDF) $f(x) = \frac{1}{2\sigma} e^{-\frac{|x-\mu|}{\sigma}}$— but not restricted to this distribution) and leads to elegant composeablility theorems.

The introduction of noise, while leading to both safe data and safe outputs, is not appropriate for all analyses and data types. Especially in the analysis of genomic data, statistical perturbation is generally problematic, even though efforts

in some applications, such as *Genome-Wide Association Study* (GWAS) successfully used DP (e.g., Simmons and Berger (2016)[44] and M. Wang et al. (2017)[45]). Another example are many analyses in the field of rare diseases. Here, the data lose much of their significance due to the small number of cases.

[44] Simmons and Berger (2016)
[45] M. Wang et al. (2017)

For many other applications, however, these specific limitations do not apply, such that the methods and error estimates of DP could be used without further issues.

2.3 Secure Multi-Party Computation and Homomorphic Encryption

Most fields of cryptography consider incomplete trust in message channels. Two parties, traditionally called Alice and Bob, want to transmit information and want to prevent an adversary, Eve, to gain access to that information with respect to some measure and security model (cf. Section 2.3.1). The goal of this kind of cryptography is to achieve all or some of the following communication properties[46]:

[46] Stinson (2005)

- **Confidentiality:** An eavesdropper may not gain any information when observing the communication over an insecure medium.
- **Authenticity:** The communicating parties can validate, that they are actually communicating with the intended party.
- **Integrity:** Any modification and tempering with a message can be detected.

These means can be achieved using (symmetric or asymmetric) encryption schemes, signature schemes, or message authentication codes. One principle that underlies modern cryptography is *Kerckhoffs' Principle*:

> THE SYSTEM MUST NOT REQUIRE SECRECY AND CAN BE STOLEN BY THE ENEMY WITHOUT CAUSING TROUBLE
> — AUGUSTE KERCKHOFFS[47] (translation by Aumasson[48])

[47] Kerckhoffs (1883)

[48] Aumasson 2017, p. 40.

This means, that the construction must be secure even if the exact algorithm, the parameter values, and so on *except* the secret key are publicly known. Furthermore, the field of *steganography* tries to hide the presence of a message in a medium, e.g., by hiding a (text) message in the least significant bits of a digital image, which leads to an image indistinguishable to the original for a human observer.

MPC and HE

Secure Multi-Party Computation (MPC) and *Homomorphic Encryption* (HE) explore a different trust model: a number of parties want to jointly engage in a calculation *but do not trust each other*. This setting is orthogonal to the security of the communication channel, however, as we will see the security of the channel might lead to different security guarantees achievable. Note, that while this trust relationship is also true for the usage of Σ-*protocols* and *Zero Knowledge Proofs* and

those techniques are sometimes used in MPC to "force" a participant to behave honestly[49], this interesting and rich field of research is outside the scope of this work.

[49] Hazay and Lindell 2010, pp. 147.

Secure Multi-Party protocols enable the joint computation of an *ideal functionality*. That means, that any calculation, that is possible by sending every party's secret inputs to a *Trusted Third Party* (TTP), which performs the calculation and only discloses the result, is possible without a TTP, emulating it in a cryptographic protocol. These protocols strive to achieve the following properties:

- **Privacy:** No party may learn anything more than the computation output, in particular regarding the parties' private inputs.
- **Correctness:** Each party is guaranteed, that the received calculation result is correct. Note, that this is, of course, related to the correctness of inputs. A corrupted input may lead to a wrong result.[50]
- **Independence of Inputs:** Corrupted, i.e., adversarial parties may not choose an input, that depends on the honest parties' inputs. Due to differing *malleability* guarantees, otherwise it might be possible for a corrupted party, e.g., in a sealed auction, to create an input bid $(x + 1)$€, based on an (encrypted) bid x, without knowing what value x takes.
- **Guaranteed Output Delivery:** Adversaries may not be able to prevent an honest party to receive its output.
- **Fairness:** The corrupted parties might only receive their outputs, iff the honest parties do so as well. Fairness is implied with guaranteed output delivery, however, as the corrupted party might opt to abort the calculation (and the output disclosure) the reverse is not necessarily true.

[50] *"On two occasions I have been asked, 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answer come out?' I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question."*
— Charles Babbage

Note, that not all protocols fulfil all guarantees, neither are all guarantees important for every application. More on that is described in Section 2.3.1.

Both MPC and HE work by representing the desired function as a *Boolean* or *Arithmetic Circuit*, that is a *Directed Acyclic Graph* (DAG). The vertices encode two kinds of operations, in the case of Boolean Circuits the logical `AND` ($\wedge$) and `XOR` ($\oplus$) operations, for Arithmetic Circuits the `MUL` ($\cdot$) and `ADD` ($+$) operations. Further borrowing the nomenclature from electrical engineering, the vertices are called *gates* and the edges *wires*. Using $(\wedge, \oplus)$ and $(\cdot, +)$, as basis operations, respectively, arbitrary (bounded) functions can be represented. The functions must be bounded, as the circuits must be constructed independent of the inputs, to prevent information leakages through side channels, like execution timings. That means, that all loops must be completely unrolled and in cases of branching functionalities all branches are evaluated. Unfortunately, this prohibits large classes of common optimization techniques and reinforces the similarity to electrical engineering, as hardware circuits are subject to these same restrictions[51].

[51] Cf. Songhori et al. (2015)

Generally speaking, most MPC protocols provide to varying degrees a separation between two distinct phases, a *setup phase* in which all initial, input independent

computations are performed. The following *online phase* marks the evaluation of the functionality on the distributed inputs. In some applications only the run time of the online phase is of importance, for example if a rare but periodic calculation must be performed. In other applications only the combined runtime is of significance, e.g., for one-off analyses. To generalize further, the size of communication data constitute the major bottleneck when it comes to runtime performance of circuit-based MPC. Most modern software implementations are optimized to use mostly symmetric cryptographic operations and the hardware's cryptographic instruction sets or coprocessors. This is not necessarily true for *Fully Homomorphic Encryption* (FHE) schemes, which are often bound by memory bandwidth[52].

[52] Castro et al. (2021)

Homomorphic Encryption provides *computations under encryption*. That means, that some operation can be performed on the cipher text without decrypting it, that leads to the *encrypted* result. In more formal notation: $\mathsf{Enc}(m_1) \odot \mathsf{Enc}(m_2) = \mathsf{Enc}(m_1 \oplus m_2)$, where the operations denoted with $\odot$ and $\oplus$ depend on the specific crypto system used. More details regarding homomorphic encryption are presented in Section 2.3.9.

2.3.1 Security Models

One of the most important aspects of modern cryptography is the formal rigidity, allowing security proofs. This makes a rigorous definition of the *threat model*, that is what is to be protected, what capabilities does an adversary have, etc., as well as the clear statements of the underlying assumptions an important necessity. Both aspects allow or require statements in a number of dimensions.

One coarse but important dimension describes the computational capabilities of the adversary. A protocol or algorithm can be *information theoretically secure*, that means that even an adversary with infinite time and unbounded computational performance can not break the encryption. This is the strongest possible guarantee in terms of adversarial abilities and comes with a wide range of conditions, e.g., in case of an encryption scheme the secret key *must* be at least as long as the message.

One example is the *One-Time Pad* (OTP) encryption: For every character of the message a character is randomly chosen and both characters are "added", where addition means adding the ordinal numbers of the respective characters in the alphabet, modulo the total numbers of possible characters. Even if the attacker tries to brute force the decryption by enumerating all possible key-character sequences, depending on the key all messages with the given length can be decrypted[53]. Due to the equal length of message and key the problem of securely transmitting a message is only evaded: now the question becomes "How to securely transmit the key?" To make matters worse, the usage of a *Pseudorandom Number Generator* (PRNG) is insufficient for achieving information theoretic security with the OTP, the characters in the key must be *truly* random. As the name suggests, a key may only be used once.

[53] Consider the ciphertext `EUAYOUIYCW`. Using the key `PKAMHKRMOY` it decodes to `SECRETTEXT`, a seemingly valid message. Using the key `HOTGDGLWPI` it decodes to `LITERATURE`. Without additional information it is impossible to determine the correct message.

A weaker model is *computational security*, where both time and computational powers of an attacker are restricted. A scheme is deemed secure under this model, when a "realistic" attacker has only "negligible" chance to break the scheme in "feasible" time. As all these things are moving targets, a more rigorous security definition is required and concepts of complexity theory are useful. If a cryptographic system takes polynomial time to execute regularly but super-polynomial time to break, the user gains a systemic advantage over the adversary[54]. Even if the gap is initially small, increasing, e.g., the length of the key widens the gap until a successful attack is sufficiently unlikely. However, new algorithms or better computing hardware might require an increase in key length or even break the cryptosystem in its entirety.

[54] Vadhan (2011)

When considering cryptographic protocols, additional adversarial capabilities are to be considered. These capabilities limit the adversary in *how many* protocol participants they can corrupt and *when*, as well as how those corrupted parties participate in the protocol's execution.

In the dimension of allowed behavior, the space is spanned by *semi-honest adversaries* on one and *malicious adversaries* on the other end. In the setting with a semi-honest adversary—also called *passive security*—the corrupted parties correctly follow the protocol's specification. However, they try to gain information that should remain private. This setting mostly protects against accidental data leakage and curious, but unauthorized technical personnel with access to the computing system. Malicious adversaries on the other hand—in the *active security* setting—are allowed to arbitrarily deviate from the protocol. Informally speaking, in this setting adversaries can do whatever they wish. A setting more closely modelling real-world threat actors, is the model of *covert adversaries*. Those adversaries may arbitrarily deviate from the protocols, however, each deviation is detected—and hence punished—with a certain, non-negligible probability.

When it comes to the time domain, this work mostly deals with *static corruption*, where the adversary-controlled parties are fixed at the beginning of the protocol execution. Corrupted parties stay corrupted and honest parties remain honest. In other models, for example *adaptive corruption* or *proactive security*, the adversary may corrupt parties dynamically based on the state of the protocol, in the latter case a corruption might be "cleaned", allowing for "temporary" compromising.

The number of corrupted participants plays an important role in which security guarantees are achievable. For example, in the *dishonest-majority* case where the number of corrupted parties t is more than half of all n participants ($t \geq n/2$), the properties of "fairness" and "guaranteed output delivery" cannot be achieved. In the *honest-majority* case ($t < n/2$), all guarantees can be given, assuming all participants have additional access to a broadcast channel. Without that broadcast channel, the full set of guarantees is achievable for $t < n/3$, if the protocol runs on a synchronous point-to-point network with either *private* or *authenticated* channels, achieving information-theoretic security or computational security, respectively[55].

[55] Lindell (2020)

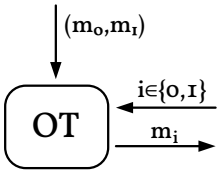


Figure 2.9: Schematic functionality of Oblivious Transfer: one party prepares two messages m_0, m_1. A second party provides a choice bit i, receiving the corresponding message. The preparing party must not learn the choice bit, the choosing party must not learn the non-selected message.

While considerably weaker than active security, the semi-honest security model is appropriate for many applications. The research projects in this work deal mostly with computations between *known* medical institution within the same legal domain with *contractual agreements*. That way protocol participants can reasonably be expected to behave honestly. Additionally, active security imposes a considerable runtime penalty, often more than one order of magnitude. This renders security infeasible for many complex applications. However, for international computations with participants in different jurisdictions the European Data Protection Board recommends active security[56].

[56] European Data Protection Board 2021, pp. 33.

2.3.2 Oblivious Transfer

Oblivious Transfer (OT) is a cryptographic primitive used in most MPC protocols. Originating in physics under the name *Conjugate Coding*[57] it described a communication protocol for quantum mechanical systems. Later, it was popularized in cryptography by Rabin [58,59].

[57] Wiesner (1983)

[58] Rabin (1981)

[59] Rabin first developed this idea to simulate noisy communications channels, closely following Wiesner's work.

Figure 2.9 shows the functionality of (basic) Oblivious Transfer schematically: One party prepares two messages, m_0 and m_1. By sending a choice bit $i \in \{0, 1\}$, the other party receives the selected message, however, the message-providing party is not allowed to learn the choice bit and the message-receiving party is not allowed to learn the non-selected message.

Wiesner's Conjugate Coding

Wiesner developed his ideas on conjugate coding around 1970[60], however, his work was published more than a decade later in 1983[61]. In the manuscript, he develops, in fact, a OT protocol based on the transmission of particles in *conjugated bases*—that is for two orthonormal bases $|a_i\rangle, |b_i\rangle, i = 1, 2, \ldots, N$ of a N dimensional Hilbert space $|\langle a_i|b_i\rangle|^2 = \frac{1}{N}$ holds for all i. This protocol is described for two or three conjugated bases, furthermore Wiesner proves, that the protocol is—in theory—extendable to N messages.

[60] Brassard (2005)

[61] Wiesner (1983)

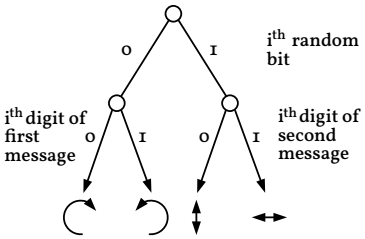


Figure 2.10: Schematic protocol for two-party conjugate coding using linearly polarized and circular polarized light as conjugate bases.

The protocol in the two message case works by first choosing two conjugated orthonormal bases—for example (1) linear polarization in horizontal and vertical direction and (2) right- and left-hand circular polarization. Next, a random bit sequence is sampled with the same length as the (binary encoded) messages. This random sequence chooses from which message to transmit a bit, e.g., if bit i in the random sequence is 0, transmit bit i of message m_0, and of message m_1 otherwise. If the bit of message m_0 is 0, transmit a single photon in vertical polarization. If it is 1 transmit the photon in horizontal polarization. For message m_1 choose right-hand and left-hand circular polarized photons, respectively. The protocol is schematically displayed in Figure 2.10.

The receiving party has to choose which basis to use for his measurements, losing the ability to measure photons in the conjugated polarization. If the receiver is set to elliptical polarization in an attempt to receive both messages, less information is received. Note, that even for "correct" polarization choices on average only half

of the bits of the message are learned. Wiesner adds this information loss to channel noise, photon shot noise, and photomultiplier noise and proposes the usage of error correcting codes. However, he also admits that theoretical, albeit unpractical attacks against his protocol exist[62].

Chou and Orlandi[63] provide an OT protocol, designed for simplicity (see Protocol 2.1). $\mathsf{H}()$ denotes a (keyed) hash function[64], g a generator of a prime-order group $\mathbb{G}$ in which the computational Diffie-Hellman problem is hard. Note the similarity to the Diffie-Hellman key exchange protocol. In their work, the authors describe several extensions and variations to the protocol, e.g., *1-out-of-n OT*[65], where arbitrarily many messages are prepared and are selectable, *random OT*, where one out of two random messages are selected (as used in the GMW protocol, cf. Section 2.3.4), or XOR and arithmetically correlated OT[66], where the messages show a chosen correlation—Figure 2.11 shows the example of XOR correlated OT, used for example in the *free XOR* optimization of Yao's Garbled Circuits (cf. Section 2.3.3).

Unfortunately, Oblivious Transfer is computationally expensive, as it can not be reduced to symmetric cryptography operations only[67]. For a complex MPC protocol millions of OT invocations are required, as they might—depending on the protocol—be necessary for every AND or every input bit. However, due to *OT precomputation*[68], *"seeded" OT*[69], and *OT Extension*[70], only a few expensive "base OTs" must be performed, to allow large numbers of fast OT computations using only symmetric crypto and one-time pad operations.

Beaver's OT precomputation uses OT on random inputs during a setup phase to enable OT operations with only XOR operations during an online phase. "Seeded" OTs and OT Extension reduce the required communication in two different dimensions (see Figure 2.12): "Seeded" OTs allow to only transmit a short seed via OT to "transfer" long strings and OT Extension allows acquiring many OTs from a few base OTs. One recent construction for efficient OT Extension is *Silent OT Extension*[71], trading computational cost for less communication.

2.3.3 Yao's Garbled Circuits

Andrew C. Yao's 1986 seminal work "How to Generate and Exchange Secrets"[72] founded the field of secure Multi-Party computation. It describes the *Millionaire's Problem*: A group of (fictive) millionaires want to know who of them is the wealthiest. However, being naturally distrusting they don't want to disclose their networth. Yao proceeds to describe the *Yao's Garbled Circuits* (GC) protocol, not only the first MPC protocol but (with many optimizations) still in use and important for practical applications of MPC. The protocol operates on a Boolean Circuit representation of the functionality, as described in the introductory text of this section.

The two-party protocol works by assigning different roles to the participants: one party assumes the role of the *garbler*, preparing the circuit, and the other party performs the duties of the *evaluator*, evaluating the circuit without having any insight regarding the semantics of the performed calculation.

[62] Later in the paper, he describes a scheme for unforgable currency suffering from the opposite problem—being theoretically sound but utterly impractical.

[63] Chou and Orlandi (2015)

[64] Modelled as a random oracle in their security proofs. The random oracle model is a weakening of the security model, compared to the cryptographic standard model. In the cryptographic standard model the adversary is only bound by its computational power and available time—cf. Naccache (2011).

[65] Kolesnikov and Kumaresan (2013)

[66] Asharov, Lindell, et al. (2017)

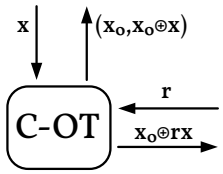


Figure 2.11: Schematic functionality of XOR-correlated Oblivious Transfer: The message preparing party prepares (random) messages, that include a defined correlation.

[67] Impagliazzo and Rudich (1989)

[68] Beaver (1995)

[69] Beaver (1996)

[70] Ishai et al. (2003)

[71] Boyle et al. (2019)

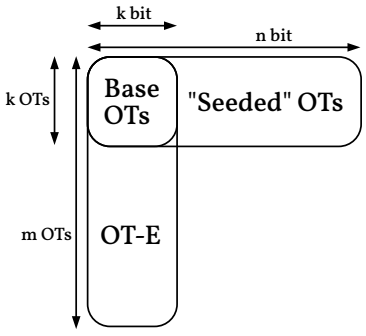
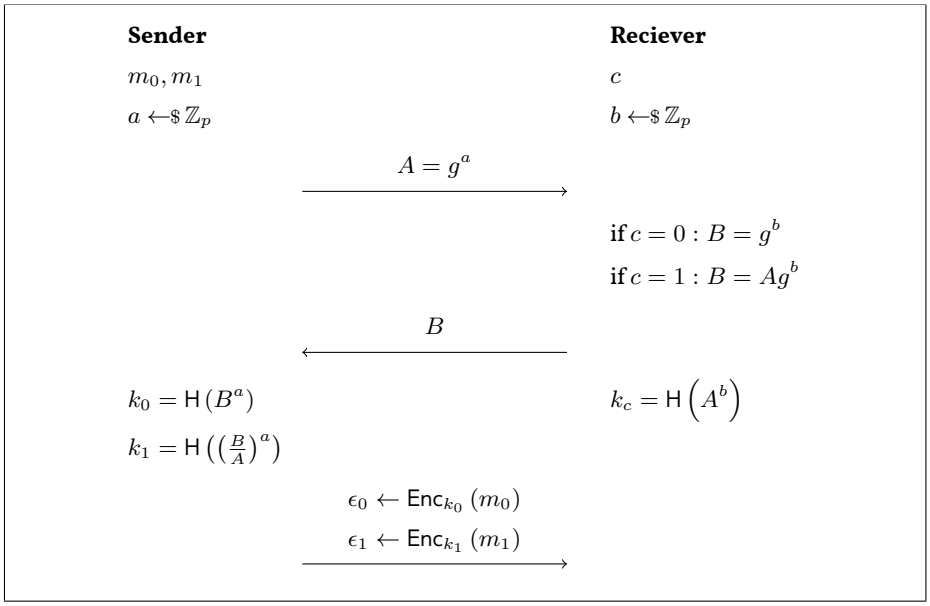


Figure 2.12: Schematic relationship between "seeded" OTs and OT Extension. "Seeded" OTs use the base OTs to extend the bit length, OT Extension uses them to provide many "cheap" OTs. Both variants are compatible.

[72] Yao (1986)

Protocol 2.1: "The Simplest Protocol for Oblivious Transfer" Chou and Orlandi (2015)



During the setup phase the garbler first translates the desired functionality into a Boolean Circuit. Next, the *garbling* takes place: The garbler assigns a random symmetric key to each possible (bit)value on every wire (i.e., k_0 for the bit value 0 on this wire and k_1 for bit value 1, accordingly). Now, the truth tables of the gates are garbled by doubly encrypting the output keys representing the logical outputs with the combination of both input keys, according to the gate's functionality. Lastly, the positions of truth table entries are shuffled to deny a reconstruction of the bit values due to the position of the decoded elements. The truth table of a (unpermuted) AND gate is shown in Table 2.1.

Table 2.1: Truth table of a garbled, unpermuted AND Gate for Yao's Garbled Circuits protocol

Input w_0	Input w_1	Output w_2	Garbled Value
$k_0^{w_0}$	$k_0^{w_1}$	$k_0^{w_2}$	$\mathsf{Enc}_{k_0^{w_0},k_0^{w_1}}(k_0^{w_2})$
$k_0^{w_0}$	$k_1^{w_1}$	$k_0^{w_2}$	$\mathsf{Enc}_{k_0^{w_0},k_1^{w_1}}(k_0^{w_2})$
$k_1^{w_0}$	$k_0^{w_1}$	$k_0^{w_2}$	$\mathsf{Enc}_{k_1^{w_0},k_0^{w_1}}(k_0^{w_2})$
$k_1^{w_0}$	$k_1^{w_1}$	$k_1^{w_2}$	$\mathsf{Enc}_{k_1^{w_0},k_1^{w_1}}(k_1^{w_2})$

This garbled circuit, that is the garbled truth tables[73], are sent alongside the keys corresponding to the garbler's secret inputs to the evaluator. This marks the beginning of the online phase. Due to the randomly assigned keys and the decoupling of keys and bit values achieved by the garbling, the evaluator cannot learn anything regarding the true input values from these keys. Using Oblivious Transfer (see Section 2.3.2), the evaluator acquires the keys corresponding to its own secret inputs. As now all first-layer input keys are known to the evaluator, he is now able to gate-wise decrypt the output keys, until the output wires are reached. In a last interaction step, the output keys of those wires are retranslated to bit values by the garbler, thus generating the plain text output.

[73] The function f, including its circuit representation, is publicly known.

Optimizations of Yao's GC

Since its original introduction, the original protocol has been heavily optimized, rendering more and more real-world applications computationally feasible. We will introduce three representative, influential optimizations: *Point-and-Permute*, *Free XOR*, and *Half-Gates*. Of course, GC also profits from efficient OT-Extension schemes, especially for large inputs of the evaluator.

Point-and-Permute

One drawback of garbled and permuted truth tables are, that the evaluator must decrypt up—in the worst case—to four entries to find the correct output key—on average still 2.5 entries. *Point-and-Permute*[74] eliminates this overhead to the cost of one additional (random) bit per wire, used as a *signal (or permutation) bit*. The combination of the signal bits of both input wires point to the correct truth table entry, hence only one entry needs to be decrypted. In addition, Point-and-Permute simplifies the output decryption, as the garbler just needs to reveal the permutation bits for each output wire to enable the evaluator to decrypt the output.

[74] Beaver, Micali, and Rogaway (1990)

Free XOR

The second optimization, Free XOR[75], further reduces the need to perform symmetric crypto operations while evaluating the garbled circuit. The garbler inserts correlation into the wire keys[76], such that the resulting key of an `XOR` is the `XOR` of the input keys. The evaluator never learns both keys for a wire, that way the randomly chosen fixed correlation remains secret. While reducing the computational cost of the evaluator this optimization incurs a cost: an additional (weak) cryptographic requirement, the circular 2-correlation robustness assumption, is added to the security model[77]. This is a very technical security assumption; suffice to say that it is weaker than the random oracle model.

[75] Kolesnikov and Schneider (2008)

[76] See Section 2.3.2

[77] Choi et al. (2012)

Half-Gates

Lastly, Half-Gates[78] reduce the required entries in the truth table of `AND` gates to two (previous state-of-the-art was three[79]), while remaining compatible with the Free-XOR optimization. The idea is, that each `AND` gate is "broken" into two halves for which each party knows one input. Each of these halves are garbled using one ciphertext. Unfortunately, instead of using only one symmetric crypto operation during evaluation (as it is the case with Point-and-Permute), the evaluator must perform two operations. However, benchmarks show, that this trade-off allows for most real-world circuits a significant increase in performance. A new advancement of this optimization—*Three-Halves Garbling* (3HG)[80]—that further reduces the required ciphertexts to $1.5 + 5$ bit per `AND` works by garbling the gates in such a way, that linear combinations of the slices of the input keys result in the correct output key. It trades higher computational complexity for less communication and is used in Chapter 3.

[78] Zahur, Rosulek, and Evans (2015)

[79] Naor, Pinkas, and Sumner (1999)

[80] Rosulek and Roy (2021)

Communication Rounds of Yao's GC

One characteristic of GC is, that it only performs a constant number of interactions between both parties: The initial transmission of the garbled circuit, the OT invocations for the input-key retrieval, and during the output reconstruction. During the online phase large numbers of symmetric cryptography operations must be performed, although in the optimized protocol variants only for AND gates. Thus, solely the *number* of AND gates is responsible for the (significant) performance cost.

2.3.4 Boolean and Arithmetic GMW

[81] Goldreich, Micali, and Wigderson (1987)

The GMW protocol[81], named after its founders Oded Goldreich, Silvio Micali and Avi Widgerson, was introduced in 1987, shortly after Yao's Garbled circuits. It exists in two variations, the first operation on Boolean Circuits and the second being an extension to algorithmic rings, working on Arithmetic Circuits.

Both variants work by *secret sharing* the input values, that is "breaking" it into multiple parts where each part in itself contains only random information and only by combining *all* shares of a value that secret can be reconstructed. The exact method of how to generate a share differs between the variants—implementing the same idea on different algebraic structures. There is an equivalency between the operations of both variants. Namely, additions on $\mathbb{Z}_p$ correspond to XOR operations on binary values (when ignoring a possible carry) and the multiplications corresponds to the logical (i.e., binary) AND operation.

In addition to the inner workings described below, modern implementations include many optimizations, such as efficient bit packing, usage of random OT, *Single Instruction Multiple Data* (SIMD) instruction vectorization, and many more (e.g., see Schneider and Zohner [82] and Braun et al.[83]).

[82] Schneider and Zohner (2013)

[83] Braun et al. (2021)

Boolean GMW

Boolean GMW operates, as the name implies, on Boolean Circuits, that is DAGs composed of AND and XOR operations. The wires signify single bit values and all inputs are bit values.

The secret values are secret shared between n parties by sampling $n-1$ uniformly *independent and identically distributed* (i.i.d.) bit values and *blinding* the secret value v for the last share by XOR-ing it with all generated random shares:

$$s_i \leftarrow\!\!\$\, \{0,1\}, \qquad \forall i \in \{1,\ldots,n-1\},$$
$$s_n \leftarrow \bigoplus_{i}^{n-1} s_i \oplus v.$$

The notation "$x \leftarrow\!\!\$\, \mathbb{D}$" signifies the random sampling of a value from domain $\mathbb{D}$.

This construction ensures, that the secret value is reconstructible by XOR-ing all shares: $v = \bigoplus_i^n s_i$.

One share is distributed to each party, such that every party holds one share of every party's input values. Note, this implies that every party retains one share of each own input value. This way the input privacy guarantees hold, even if all other $n-1$ parties collude.

The circuit's XOR operations are non-interactively, i.e., locally, computable as the XOR of all locally held shares equate to one share of the result. Without loss of generality consider the XOR operation between two parties p_1 and p_2 with secret inputs v^1 and v^2, respectively. The parties secret share their inputs (with random bits r_1 and r_2) and exchange one share, say s_2^i. The local operations yield:

Party p_1	**Party p_2**
$S^1 = s_1^1 \oplus s_2^2$	$S^2 = s_1^2 \oplus s_2^1$
$= r_1 \oplus v_2 \oplus r_2$	$= r_2 \oplus v_1 \oplus r_1$
Reconstruction:	
$S^1 \oplus S^2 = s_1^1 \oplus s_2^2 \oplus s_1^2 \oplus s_2^1$	
$= r_1 \oplus v_2 \oplus r_2 \oplus r_2 \oplus v_1 \oplus r_1$	
$= v_2 \oplus v_1$	

The computation of an AND requires an interactive protocol, e.g., in the two-party case the usage of *Multiplication Triples* (MT)[84]—sometimes called *Beaver Triples*: in the setup phase (see introductory text to this chapter) both parties generate a secret share triple a^i, b^i, c^i, with $i \in \{0, 1\}$ indicating the party, such that $c^1 \oplus c^2 = (a^1 \oplus a^2)(b^1 \oplus b^2) = a^1b^1 \oplus a^2b^1 \oplus a^1b^2 \oplus a^2b^2$. This is possible by performing a random OT protocol twice[85].

[84] Beaver (1991)

[85] Asharov, Lindell, et al. (2013)

These triples are used for the calculation of $x \wedge y$ in the online phase—x and y being secret shared, of course. Both parties exchange intermediary values $d^i = x^i \oplus a^i$ and $e^i = y^i \oplus b^i$. After that exchange both parties can reconstruct the plain text values of e and d. Note, that this reconstruction does *not* reveal any secret value, as they are still blinded with unknown random values.

In a last step, both parties are now able to calculate the result of the AND operation: $z^1 = d \cdot b^1 \oplus e \cdot a^1 \oplus c^1 \oplus d \cdot e$ and $z^2 = d \cdot b^2 \oplus e \cdot a^2 \oplus c^2$. As in the XOR case, the correctness is easily provable by reconstructing $z^1 \oplus z^2$ and substituting the equivalent terms.

With those two operations the circuit can be evaluated and the output reconstructed.

Arithmetic GMW

The Arithmetic GMW protocol is an extension to the Boolean case, operating on values in the (finite) algebraic ring $\mathbb{Z}_p$ with p elements and representing the desired functionality as arithmetic circuits consisting of multiplications and additions. Due to the underlying ring structure, we are dealing with modular arithmetic, i.e., multiplications are defined as $x \cdot y \mod p$ and additions as $x + y \mod p$. Arithmetic GMW is sometimes called *Arithmetic Sharing*.

The secret sharing procedure is mostly equivalent:

$$s_i \leftarrow\$ \mathbb{Z}_p, \qquad \forall i \in \{1, \ldots, n-1\},$$
$$s_n \leftarrow v - \sum_{i}^{n-1} s_i \mod p.$$

The sum of all shares reconstructs the secret value.

As in the Boolean case, additions can be computed locally, that is non-interactively, and the multiplication requires an interactive protocol, again based on an (arithmetic) MT for the two party case.

Communication Rounds in GMW

As discussed in the sections above, the GMW protocol requires multiple interaction rounds for most non-trivial computations[86]. These interactions can be "aggregated" according to the dependencies in the circuit, such that all interactions in a circuit "layer" are performed simultaneously. Hence, the relevant metric for assessing the performance of the protocol execution is not the number of AND/MUL gates—as in the GC case, but the *multiplicative depths* of the circuit—that is the number of AND/MUL gates on the longest path between input- and output gates.

[86] The joint computation of histograms over distributed data is a notable exception, as the only required interactions occur during secret sharing and reconstruction. All other operations can be performed locally. This is a main consideration for the EasySMPC project, cf. Wirth et al. (2021).

Furthermore, with the pre-computation of all MTs and the setup of OT-Extension beforehand, most computationally expensive tasks are performed in the setup phase, independent of the parties' input data. The online phase, while requiring communication for each layer of AND gates, only involves fast, local XOR/ADD operations.

2.3.5 Hybrid Protocols

As seen in the previous sections, the most efficient choice of a protocol[87] depends among other factors on the circuit layout (high multiplicative depths or large size), the types of operations (arithmetic operations or Boolean operations and comparisons), and the desired communication characteristics (constant number of large rounds or multiple smaller rounds).

[87] Assuming the same security level

Most non-trivial applications perform different subtasks requiring different protocols for maximum efficiency. It is possible to convert between different sharing types, however, most conversions incur additional computation and communication costs. For details regarding conversion methods and costs, see Demmler, Schneider, and Zohner [88].

[88] Demmler, Schneider, and Zohner (2015)

2.3.6 Outsourced Multi-Party Computation

Contrary to "full" n-party protocols with n parties participating in the computation, many applications can be designed to work in an outsourced computation model. In this model only a small subset of parties m perform the joint computation and the other parties are data providers. This model has multiple advantages:

- The complexity of the outsourced computation is $\mathcal{O}(m^2)$, instead of $\mathcal{O}(n^2)$. As usually $n \gg m$, e.g., $m = 2, n = 100$, outsourcing improves the efficiency by a factor $\mathcal{O}\left(\left(\frac{n}{m}\right)^2\right)$.
- The data providers do not participate in the calculation itself and are therefore not able to attack the computation—other than breaking the correctness by providing wrong or malformed inputs. Even if the computation is only secure against semi-honest adversaries, malicious *data providers* can not compromise privacy.
- By being able to freely choose the computing parties, choices can be made, e.g., to ensure high-bandwidth, low-latency network connections between the computation parties, hence, improving the overall performance.

These advantages come at a cost. An additional security assumption is added, as the computation parties are not allowed to collude. For a detailed analysis of outsourced MPC, see Kamara, Mohassel, and Raykova [89].

[89] Kamara, Mohassel, and Raykova (2011)

2.3.7 MPC Programming Frameworks

The circuit representations of non-trivial functions require many thousands to millions of gates[90]. The construction of those circuits "by hand" is infeasible. Fortunately, many compiler and programming frameworks allow the construction of MPC circuits at a higher abstraction level. Compilers, like Fairplay[91], CBMC-GC[92], or hardware synthesis based compiler[93] translate domain specific languages, standard `C` code, or hardware definition languages to circuits and provide optimization. Programming frameworks on the other hand provide some language programming interface to construct and evaluate circuits, often including the networking tasks as well. For an actively updated overview over MPC compiler and frameworks, see https://github.com/rdragos/awesome-mpc (accessed: 2022-05-19) and the repository https://github.com/MPC-SoK/frameworks (accessed: 2022-05-27) associated with Hastings et al.[94].

[90] The AES-128 circuit, for example, requires 36,663 gates and even a "simple" function, such as the 64-bit adder requires 376 gates. For details, see https://homes.esat.kuleuven.be/~nsmart/MPC/ accessed: 2022-05-19.

[91] Malkhi et al. (2004)

[92] Franz et al. (2014)

[93] Songhori et al. (2015); Demmler, Dessouky, et al. (2015)

[94] Hastings et al. (2019)

ABY

The secure two-party computation framework ABY[95] is a `C++` framework implementing three MPC protocols under the semi-honest adversary model: **A**rithmetic GMW, **B**oolean GMW, and **Y**ao's Garbled Circuits. Furthermore, it allows the efficient conversion between these protocols. The main focus of ABY is high performance and the ability to modify and optimize low-level primitives and building blocks. While providing state-of-the-art optimizations, this focal point leads to a comparably low abstraction level, requiring more MPC expert knowledge for the implementation of secure algorithms.

[95] Demmler, Schneider, and Zohner (2015)

MOTION

MOTION[96] is a full-threshold, semi-honest adversary n-party framework written in `C++`. It implements the n-party GMW protocol in both Boolean and arithmetic variant, as well as the BMR protocol and the conversions between those protocols. We extend this framework in Chapter 3 to provide Yao's Garbled Cir-

[96] Braun et al. (2021)

cuits in a two-party setting, these extensions will be included in the public version of the framework as well. Like ABY, MOTION favors high efficiency over a high abstraction level as well, however, newer software architecture patterns and an additional focus on modularity provide some abstractions for common uses without losing the ability to access and optimize low-level structures.

MP-SPDZ

MP-SPDZ[97] is a Python framework implementing many protocols across the MPC security space. This includes protocols against malicious adversaries with both honest and dishonest majority. Its main goal is to allow the performance comparison of a MPC functionality across many protocols. Providing functionalities for the Python programming language, it provides a high level of abstraction and the simple "switching" between different protocols. Being primarily a benchmarking framework, however, poses challenges for embedding MP-SPDZ as a MPC-providing component into full software applications.

[97] Keller (2020)

2.3.8 MPC and Data Protection Laws

As introduced in Section 2.2, the transfer and processing of sensitive (medical) or identifying data is regulated according to strict data protection laws. For many research questions in the medical realm, gathering the patients' informed consent is the only possibility to conduct a study involving the aggregation of many datasets—a difficult to impossible task in retrospective studies or for datasets including deceased patients. The high, provable security guarantees of MPC might allow researcher to tap into those previously unusable datasets for distributed computations, given that the computation results do not reveal individuals' data or risk the re-identification of patients. Unfortunately, the legal examination is still inconclusive at the time of writing.

Decisions of the European Court of Justice and the German Federal Court of Justice predating the GDPR suggest the possibility of record linkage[98] without informed consent, as encrypted data was only considered "personal data" for parties with access to the encryption key or third parties with the legal right to demand disclosure of said key[99]. Whether this ruling holds under GDPR is not assessed, yet. A more recent work[100] pursues a different reasoning: The authors argue, that the usage of—at least secret sharing based—MPC techniques does not constitute a transfer of data as defined in the GDPR. This assessment is dependent on many factors, such as the safety of the specific implementation, the safety of the outputs, and the processual framework. These additional considerations must be assessed on a case-by-case basis.

[98] See chapter 5

[99] The Court of Justice of the European Union (2016); Federal Court of Justice of Germany (2017)

[100] Helminger and Rechberger (2022)

2.3.9 Homomorphic Encryption

Homomorphic Encryption (HE) is a group of techniques providing *computation under encryption*—a party can perform computations on a ciphertext without decrypting it—based on the homomorphic properties of the underlying crypto system—that is $\mathsf{Enc}(m_1) \odot \mathsf{Enc}(m_2) = \mathsf{Enc}(m_1 \oplus m_2)$, where $\odot, \oplus$ denote operations specific to the used crypto system. For a detailed survey, see Halevi [101].

[101] Halevi (2017)

Based on what operations can be performed under encryption, HE schemes are grouped into multiple groups, the most important being *Somewhat Homomorphic Encryption* (SWHE) and FHE. SWHE can only perform some limited operations and is known for some time—starting with RIVEST, ADLEMAN, DERTOUZOS, ET AL.[102] and more explicitly with ELGAMAL [103] and PAILLIER [104]—while FHE can compute any operation. As we have seen in Section 2.3.4, being able to perform multiplication *and* additions is sufficient to calculate every function in an arithmetic circuit. SWHE schemes are only homomorphic with respect to one of those operations or are limited in the number of possible successive operations. For example, protocols based on the *Learning with Errors* (LWE) hardness assumption are limited in the number of operations, as they rely on the presence of noise for their security. The error accumulates or multiplies with each operation, leading to an eventual loss of the encrypted value at some point. The LWE-based fully homomorphic BGV protocol[105] mitigates this problem by specifically employing noise management techniques.

[102] Rivest, Adleman, Dertouzos, et al. (1978)

[103] ElGamal (1985)

[104] Paillier (1999)

[105] Brakerski, Gentry, and Vaikuntanathan (2014)

Paillier Cryptosystem

THE PAILLIER CRYPTOSYSTEM[106] is an additively homomorphic system, providing the following functionality: $\mathsf{Enc}(m_1) \cdot \mathsf{Enc}(m_2) \mod n^2 = \mathsf{Enc}(m_1 + m_2) \mod n^2$ and $\mathsf{Enc}(m_1)^k \mod n^2 = \mathsf{Enc}(k \cdot m_1)$, where n is the product of two large primes p and q. It is an asymmetric encryption scheme with a public key $\mathsf{pk} = (n, g)$ and a secret key $\mathsf{sk} = (p, q)$. The original paper describes two encryption functions: First, $m \mapsto g^m r^n \mod n^2$, where $r < n$ is a random value, and second, $m \mapsto g^{m+nr} \mod n^2$. Using the first function and two messages m_1, m_2 the identity

$$\begin{aligned}
&\mathsf{Enc}(m_1) \cdot \mathsf{Enc}(m_2) \mod n^2 \\
&= g^{m_1} g^{m_2} r_1^n r_2^n \mod n^2 \\
&= g^{m_1+m_2} (r_1 r_2)^n \mod n^2 \\
&= g^{m_1+m_2} r'^n \mod n^2 \\
&= \mathsf{Enc}(m_1 + m_2) \mod n^2
\end{aligned}$$

holds. However, using this encryption scheme there is no known non-interactive way to multiply two ciphertexts under encryption.

[106] Paillier (1999)

Fully Homomorphic Encryption

FHE HAS BEEN a long unsolved problem until the first fully homomorphic system designed by GENTRY [107]. However, while much more powerful, FHE is—to this day—computationally very expensive, hence, qualifying for only limited applications. In recent years, FHE has become feasible for more and more applications, dedicated programming libraries matured[108], and standardization efforts were undertaken[109].

[107] Gentry (2009)

[108] For example, Halevi and Shoup (2014), Dai and Sunar (2016), and Microsoft Research (2022)

[109] See https://homomorphicencryption.org/

2.3.10 Quantum Secret Sharing

As we have seen in Section 2.3.2, cryptography profited of ideas in the realm of physics by translating WIESNER 's conjugate coding to the cryptographic primitive of OT. However, the reverse can be observed as well—the re-convergence of physics and cryptography. One such example is *quantum secret sharing*. The first

work in this field by HILLERY, BUŽEK, AND BERTHIAUME [110] in 1999, describes the usage of entangled multi-particle quantum states to "split" a secret—either classical or a quantum state—in such a way, that only having *all* information allows the recombination of the secret.

[110] Hillery, Bužek, and Berthiaume (1999)

This field is an active field of research, e.g., with GOTTESMAN [111] deriving many theoretical insights, such as that "the size of each share in a quantum secret sharing scheme must be at least as large as the size of the secret"[112]—very similar to the OTP length restrictions—while when sharing classical secrets, each share can potentially be as small as half the classical secret's size. Furthermore, Z.-J. ZHANG, Y. LI, AND MAN [113] derived a scheme connecting n parties using single photons instead of multi-particle states. Given the promising results, it is to hope that the protocols and methods devised in this work may be implemented in quantum physical systems one day, even further optimizing the required communication.

[111] Gottesman (2000)

[112] Gottesman (2000)

[113] Z.-j. Zhang, Y. Li, and Man (2005)

Greenberger-Horne-Zeilinger (GHZ) States

THIS FIRST WORK uses a *Greenberger-Horne-Zeilinger* (GHZ) triplet to distribute a secret held by Alice between Bob and Charlie. The combination of three particles is the smallest form of GHZ state, larger entangled systems are possible as well. For a combination of n two-dimensional systems, the GHZ state is the superposition of all particles being in the one state and its inverse:

$$|\mathsf{GHZ}\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right).$$

The HBB Protocol

USING THESE GHZ states for quantum secret sharing is not directly straightforward and requires an additional classical channel between Alice and Bob and Alice and Charlie. Using quantum cryptography, the quantum channel is protected against eavesdropper "by default", as adversarial measurements can be detected as noise in the system, skewing the expected experiment probability distributions, hence, being detectable.

The protocol by HILLERY, BUŽEK, AND BERTHIAUME [114] begins by assuming, that all three parties have one particle of the GHZ triplet $|\psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$. All three parties decide randomly whether to measure their particle in the x or in the y direction. The chosen direction, but not the results of the measurements, are made public in the following way: Bob and Charlie announce their chosen direction to Alice, who then returns the directions of all three parties. This disclosure procedure mitigates a possibility for Bob or Charlie to cheat.

[114] Hillery, Bužek, and Berthiaume (1999)

The x and y eigenstates can be defined as:

$$|+x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |+y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle),$$
$$|-x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \quad |-y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle).$$

Using this notation and the relationships $|0\rangle = \frac{1}{\sqrt{2}}(|+x\rangle + |-x\rangle)$ and $|1\rangle = \frac{1}{\sqrt{2}}(|+x\rangle - |-x\rangle)$, we can express the triplet state as:

$$\begin{aligned}|\psi\rangle &= \frac{1}{2\sqrt{2}}[(|+x\rangle_a|+x\rangle_b + |-x\rangle_a|-x\rangle_b)(|0\rangle_c + |1\rangle_c) \\ &= (|+x\rangle_a|-x\rangle_b + |-x\rangle_a|+x\rangle_b)(|0\rangle_c - |1\rangle_c)].\end{aligned}$$

Given this decomposition of $|\psi\rangle$, one can create a table of Charlie's result (measuring in the x direction) based on the measurements of Alice and Bob—as displayed in Table 2.2.

Table 2.2: The influence of Alice's and Bob's measurements on the result of Charlie's measurement in the x direction following the HBB protocol by Hillery, Bužek, and Berthiaume (1999).

Bob \ Alice	$+x$	$-x$	$+y$	$-y$
$+x$	$\|0\rangle + \|1\rangle$	$\|0\rangle - \|1\rangle$	$\|0\rangle - i\|1\rangle$	$\|0\rangle + i\|1\rangle$
$-x$	$\|0\rangle - \|1\rangle$	$\|0\rangle + \|1\rangle$	$\|0\rangle + i\|1\rangle$	$\|0\rangle - i\|1\rangle$
$+y$	$\|0\rangle - i\|1\rangle$	$\|0\rangle + i\|1\rangle$	$\|0\rangle - \|1\rangle$	$\|0\rangle + \|1\rangle$
$-y$	$\|0\rangle + i\|1\rangle$	$\|0\rangle - i\|1\rangle$	$\|0\rangle + \|1\rangle$	$\|0\rangle - \|1\rangle$

Two things are visible from the table: First, from having its own measurement and the *direction* of measurements of Alice and Bob, Charlie can determine whether Alice's and Bob's state are correlated or anti-correlated, and, Second, when each party chooses the direction of measurement at random, Charlie chooses the right direction to gain any information regarding Alice's and Bob's state half of the time. By announcing *all* measurement directions, all parties know whether to discard or accept this measurement. The first thing is actually the completed secret sharing of this qbit: on his own Charlie knows nothing with regard to Alice's secret. However, when combining his information with Bob's, the secret state of Alice is revealed.

The original publication extends this protocol to sharing arbitrary quantum states and to four participating parties.

2.4 General Notation

Although, the individual chapters will introduce chapter-specific notation, some abbreviations and symbols are used throughout this dissertation:

The Boolean operations are concisely noted using logic symbols: $\wedge$ is `AND`, $\vee$ is `OR`, $\neg$ is `Not`, and $\oplus$ is `XOR`. `0`/`1` signify False/True. $|x|$ is used to indicate the length of a vector x, i.e., the number of elements.

Non-trivial variable names in protocols are written in sans serif, function names (and calls) `monospaced`.

Branching, implemented with `MUX` (multiplex) Gates, is written using ternary notation: condition **?** true statement **:** false statement. The MPC protocols Arithmetic GMW, Boolean GMW, and Yao's Garbled Circuit are abbreviated with $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{Y}$, respectively. Secret shared values are (mostly) written as $\langle x \rangle^S$, where $S \in \{\mathcal{A}, \mathcal{B}, \mathcal{Y}\}$. The symmetric security parameter is denoted with κ[115].

[115] Throughout this work $\kappa = 128$.

Different modes of arithmetic rounding are used: $\lceil x \rceil$ means rounding x up to the nearest integer, $\lfloor x \rfloor$ means rounding x down to the nearest integer, and, accordingly, $\lfloor x \rceil$ means rounding x to the nearest integer.

Finally, the uniform (random) sampling of a from a set A is denoted with $a \leftarrow\!\!\$\ A$.

Part II

Results

Chapter 3
Efficient Privacy-Preserving Epistasis Analysis

One field of medical research showing great progress in recent years is *personalized health*. A field, promising a significant paradigm shift in healthcare. The basic premise of personalized health is to develop and adapt novel drugs specifically based on the genetic profile of an individual patient, hence, providing better intervention outcomes and fewer side effects.

This field is founded on the analysis of vast amounts of genetic data, as the dependencies between observable traits, the *phenotype*, like the onset of a disease or the expression of a severe drug side effect, and the person's genetic information, the *genotype* can be complex and hard to qualify. By applying statistical analysis methods to those growing genomic databases correlations—or in rarer cases even causal links—between genetic variations and phenotypical traits can be found. Due to the decreasing cost, full genome sequencing is at the brink of becoming a standard procedure[1]. This would open up many research opportunities—while requiring strong data protection frameworks.

[1] Biesecker et al. (2021)

One method, the *Genome-Wide Association Study* (GWAS), leads to notable insights into the associations of genome and diseases. However, the expression of a phenotype, e.g., the outbreak of a disease, is often not linkable to a single genotypical variant (*allele*). The various biological processes form complex regulatory networks leading to non-linear gene-gene dependencies. Furthermore, environmental influences (*epigenetic effects*), that is gene-environment interactions, can drive or suppress important biological mechanisms.

By quantitatively analyzing those regulatory networks, the expression of certain proteins, enzymes, or other molecules can be probabilistically described, conditioned on the presence or absence of certain genetic features. Furthermore, the concentrations of these proteins, that is the balance between expression, degradation and transport processes, influence the probability of expressing other molecules. While an omniscient observer could model the relevant biological processes involved as this *Bayesian Network*, a graph model of interconnected conditional probabilities—for all human researchers large portions of the graph topology are hidden. The aim of methods for *Epistasis Analysis* (EA) are to identify probable interactions—edges—between those vertices—features like the presence or absence of gene variations, environmental factors, and so on—based on large databases of labeled graphs. In other words, the recognition of correlating pattern in those graphs. It is a practical application of the general problem of reconstructing *Probabilistic Graphical Model* (PGM) encoding the causal flow of conditional probabilities.

The number of possible edges quickly render the statistical tools used in GWAS computationally infeasible. Our goal in this project was first to reduce the vertex set by removing noisy and probably unrelated features. For this task we chose the *Relief-F*[2] feature selection algorithm, an iterative, multicategorical information gain maximization algorithm. This reduced dataset then was used to fulfill our second goal, the extraction of a probable edge set. For this goal we projected the high-dimensional dataset to one dimension using a technique called *Multifactor Dimensionality Reduction* (MDR)[3].

[2] Kononenko (1994)

[3] M. D. Ritchie et al. (2001)

From a privacy perspective, genomic data is highly sensitive, as it constitutes the ultimate personal identifier[4]. Furthermore, the statistical evaluation requires vast amounts of genomic data—posing a privacy risk for large numbers of patients. The utility loss by using anonymized data or approaches like statistical disclosure control[5] make those techniques difficult to apply to genomic data.

[4] Hamacher et al. (2020)

[5] Duncan (2011)

To retain full utility of the data and perform a full-precision computation while providing high levels of data privacy, we designed, implemented and experimentally evaluated *Practical Private Epistasis Analysis using MPC* (PEA), a *Secure Multi-Party Computation* (MPC) protocol for privacy-preserving EA. While MPC has been applied to many real-world application, the strong privacy guarantees come with acute limitations: MPC protocols require substantially more communication bandwidth and computational resources than the corresponding clear text analysis—often multiple orders of magnitude more. Hence, the design of efficient, optimized protocols for complex applications is challenging. PEA employs a composition of a feature selection protocol—either Relief-F or its variation *Tuned Relief-F* (TuRF)—and MDR for private and secure EA.

While the exploration of gene-gene and gene-environment interactions allowed notable insights using those algorithms, for example analyzing specific diseases[6], in statistical tests[7], or while adjusting them for novel challenges, like the amount of *Single Nucleotide Polymorphisms* (SNPs) in GWAS datasets[8], no prior research aimed to improve the privacy in decentralized, partitioned datasets.

[6] H. Zhang et al. (2017); Yang et al. (2017); Meng et al. (2017); Y. M. Cho et al. (2004); Liu et al. (2009)

[7] J. He et al. (2011); Lee et al. (2018)

[8] Kim and Park (2015)

This chapter of the dissertation draws upon work accepted for publication in Hamacher, K., Kussel, T., Schneider, T., Tkachenko, O. (2022) "PEA: Practical Private Epistasis Analysis using MPC". *ESORICS 2022*. The author was deeply involved in all aspects of the described work contributing significantly to the design, implementation, experimental setup, and manuscript of the publication. The author and O. Tkachenko contributed equally to this work.

3.0.1 Related Work

Recent work explored *Private Epistasis Analysis* (PEA)[9] and *Private Feature Selection* (PFS)[10] using *Differential Privacy* (DP), however, since DP relies conceptually on the trade-off between privacy and utility the achievement of adequate privacy levels in DP-based genomic analyses remains a well-known problem[11].

[9] Chen, X. Zhang, and R. Zhang (2019)

[10] T. T. Le et al. (2017)

[11] Naveed et al. (2015)

While several articles on private *Genome-Wide Association Study* (GWAS) have been published[12], there have—to the best of our knowledge—been no research efforts

[12] H. Cho, Wu, and Berger (2018); Tkachenko et al. (2018); Bonte et al. (2018)

investigating PEA or PFS outside statistical disclosure control, that is achieving exact results. While GWAS tries to answer similar research questions than EA, it only considered marginal probabilistic links—that is linear 1-*Single Nucleotide Polymorphism* (SNP) interactions.

Lastly, private genomic "utility" functions—for example private genome variant query protocols—have been studied extensively[13] with still ongoing research activities.

[13] Demmler, Hamacher, et al. (2017); Asharov, Halevi, et al. (2018); Schneider and Tkachenko (2019)

3.0.2 Our Contribution

This chapter describes PEA, a privacy-preserving protocol for epistasis analysis. While it enables novel opportunities for biomedical research, PEA provides the following research contributions:

- Design and implementation of the first secure protocol with fully retained accuracy for:
  - Relief-F [14] and TuRF[15], two popular feature selection algorithms EAs that complete in less than a day for real-world database sizes containing a=10,000 SNPs and L=100 records[16].
  - *Multifactor Dimensionality Reduction* (MDR)[17], a wide-spread exponential-time algorithm for EA with (extrapolated) runtimes of around three days for real-world database sizes—dimensioned as stated before. The communication of our private MDR (PMDR) protocol is independent of the number of records.
- New efficient, generic arithmetic GMW building blocks:
  - A $\binom{N}{1}$-OT[18]-based custom protocol for Arithmetic Greater Than (AGT)—that is, a GT operation on arithmetic shares, that, while incurring slightly more rounds of communication, achieves $1.5\times$ less communication than the current state-of-the-art[19].
  - Arithmetic Swap (ASWAP), a generalization of the Boolean swap gates by Kolesnikov and Schneider [20] for the arithmetic GMW protocol with $4\times$ less communication than the naïve design.
  - Batched versions of both building blocks introduced before with $\mathcal{O}(\kappa)$ less communication for a fixed bit-length, where κ is the symmetric security parameter.
- The first actual implementation of three-halves garbling[21], including its performance analysis. The analysis shows an unexpected slowdown, as three-halves garbling exhibits a higher degree of branching compared to the prior best garbling scheme. Nevertheless, the network communication still remains a protocol's bottleneck.
- Design and analysis of secure outsourcing for PEA, which considers settings with more than two input owners and adds less than 1% communication overhead.

[14] Kononenko (1994)
[15] Moore and White (2007)
[16] Chen, X. Zhang, and R. Zhang (2019)
[17] M. D. Ritchie et al. (2001)
[18] Kolesnikov and Kumaresan (2013)
[19] Rathee et al. (2020)
[20] Kolesnikov and Schneider (2008)
[21] Rosulek and Roy (2021)

3.1 Background

Being at the intersection of statistical physics, genetics, and cryptography, PEA requires some background information in all those fields. This section introduces the PEA-specific subjects not already described in Chapter 2. Throughout this chapter we use the notation displayed in Table 3.1

Table 3.1: Notation used throughout the description of PEA.

Genetics	
L	Number of combined loci, i.e., interaction depth
a	Total number of attributes, i.e., SNPs
A	The set of all attributes, i.e., SNPs
R	The set of all records, i.e., database records
r^j	Data record j
W	Weights generated by feature selection (see Section 3.1.4)
s	Number of cross-validation steps
λ	Locus
$g^{j,\lambda}$	Genotype of person j at locus λ
Δ	Distance metric in Relief-F (see Equation 3.3)
Secure Multi-Party Computation	
κ	Symmetric security parameter, κ=128 in PEA
N	Number of parties
M	Number of outsourcing (computation) servers
P_i	Party i
$\langle x \rangle^S$	Secret-share of value x in sharing $S \in \{\mathcal{A}, \mathcal{B}, \mathcal{Y}\}$
$\langle x \rangle_i^S$	Secret-share of value x in sharing S held by P_i
$\binom{N}{1}$-OT	1-out-of-N Oblivious Transfer

3.1.1 Genomic Primer

All living beings encode the "blueprints" for biological structures and functions in molecular forms—in most organisms in the form of *desoxyribonucleid acid* (DNA), a double-helical macromolecule. Each of the two helix strands consists of a sugar-phosphate backbone and a sequence of *nucleotides*—also called *bases*. The four nucleotides creating the DNA's alphabet are: **A**denine, **C**ytosine, **G**uanine, and **T**hymine. Usually, the bases between the complimentary DNA strands form Watson-Crick-pairs—base pairs—i.e., hydrogen bonds pairing preferably adenine with thymine and cytosine with guanine. This implies, that one strand redundantly encodes the same information as the other—as a complementary base sequence. DNA helices combined with other biomolecules form more involved macrostructures, such as chromatin and chromosomes.

The central dogma of molecular biology describes the processes necessary to form proteins based on the DNA "instructions". The process of *transcription* converts the *genotype*—the nucleotide sequence stored on the DNA—into a *messenger ribonucleid acid* (mRNA) molecule, a "working copy" of the specific information. In the *translation* the mRNA molecules are used by *ribosomes* to construct an *amino acid* sequence, thus, forming proteins. Proteins are the biomolecules required for function of the cell and organism. Ribosomes translate every *codon*—a triplet of

nucleotides on the mRNA—into one of the 20 standard amino acids. Additional codons encode start and stop symbols.

This relatively simple system drastically gains complexity by the introduction of *transcription factors*, proteins inhibiting or promoting the transcription of DNA regions, hence, forming complex regulatory networks[22] where the interaction of multiple genes influence the expression of some *phenotypes*—observable traits, like eye and hair color, or, as in this context, the occurrence of a disease. While often modeled as "Kauffman Networks"[23]—Boolean networks encoding the *state* of a gene as "switched on" or *switched of*—or continuous concentration networks with associated *ordinary differential equations* (ODEs) governing the dynamics of the system[24], these regulatory networks can be modeled as *Probabilistic Graphical Models* (PGMs)[25], due to the probabilistic nature of individual biological events—of course influenced by concentrations, temperature, steric effects, etc.

[22] Monod, Changeux, and Jacob (1963)

[23] Kauffman (1969)

[24] Del Vecchio and Sontag (2007)

[25] Larrañaga, Inza, and Flores (2005)

The human genome consists of roughly 3.2 billion base pairs[26] with a small variance of only 0.1 % of base pairs vary between two individuals [27]. Variations of specific *loci*—positions in the DNA sequence—are called *alleles*. As most base pairs are identical between individuals, it is more efficient to store only the variations in comparison to a common reference genome. A specific form of genomic variation is the *Single Nucleotide Polymorphism* (SNP). It is the change of exactly one nucleotide, for example "rs248"[28] denotes the change G → A on base position chr8:19953315. As humans are *diploid*, two alleles may occur for every locus—one on each of the two duplicate chromosomes. A useful shorthand notation for presence or absence of a gene variation or SNP is to denote the double occurrence of the major allele with "AA", the presence of both major and minor allele with "Aa"—the chromosome pair is unordered—and the presence of the minor allele on both chromosomes with "aa".

[26] In comparison, peas have a vastly larger genome with around 4.45 billion base pairs (Kreplak et al. (2019))

[27] Barbujani and Colonna (2010); Schuster et al. (2010)

[28] https://www.ncbi.nlm.nih.gov/snp/rs248 accessed: 2022-04-13

3.1.2 GWAS and Epistasis

Genome-Wide Association Study (GWAS) try to find correlations between phenotypical expressions and genotype traits. Often times that means linking the presence of specific SNPs to the probability of the occurrence of a specific disease, for example the first published GWAS[29] linked five SNPs to multiple mechanisms increasing the risk of myocardial infarction. As the name suggests, GWAS analyses the whole or at least a significant fraction of the genome, while *candidate-driven* analysis concentrate on specific genes, based on a model.

[29] Ozaki et al. (2002)

The *penetrance* of a SNP—the probability of a genome variant affecting the trait—is analyzed by performing statistical tests on large labeled datasets. Many statistical tests are used, from simple *odds ratio* analysis to more complex tests like a χ^2 *hypothesis test*[30]. By comparing the most correlated loci to known regulatory pathways, GWAS can provide a starting point for determining causal links between genotype and phenotype[31].

[30] M. H. Wang, Cordell, and Van Steen (2019)

[31] Newton-Cheh et al. (2009)

Section 3.1.1 briefly describes regulatory networks. Unsurprisingly, many diseases—especially complex systematic diseases like cancer—do not only depend on one single genetic variation, but are caused by the non-linear interplay

of many factors, such as the (simultaneous) presence of one SNP and the absence of another while a third locus is *methylated*. This *gene-gene* and *gene-environment* interaction is called *epistasis*. The analysis of epistasis effects becomes computationally expensive very quickly, as combinatorics very quickly expand the search space. For gene-gene interaction, the algorithmic complexity scales exponentially in the number of interactions considered—a parameter we call *interaction depth*.

As a result, exact and exhaustive analysis methods are only practical for small fractions of the genome or—as for GWAS—low interaction depths. In PEA, the two feature-selection algorithms *Relief-F*[32] and TuRF[33] (see Section 3.1.4), as well as the dimension reduction algorithm MDR[34] (see Section 3.1.5) are used to achieve practical runtimes for privacy-preserving epistasis analysis.

[32] Kononenko (1994)
[33] Moore and White (2007)
[34] M. D. Ritchie et al. (2001)

3.1.3 Probabilistic Graphical Models

The seminal work of Pearl [35] started the notion of a *Probabilistic Graphical Model* (PGM). PGMs provide a general framework for many aspects of probabilistic theory, namely the *representation* and *modelling* of probabilistic relationships, the *inference* based on statistical models, and the *learning* of said models based on empirical evidence. PGMs come in a variety of forms and are useful tools in many disciplines, such as physics[36], bioinformatics[37], and machine learning[38].

[35] Pearl (1988)
[36] Pelizzola (2005)
[37] Joung and Fei (2009)
[38] Arnab et al. (2018)

Generally, PGMs represent the probabilistic relationships between states or entities in a graph structure. Undirected graphs are called *Markov networks*, directed graphs are called *Bayesian networks*[39]. The (computationally and epistemologically) most basic Bayesian network is a *Directed Acyclic Graph* (DAG).

[39] Note, that "Bayesian network" are called that way because of their usage of Bayes' rule for inference. It implies no usage of Bayesian statistics in the stricter sense—in fact conditional probabilities are often acquired using frequentist methods.

Multiple extensions exist, for example a *Cyclic Bayesian network* allows the formation of cycles in a directed network or *Dynamic Bayesian networks*, introducing a temporal dimension and generalizing *Hidden Markov Models* and *Linear Dynamic Systems*, further introducing Gaussian models and Kalman filters into the PGM domain. Of course, various non-linear variants, mixture models, hierarchical models, and combinations of all those are used in different applications.

One de-facto default toy example of a (simple) Bayesian PGM is the student model in Figure 3.1, described by Koller and Friedman [40]. In this model, it is described, how the difficulty of a test and a student's intelligence determine the grade of a test. The grade in turn determines the probability of the professor giving a favorable letter of recommendation. Additionally, the student's intelligence also influences their SAT score.

[40] Koller and Friedman (2009)

The graph representation makes it easy to locate statistically independent random variables and to construct joint probabilities using the *chain rule for Bayesian networks*—for the example $P(I, D, G, S, L) = P(S|I)P(L|G)P(G|I, D)P(I)P(D)$. Once constructed, both *causal reasoning*, where "upstream" observations are used to infer causally linked outcomes and *evidential reasoning*, where "outcomes" are used to predict cause probabilities, can be constructed "mechanically".

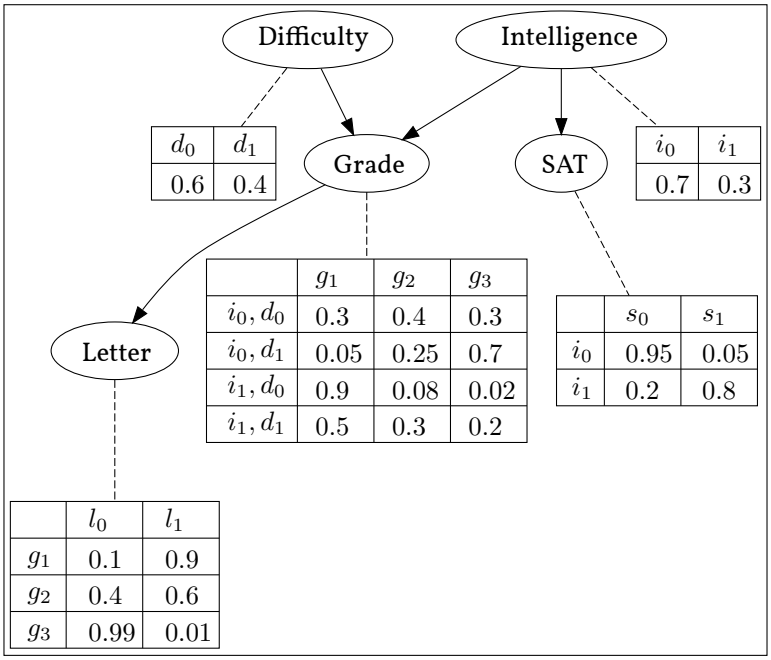


Figure 3.1: Example Bayesian network, taken from Koller and Friedman (2009). The conditional probability distributions for each node state are shown in the tables.

Example inference

The probability of a student totally unbeknownst to us to get a favorable letter of recommendation is $P(l_1) \approx 0.502$. If, somehow, the information that the student's intelligence is not up to par—that is $I = i_0$—is gained, the probability becomes $P(l_1|i_0) \approx 0.389$—an example of causal reasoning. An example of evidential reasoning would be a hiring manager seeing only the students test results—a disappointing g_3. Based on the available information—the grade—the probability of the student being intelligent is $P(i_1|g_3) \approx 0.079$. At the same time the probability that the test was difficult rises to $P(d_1|g_3) \approx 0.629$.

EA in this formulation is a learning task, that is a model's topology and conditional probability distributions is to be inferred based on observed samples. Of course, the networks for EA are much more complex, vastly larger and possibly cyclic. With unknown structure and (hopefully) full observability—that is all relevant genome loci are measured—the EA objective of structure reconstruction is—even when only allowing acyclic solutions—$\mathcal{NP}$-hard[41].

[41] This is evident, as the number of DAGs with N nodes is super-exponential in N.

To nevertheless tackle the problem efficiently, we first employ a feature selection algorithm to reduce the problem space and then employ a heuristic dimensionality reduction.

3.1.4 Feature Selection

Typical epistasis studies require vast amounts of genomic data to reach significant results, often thousands of patients, each contributing hundreds of thousands or even millions of SNPs. In comparison, single digits to a few dozen SNPs are connected to the phenotype of interest—the overwhelming majority of features are "noise" with respect to the research question. To make matters worse,

all those unrelated SNPs contribute massively to the combinatorial explosion of the search space. Hence, feature selection algorithms are used as a heuristic to select the potentially most significant SNPs. Typical estimators like *information gain*[42], *Gini index*[43], and *j-measure*[44] assume independence between attributes—an assumption not true in the case of epistasis, by definition.

[42] Hunt, Marin, and Stone (1966)
[43] Breiman et al. (2017)
[44] Smyth and Goodman (1990)

Relief Algorithm

Kira and Rendell [45] introduced *Relief*, a feature selection algorithm for nominal and numerical features in two-class classification problems. Relief proved to work efficiently in estimating the intended information gain for dependent and (lightly) noisy data. The key concept of Relief is that important attributes differ between records in different classes but show similar values for records in the same class. The algorithm creates weights W for the attributes A and "neighboring" records i, j with classes α according to:

[45] Kira and Rendell (1992)

$$W(A) = P(A_i \neq A_j | \alpha_i \neq \alpha_j) - P(A_i \neq A_j | \alpha_i = \alpha_j) \tag{3.1}$$
$$= P(A_i = A_j | \alpha_i = \alpha_j) - P(A_i = A_j | \alpha_i \neq \alpha_j). \tag{3.2}$$

A neighboring *instance*—that is, record—in the same class is called *near hit* and a neighboring instance from the complimentary class is called *near miss*. The neighborhood (and as we will see the conditional probability) is estimated using a definable *difference function* $\Delta(A, r^i, r^j)$. In the most basic case, that is for nominal values with two possible classes and no missing values, the difference function used here and in PEA is defined as:

$$\Delta(A, r^i, r^j) := \begin{cases} 1 & \text{if } A^i \neq A^j \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

Relief-A to Relief-F

Kononenko [46] extends Relief in various dimensions. The first variation—*Relief-A*—increases the algorithm's robustness against noisy datasets. "Noisy" means in this context, that a certain percentage of records are mislabeled. This robustness is achieved by not only considering the nearest hit-and-miss, but the k-neighborhood around a chosen instance. By introducing a shorthand notation $P_{\text{eqval}}, P_{\text{sameclass}}$ and using Bayes rule on Equation 3.1, it can be shown, that the weights calculated by Relief-A are highly correlated to the Gini index G and, hence, information gain[47]:

[46] Kononenko (1994)

[47] For the full derivation see Kononenko (1994).

$$W(A) = \frac{P_{\text{eqval}} G(A)}{P_{\text{sameclass}}(1 - P_{\text{sameclass}})}, \text{ where}$$
$$G(A) = \sum_V \left(\frac{P(V)^2}{\sum_V P(V)^2} \cdot \sum_C P(C|V)^2 \right) = \sum_C P(C)^2.$$

Furthermore, the correlation between the measures increase with increasing k, the number of considered neighbors. The author shows, that for dependent attributes the estimation quality of Relief-A exhibits a maximum as it first increases and then, when k gets so large, that records from other "clusters" in the distribution space are taken into account, decreases again. In accordance to the literature body using Relief and "mdr", we use $k = 10$ throughout this chapter. To

further increase the performance in presence of class noise, KONONENKO note, that by increasing m, the number of randomly sampled records used in the calculation (see Protocol 3.1), the robustness increases. This causes him—and following works using Relief—to choose a maximum m and iterate over *every* record.

Input: Attributes $A = A_1, \ldots, A_a$,
Records $R = r^1, \ldots, r^n$
$W_i \leftarrow 0, \forall i$
for $i = 1 \ldots m$ **do**
  `// randomly select record`
  $r^i \leftarrow\!\!\$\ [r^1, r^n]$
  $H, M \leftarrow \texttt{kNN}(r^i, k)$
  for $j = 1 \ldots k$ **do**
    for $l = 1 \ldots a$ **do**
      $W_l \leftarrow W_l - \Delta(A_l, r^i, H_j)/m + \Delta(A_l, r^i, M_j)/m$
return W

Protocol 3.1: Relief-F/A protocol. The difference function Δ used in those variations is shown in Equation 3.3.

By modifying the Δ difference functions, KONONENKO introduces Relief-B to Relief-D, which are able to deal with missing values. For example, the best performing Relief-D uses

$$\Delta(A, r^i, r^j) = 1 - P(A^j|\alpha^i)$$

if—for example—r^1 has unknown value, and

$$\Delta(A, r^i, r^j) = 1 - \sum_{V}^{\#\text{values}(A)} (P(V|\alpha^i)P(V|\alpha^j))$$

if both records are missing the value for attribute A.

Lastly, he extends the algorithm to *multi-class* problems with Relief-E and Relief-F. Relief-F uses the average of one near miss of every class for the weight calculation and weights this contribution, again, using the prior probability of each class:

$$W(A) = W(A) - \Delta(A, r^i, H)/m + \sum_{C \neq \alpha^i} P(C)\Delta(A, r^i, M(C))/m.$$

While most following work using a Relief variation and MDR for EA claim to use *Relief-F*—often written without the hyphen—the binary class case of *affected by disease / not affected*, technically, represents Relief-A. However, Relief-F turns into Relief-A in the two-classes case. To follow the nomenclature of the literature body, we designate the used variant in PEA as "Relief-F".

Tuned Relief-F

A SUBSEQUENT OPTIMIZATION, *Tuned Relief-F* (TuRF)[48], iteratively performs the Relief-F algorithm and after each chosen record it prunes the least significant attributes—the SNPs with the least weight. This (potentially) speeds up the computation as the number of features decreases for each iteration and increases the robustness against noisy attributes, as badly performing attributes do not influence subsequent iterations. This variation is shown in Protocol 3.2. The details of our privacy-preserving implementations of Relief-F and TuRF are given in Section 3.2.

[48] Moore and White (2007)

Protocol 3.2: TuRF protocol

Input: Attributes $A = A_1, \ldots, A_a$,
Records $R = r^1, \ldots, r^n$
for $i = 1 \ldots n$ **do**
  $W \leftarrow \texttt{ReliefF}(A, R)$
  $W \leftarrow \texttt{sort}(W)$
  `// remove last` α/n `attributes`
  $W \leftarrow W[0 : a - (\alpha/n)]$
  $A \leftarrow A[0 : a - (\alpha/n)]$
return W

3.1.5 MDR

[49] M. D. Ritchie et al. (2001)

Multifactor Dimensionality Reduction (MDR)[49] is a model-free, non-parametric statistical method for dimensionality reduction, explicitly developed for the detection and modelling of epistasis. Since its introduction in 2001, it has become one of the standard methods in EA used to successfully identify higher-order gen-gen interactions linked to the onset of diseases, such as sporadic breast cancer, essential hypertension[50], type 2 diabetes[51], and coronary atrial calcification[52].

[50] Meng et al. (2017)
[51] Y. M. Cho et al. (2004)
[52] Liu et al. (2009)

The idea behind the algorithm is to reduce the interaction dimensionality to one by categorizing groups of loci into high and low risk combinations. These one-dimensional combinations are ranked measuring classification and prediction errors. To avoid typical artifacts caused by the statistical method itself, *Leave-one-out cross validation* is usually employed. That means that the dataset is partitioned into n equally large sets and the model is trained on $n-1$ of those sets. The last remaining partition is used to determine prediction error. This process is repeated for all partitions. The final model error is the average of all prediction errors. Figure 3.2 shows an exemplary overview of the method.

3.2 Private Relief-F and Tuned Relief-F Feature Selection

Section 3.1.4 described the principle of the employed feature selection algorithms: relevant features—features useful for the distinction between classes—are given a high weight, irrelevant features are reduced in weight. PEA implements both *Private Relief-F* (PRelief-F) and *Private Tuned Relief-F* (PTuRF). The description of PRelief-F is shown in Protocol 3.3 and the description of PTuRF—as it is comparatively similar—is shown in Protocol C.1 in Appendix C.1. As TuRF—and PTuRF—iteratively prunes features, the ordering of records might introduce a sampling bias. To avoid this, we randomly shuffle the datasets before feature selection. As both algorithms make extensive use of comparisons and the k *Nearest Neighbors* (kNN) sorting (cf. Section 3.2.1) generates a circuit with linear (multiplicative) depth in the number of records, the feature selection is most efficiently implemented using Yao's Garbled Circuits[53]. For that, we employ the—to our knowledge—first implementation of the *three-halves garbling*[54] (cf. Section 3.5.1).

[53] Demmler, Schneider, and Zohner (2015)
[54] Rosulek and Roy (2021)

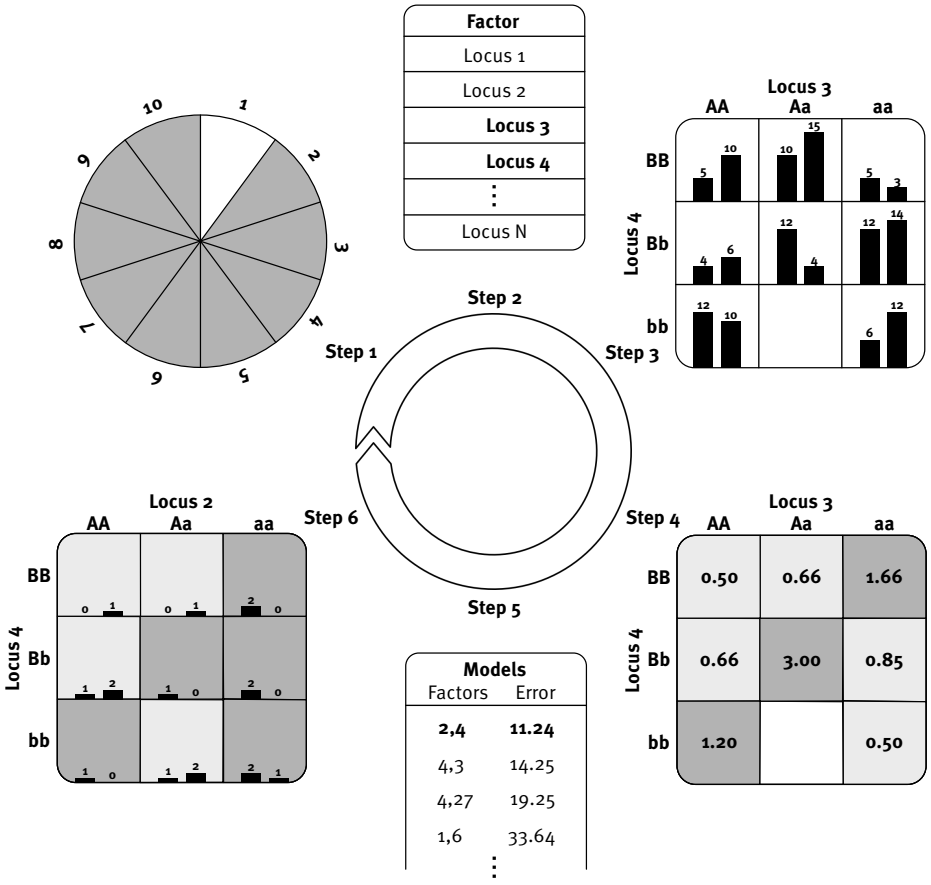


Figure 3.2: High-level exemplary visualization of the MDR analysis method (adapted from M. D. Ritchie et al. (2001)).
Step 1 consists of the partitioning of the data into a training set (e.g., 90 % of the data) and a cross-validation set (e.g., 10 % of the data).
In Step 2, possible combinations are chosen from all possible locus combinations. In this figure, combinations between two loci are shown.
Step 3 shows the number of cases and controls in the (usually) high-dimensional dataset.
In Step 4, all combinations where the ratio of affected and unaffected patients exceed a configurable threshold are labled as "high-risk".
Step 5 ranks the models using the misclassification error.
The prediction error of the best model is then tested against the cross-validation set in Step 6.
Steps 1 trough 6 are repeated for every cross-validation interval. The bars represent hypothetical distributions of affected (left) and unaffected (right) patients. Light gray shaded cells indicate low-risk combinations, dark gray shaded cells high-risk combinations. White cells show combinations with no observed occurrences.

The PTuRF implementation allows for an optional approximation: Instead of recalculating the distances between features in each iteration, it might be considered constant. As only a small fraction of features is removed in each iteration, the introduced error is small, while reducing the identification cost of the algorithm.

3.2.1 Private kNN

Relief, as well as the Relief-A to F variants, require the identification of the nearest or the k nearest neighbors for all classes. PRelief-F and PTuRF us an adapted version of the linearly scaling kNN clustering described by Järvinen et al.[55]. The distance metric used for sorting the records is the *Hamming distance*, as it runs with comparatively low runtime cost and performs well with nominal features.

[55] Järvinen et al. (2019)

3.2.2 Hamming Distance

The *Hamming distance* between two bit vectors x and y is the number of set bits in the element wise conjunction: $\mathrm{Hd}(x, y) = \sum_i x_i \wedge y_i$. This procedure is not useful for PRelief-F and PTuRF for two reasons: First, we need the negation of the element wise `AND`—the `NAND`, as per the definition of the Δ distance function in Equation 3.3—and second, we need to reuse parts of the computation for the weight calculation, namely the bit vector resulting from the element wise `NAND`.

Protocol 3.3: PEA's Private Relief-F protocol

Function `PReliefF`(R, φ)**:**

The dataset R is the concatenation of each data owner's P_i raw dataset R_i.

The dataset consists of all records $R := (r^1, \ldots, r^k)$, where the record $r^j := ((g^{j,1}, \ldots, g^{j,m}), \alpha^j) : r^j \in R$ with each genotype $g^{j,\lambda} \in \{1, 2, 3\}$ of person j at locus λ and each group $\alpha \in \{+, -\}$, denotes the case and control group, respectively. The function returns the index positions of the most weighted genotypes. $\varphi = 1 - \alpha/a$ denotes the ratio of attributes to return.

for $j = 1 \ldots k$ **do** `// For all records in the Dataset`

`// Initialize distance and difference matrices to the`
`// numerical maximum value and zero, respectively`

$\langle m^{\text{hit}}_{\text{dist}} \rangle^{\mathcal{Y}} \leftarrow [\langle \text{MAX\_VALUE} \rangle^{\mathcal{Y}}, \ldots, \langle \text{MAX\_VALUE} \rangle^{\mathcal{Y}}]$

$\langle m^{\text{hit}}_{\text{ineq}} \rangle^{\mathcal{Y}} \leftarrow [[\langle 0 \rangle^{\mathcal{Y}}, \ldots, \langle 0 \rangle^{\mathcal{Y}}], \ldots, [\langle 0 \rangle^{\mathcal{Y}}, \ldots, \langle 0 \rangle^{\mathcal{Y}}]]$

$\langle m^{\text{miss}}_{\text{dist}} \rangle^{\mathcal{Y}} \leftarrow [\langle \text{MAX\_VALUE} \rangle^{\mathcal{Y}}, \ldots, \langle \text{MAX\_VALUE} \rangle^{\mathcal{Y}}]$

$\langle m^{\text{miss}}_{\text{ineq}} \rangle^{\mathcal{Y}} \leftarrow [[\langle 0 \rangle^{\mathcal{Y}}, \ldots, \langle 0 \rangle^{\mathcal{Y}}], \ldots, [\langle 0 \rangle^{\mathcal{Y}}, \ldots, \langle 0 \rangle^{\mathcal{Y}}]]$

for $i > j$ **do** `// For all pairs of records`

$\langle D_{ji} \rangle^{\mathcal{Y}} \leftarrow \emptyset$

for $\lambda = 1 \ldots m$ **do** `// For all genotypes`

$\langle D_{ji} \rangle^{\mathcal{Y}}$**.append** $\left(\Delta\left(\langle g^{j,\lambda} \rangle^{\mathcal{Y}}, \langle g^{i,\lambda} \rangle^{\mathcal{Y}}\right)\right)$

for $\forall i \neq j$ **do** `// For all (unordered) pairs`

if $j < i$ **then**

$\langle d \rangle^{\mathcal{Y}} \leftarrow \text{Hw}(\langle D_{ji} \rangle^{\mathcal{Y}})$

else

$\langle d \rangle^{\mathcal{Y}} \leftarrow \text{Hw}(\langle D_{ij} \rangle^{\mathcal{Y}})$

if $\langle \alpha^j \rangle^{\mathcal{Y}} == \langle \alpha^i \rangle^{\mathcal{Y}}$ **then** `// If records have same label`

$\langle m^{\text{hit}}_{\text{dist}} \rangle^{\mathcal{Y}}, \langle m^{\text{hit}}_{\text{ineq}} \rangle^{\mathcal{Y}} \leftarrow \texttt{kNN}(\langle m^{\text{hit}}_{\text{dist}} \rangle^{\mathcal{Y}}, \langle m^{\text{hit}}_{\text{ineq}} \rangle^{\mathcal{Y}}, \langle d \rangle^{\mathcal{Y}}, k)$

else

$\langle m^{\text{miss}}_{\text{dist}} \rangle^{\mathcal{Y}}, \langle m^{\text{miss}}_{\text{ineq}} \rangle^{\mathcal{Y}} \leftarrow \texttt{kNN}(\langle m^{\text{miss}}_{\text{dist}} \rangle^{\mathcal{Y}}, \langle m^{\text{miss}}_{\text{ineq}} \rangle^{\mathcal{Y}}, \langle d \rangle^{\mathcal{Y}}, k)$

$\langle W \rangle^{\mathcal{Y}} \leftarrow \langle W \rangle^{\mathcal{Y}} + \langle m^{\text{miss}}_{\text{ineq}} \rangle^{\mathcal{Y}} - \langle m^{\text{hit}}_{\text{ineq}} \rangle^{\mathcal{Y}}$

for $\forall j$ **do**

`// The features are sorted by weight and only the first`
`// (best)` $\varphi \cdot a$ `are retained`

$\langle g'^j \rangle^{\mathcal{Y}} \leftarrow \texttt{kNN}(\langle g^j \rangle^{\mathcal{Y}}, \langle W \rangle^{\mathcal{Y}}, \varphi \cdot a)$

$\langle r'^j \rangle^{\mathcal{Y}} \leftarrow (\langle g'^j \rangle^{\mathcal{Y}}[1 : \varphi \cdot a]), \langle \alpha^j \rangle^{\mathcal{Y}})$

$\langle R' \rangle^{\mathcal{Y}} := \{\langle r'^1 \rangle^{\mathcal{Y}}, \ldots, \langle r'^k \rangle^{\mathcal{Y}}\}$

return $\langle R' \rangle^{\mathcal{Y}}$

To accommodate both arguments, we first calculate all absolute values of the differences between the genomes (Protocol 3.3, line 12) and calculate the *Hamming weight* of this resulting bit vector (lines 15 and 17), that is The Hamming distance to a zero vector $\mathrm{Hw}(x) = \sum_i x_i$, where $x_i \in \{0, 1\}$.

3.3 Private Multifactor Dimensionality Reduction

The base operations of *Private Multifactor Dimensionality Reduction* (PMDR) (Protocol 3.4) are aggregations of (integer) allele frequencies and prediction errors. Hence, arithmetic secret sharing—allowing the non-interactive, local aggregation of values—is the obvious protocol choice. However, the required "utility functions"—comparisons and swaps—are so much more efficient in a protocol operating on Boolean circuits, that the evaluation of PMDR in pure Boolean sharing is faster, than the arithmetic version which converts to Boolean sharing for those operations. To optimize the computation, we develop two novel arithmetic building blocks, enabling us to construct PMDR in arithmetic secret sharing with two orders of magnitude improvements over the pure Boolean construction—see Section 3.6.3. A local pre-processing aggregating the counts of (locally) occurring combinations is performed (Protocol 3.5).

3.3.1 Secure Arithmetic Greater Than

Boolean *Greater Than* (GT) gates can be constructed either optimized for multiplicative size—requiring ℓ AND gates[56]—or for multiplicative depth—requiring $3\ell - \lceil \log_2 \ell \rceil - 2$ AND gates at an AND depth of $\lceil \log_2 \ell \rceil + 1$. In this section, we will introduce a baseline protocol for the *Arithmetic Greater Than* (AGT) operation, an optimized protocol for small bit lengths, and extend this optimized protocol to arbitrary bit length. We compare our novel AGT protocol to the current state-of-the-art[57], which efficiently compares Boolean values. A version of our novel protocol for batch operation (Appendix C.2), as well as the security discussion (Appendix C.3) is given in Appendix C.

[56] Kolesnikov, Sadeghi, and Schneider (2009)

[57] Rathee et al. (2020)

Baseline Arithmetic Greater Than Protocol

A baseline protocol for an AGT operation comparing two integers $x_0, x_1 \in \mathbb{Z}_{2^\ell} : x_0, x_1 < 2^{\ell-1}$ in arithmetic sharing is shown in Protocol 3.6. It requires the Boolean re-sharing of ℓ bits for both the garbler and the evaluator—2ℓ in total—and $\ell - 1$ AND gates. Overall, this protocol requires $\ell(4.5\kappa + 5) - 1.5\kappa - 5$ bits of communication—κ is the chosen security parameter. Of this total size, the re-sharing takes up $3\ell\kappa$ bits and the following sum (in Yao's GC) takes $(\ell - 1) \cdot (1.5\kappa + 5)$ bits. Using 1-out-of-N OT[58] and the insight, that only the *Most Significant Bit* (MSB) is required, we will improve this protocol. However, this version provides a performance baseline for evaluating the novel, optimized construction. Note, that this baseline version requires only one round of communication and is restricted to input values smaller than $2^{\ell-1}$.

[58] Kolesnikov and Kumaresan (2013)

Optimized AGT Construction with Low Communication

The optimized AGT protocol is inspired by constructions by Dessouky et al.[59] and Rathee et al.[60]. As mentioned before, the general idea for optimization is, that it is sufficient to compute the MSB of the difference to determine the

[59] Dessouky et al. (2017)

[60] Rathee et al. (2020)

Protocol 3.4: Private MDR protocol. The Count used in line 10 is the local pre-processing described in Protocol 3.5.

1 **Function** PMDR($\boldsymbol{R_i}$):
2 Each data owner P_i locally randomly permutes its raw dataset $\boldsymbol{R}_i := (r_i^1, \ldots, r_i^{k_i})$, where the record $r_i^j := ((g_i^{j,1}, \ldots, g_i^{j,a_i}), \alpha_i^j) : r_i^j \in \boldsymbol{R}_i$ with each genotype $g^{j,\lambda} \in \{1,2,3\}$ of person j at locus λ and each group $\alpha \in \{+,-\}$, denotes the case and control group, respectively, and splits it into s equal parts $\boldsymbol{R}_i^1, \ldots, \boldsymbol{R}_i^s$.
3 **for** $j = 1 \ldots s$ **do** // For each of s cross-validation steps
4 **for** $i = 1 \ldots N$ **do** // Each party splits its dataset for cross-validation.
5 $\boldsymbol{R}_i^{\text{val}} := \{r_i^j\}_{j=|\boldsymbol{R}_i|\cdot j/s}^{|\boldsymbol{R}_i|\cdot(j+1)/s}$
6 $\boldsymbol{R}_i^{\text{test}} := \boldsymbol{R}_i \setminus \boldsymbol{R}_i^{\text{val}}$
7 **for** $\lambda_1 \in [a]$ **do** // For each pair of loci λ_1 and λ_2
8 **for** $\lambda_2 \in [a] \setminus \{\lambda_1\}$ **do**
// Each party locally counts the observed genotypes
// for test ($\boldsymbol{T}$) and validation ($\boldsymbol{V}$) sets.
9 **for** $i = 1 \ldots N$ **do**
10 $(\boldsymbol{T}_i^{\lambda_1,\lambda_2,+}, \boldsymbol{T}_i^{\lambda_1,\lambda_2,-}, \boldsymbol{V}_i^{\lambda_1,\lambda_2,+}, \boldsymbol{V}_i^{\lambda_1,\lambda_2,-}) \leftarrow$ Count$(\boldsymbol{R}_i^{\text{test}}, \boldsymbol{R}_i^{\text{val}}, \lambda_1, \lambda_2)$
// All parties share and aggregate their counts using Arithmetic
// sharing. Remark: Sharing is done locally using a PRG.
11 $\langle \boldsymbol{X}^{\lambda_1,\lambda_2,g} \rangle^{\mathcal{A}} \leftarrow \sum_{i=1}^{N} \langle \boldsymbol{X}_i^{\lambda_1,\lambda_2,g} \rangle^{\mathcal{A}}$ for $\boldsymbol{X} \in \{\boldsymbol{T}, \boldsymbol{V}\}$ and $g \in \{+,-\}$
// Compute the high risk prediction model as Boolean matrix $\boldsymbol{H}^{\lambda_1,\lambda_2}$.
// If #cases/#controls is greater than a public threshold $t_h = t_h^+/t_h^-$,
// the cell that corresponds to the genotype combination (i,j) is
// marked as high risk, indicated with $\langle 1 \rangle^{\mathcal{B}}$.
12 **for** $i, j \in \{1,2,3\}$ **do** // For each combination of genotypes
// This is equivalent to computing $(\boldsymbol{T}^{\lambda_1,\lambda_2,+}[i,j]/\boldsymbol{T}^{\lambda_1,\lambda_2,-}[i,j]) > t_h$.
13 $\langle \textit{num\_cases} \rangle^{\mathcal{A}} \leftarrow t_h^+ \cdot \langle \boldsymbol{T}^{\lambda_1,\lambda_2,+}[i,j] \rangle^{\mathcal{A}}$
14 $\langle \textit{num\_controls} \rangle^{\mathcal{A}} \leftarrow t_h^- \cdot \langle \boldsymbol{T}^{\lambda_1,\lambda_2,-}[i,j] \rangle^{\mathcal{A}}$
// Mark this cell as high risk if #cases/#controls $> t_h$.
15 $\langle \boldsymbol{H}^{\lambda_1,\lambda_2}[i,j] \rangle^{\mathcal{B}} \leftarrow$ AGT$(\langle \textit{num_cases} \rangle^{\mathcal{A}}, \langle \textit{num_controls} \rangle^{\mathcal{A}})$
// Swap validation counts if the current cell is high risk.
16 ASWAP$(\langle \boldsymbol{H}^{\lambda_1,\lambda_2}[i,j] \rangle^{\mathcal{B}}, \langle \boldsymbol{V}^{\lambda_1,\lambda_2,+}[i,j] \rangle^{\mathcal{A}}, \langle \boldsymbol{V}^{\lambda_1,\lambda_2,-}[i,j] \rangle^{\mathcal{A}})$
// Compute number of correcly and incorrectly classified samples.
17 $\langle \textit{num\_correct} \rangle^{\mathcal{A}} \leftarrow \sum_{i,j} \langle \boldsymbol{V}^{\lambda_1,\lambda_2,-}[i,j] \rangle^{\mathcal{A}}$ for $i, j \in \{1,2,3\}$
18 $\langle \textit{num\_wrong} \rangle^{\mathcal{A}} \leftarrow \sum_{i,j} \langle \boldsymbol{V}^{\lambda_1,\lambda_2,+}[i,j] \rangle^{\mathcal{A}}$ for $i, j \in \{1,2,3\}$
// Store a bit indicating good/bad accuracy given a public accuracy
// threshold $t_a = t_a^+/t_a^-$.
19 $\boldsymbol{A}^j[\lambda_1,\lambda_2] \leftarrow$ AGT$(t_a^+ \cdot \langle \textit{num_correct} \rangle^{\mathcal{A}}, t_a^- \cdot \langle \textit{num_wrong} \rangle^{\mathcal{A}})$
20 **for** $\lambda_1 \in [a]$ **do** // For each pair of loci λ_1 and λ_2
21 **for** $\lambda_2 \in [a] \setminus \{\lambda_1\}$ **do**
// Output 1 if at least one of the cross validation steps λ_1 and λ_2
// were marked as high risk.
22 $\boldsymbol{O}[\lambda_1,\lambda_2] = \bigvee_{j=1}^{s} \boldsymbol{A}^j[\lambda_1,\lambda_2]$
23 **return** $\boldsymbol{O}$

Function Count($\boldsymbol{R}^{test}, \boldsymbol{R}^{val}, \lambda_1, \lambda_2$):
  // Each party counts the observed genotypes in the clear.
  $\boldsymbol{T}^{\lambda_1,\lambda_2,+}, \boldsymbol{T}^{\lambda_1,\lambda_2,-}, \boldsymbol{V}^{\lambda_1,\lambda_2,+}, \boldsymbol{V}^{\lambda_1,\lambda_2,-} \leftarrow 0^{3\times 3}$
  for $r^k \in \boldsymbol{R}^{test}$ **do** // Count genotype occurrences in the test set
    Extract group α^k, and genotype expressions g^{k,λ_1} and g^{k,λ_2} from record r^k.
    $\boldsymbol{T}^{\lambda_1,\lambda_2,\alpha^k}[g^{k,\lambda_1}, g^{k,\lambda_2}] \mathrel{+}= 1$
  for $r^k \in \boldsymbol{R}^{val}$ **do** // Count genotype occurrences in the validation set
    Extract group α^k, and genotype expressions g^{k,λ_1} and g^{k,λ_2} from record r^k.
    $\boldsymbol{V}^{\lambda_1,\lambda_2,\alpha^k}[g^{k,\lambda_1}, g^{k,\lambda_2}] \mathrel{+}= 1$
  return $(\boldsymbol{T}^{\lambda_1,\lambda_2,+}, \boldsymbol{T}^{\lambda_1,\lambda_2,-}, \boldsymbol{V}^{\lambda_1,\lambda_2,+}, \boldsymbol{V}^{\lambda_1,\lambda_2,-})$

Protocol 3.5: PEA's local preprocessing algorithm to count combination occurrences.

Function $\mathtt{AGT}^{\mathcal{B}}(\langle x_0\rangle^{\mathcal{A}}, \langle x_1\rangle^{\mathcal{A}})$
  $\langle\delta\rangle^{\mathcal{A}} = \langle x_1\rangle^{\mathcal{A}} - \langle x_0\rangle^{\mathcal{A}}$
  $\langle\delta\rangle^{\mathcal{B}} \leftarrow \mathtt{a2b}(\langle\delta\rangle^{\mathcal{A}})$
  // MSB indicates whether $x_0 > x_1$
  return $\langle\delta\rangle^{\mathcal{B}}[\ell]$

Protocol 3.6: Baseline AGT protocol

order of the input values. By utilizing 1-out-of-N OT[61], the calculation of intermediate carry bits in the (Boolean) summation chain can be skipped and the calculation of the MSB can be performed with substantially less communication than the baseline version. The optimized protocol is shown in Protocol 3.7.

[61] Kolesnikov and Kumaresan (2013)

Example for Small Bit Lengths

Let ℓ be small —for example ℓ=4—and denote the arithmetic shares of $\delta = x_1 - x_0 : x_0, x_1 < 2^{\ell-1}$ with $\langle\delta\rangle_0^{\mathcal{A}}, \langle\delta\rangle_1^{\mathcal{A}} \in \mathbb{Z}_{2^\ell}$. The first party—$P_0$—uses the *Oblivious Transfer* (OT) choice bit $\langle\delta\rangle_0^{\mathcal{A}}$. The second party—$P_1$—samples a uniformly random masking bit $r \leftarrow\$ \{0,1\}$ and prepares messages $\{(i + \langle\delta\rangle_1 \mod 2^\ell > 2^{\ell-1} - 1) \oplus r\}_{i=0}^{2^\ell-1}$.

Using the received message, P_0 sets $\langle\mathsf{MSB}\rangle_0^{\mathcal{B}} := (\langle\delta\rangle_0^{\mathcal{A}} + \langle\delta\rangle_1^{\mathcal{A}} \mod 2^\ell > 2^{\ell-1}-1) \oplus r$, and P_1 sets $\langle\mathsf{MSB}\rangle_1^{\mathcal{B}} := r$. This concludes the protocol, as both parties now hold a share of the MSB, which gives the ordering of both input values. The protocol requires $2\kappa + 2^\ell$ bits of communication—that is 272 bits for bit length ℓ=3—and is 7.5× more efficient than the baseline Protocol 3.6. Unfortunately, this only holds for small bit lengths. For ℓ=31 the protocol requires 4.29 GB of communication—orders of magnitude larger than the baseline.

Optimized AGT for Integers of Arbitrary Bit Length

Using an iterative approach by splitting the input integers into chunks, the communication size of the optimized AGT protocol can be reduced significantly. It is sufficient to calculate the carry bits for every intermediate chunk to extract the MSB from the last chunk. The minimal (amortized) communication cost per bit is achieved by instantiating the 1-out-of-N OTs with $N = 2^6$ with $(2\kappa + 2^6)/6 = 53.3$ bits. However, choosing N=2^7 while incurring a higher 54.8

Protocol 3.7: Optimized AGT protocol

Function AGT($\langle x_0\rangle^{\mathcal{A}}, \langle x_1\rangle^{\mathcal{A}}, \ell_s$):

// $\langle x_0\rangle^{\mathcal{A}}, \langle x_1\rangle^{\mathcal{A}}$ are secret-shared in $\mathbb{Z}_{2^\ell}$ but $x_0, x_1 < 2^{\ell-1}$

$\langle \delta\rangle^{\mathcal{A}} = \langle x_1\rangle^{\mathcal{A}} - \langle x_0\rangle^{\mathcal{A}}$

// Counter for the previous chunk

$\ell_{\text{prev}} = 1$

if $\ell > \ell_s$ **then**

  $sel \leftarrow \langle \delta\rangle_0^{\mathcal{A}}[1 : \ell_s]$

  $M \leftarrow (j + \langle \delta\rangle_1^{\mathcal{A}}[1 : \ell_s] > 2^{\ell_s})_{j=1}^{2^{\ell_s}}$

  $r \leftarrow\!\!\$\ \{0,1\}$

  $c \leftarrow \binom{N}{1}$-OT($sel, \{m \oplus r\}_{m\in M}$)

  $\ell_{\text{prev}} \leftarrow \ell_s + 1$

while $\ell_{prev} < \ell - 1$ **do**

  $\ell'_s \leftarrow \min(\ell_s - 1, \ell - \ell_{\text{prev}-1})$

  $\ell_{next} \leftarrow \ell_{\text{prev}} + \ell'_s$

  $sel \leftarrow \langle \delta\rangle_0^{\mathcal{A}}[\ell_{\text{prev}} : \ell_{next}]$

  $sel \leftarrow sel + c \cdot 2^{\ell'_s}$

  $\langle \delta'\rangle_1^{\mathcal{A}} \leftarrow \langle \delta\rangle_1^{\mathcal{A}}[\ell_{\text{prev}} : \ell_{next}]$

  $M_0 \leftarrow \{j + \langle \delta'\rangle_1^{\mathcal{A}} > 2^{\ell'_s}\}_{j=1}^{2^{\ell'_s}}$

  $M_1 \leftarrow \{j + \langle \delta'\rangle_1^{\mathcal{A}} + 1 > 2^{\ell'_s}\}_{j=1}^{2^{\ell'_s}}$

  $M \leftarrow M_r \cup M_{1-r}$

  $r \leftarrow\!\!\$\ \{0,1\}$

  $c \leftarrow \binom{N}{1}$-OT($sel, \{m \oplus r\}_{m\in M}$)

  $\ell_{\text{prev}} \leftarrow \ell_{\text{next}} + 1$

$\langle b\rangle^{\mathcal{B}} = (\langle b\rangle_0^{\mathcal{B}}, \langle b\rangle_1^{\mathcal{B}}) := (c \oplus \langle \delta\rangle_0^{\mathcal{A}}[\ell], r \oplus \langle \delta\rangle_1^{\mathcal{A}}[\ell])$

return $\langle b\rangle^{\mathcal{B}}$

bits of amortized communication per bit leads to less communication rounds and is preferable in our use case.

The optimized AGT protocol for arbitrary bit lengths invokes two subprotocols: First, the OT operation on the first chunk and second, the OTs on the intermediate chunks and carry bits. The result is computed by XOR-ing the last computed carry bit with the calculated MSBs of the shares of difference δ.

The first subprotocol requires $(2\kappa + 2^{\ell_s})$ bits—ℓ_s being the bit length of the split chunk. The second subprotocol requires $\gamma(2\kappa + 2^{\ell_s}) + 2\kappa + 2^{\epsilon}$ bits, where $\gamma = \lceil(\ell - \ell_s)/(\ell_s - 1)\rceil - 1$ is the number of the intermediate chunks and $\epsilon = \ell - \ell_s - 1 \bmod \ell_s - 1$ corresponds to the size of the remainder.

For $\ell \geq \ell_s$ the total communication is equal to $(\gamma+1)(2\kappa+2^{\ell_s})+\lceil\epsilon/(\ell_s-1)\rceil(2\kappa+2^{\epsilon})$ and the number of communication rounds is $\gamma + \lceil\epsilon/(\ell_s - 1)\rceil + 2$, due to sequential calls to the OT functionality. Specifically, for the discussed lengths ℓ_s=7 and N=2^{ℓ_s} OT messages this translates to 384 bits and 2 rounds of communication. ℓ=15 results in 1,028 bits and 4 rounds, ℓ=31 in 1,920 bits and 6 rounds, and, finally, ℓ=63 in 4,100 bits and 12 rounds. Note, that while ℓ denotes the maximum bit-length of the integers to be compared, the inputs are elements of the ring $\mathbb{Z}_{2^{\ell+1}}$.

Comparison to State-Of-The-Art

The only other secure protocol for comparing additively secret shared integers recently introduced by Rathee et al.[62] was shown to be more efficient than the best comparison operation on XOR-shared integers[63]. While their protocol compares two clear text integers to generate a secret shared result, our protocol can be interpreted as an extension of their protocol for comparing (arithmetically) secret shared integers $\langle x\rangle^{\mathcal{A}} > \langle y\rangle^{\mathcal{A}}$ by restricting the allowed inputs to $x, y < 2^{\ell-1} - 1$ and computing the comparison via the difference $\langle x\rangle^{\mathcal{A}} - \langle y\rangle^{\mathcal{A}} < 2^{\ell-1}$. This effectively "sacrifices" one bit for the result. The subtraction can be performed locally, hence our optimized AGT can be seen as a MSB extraction from a secret shared integer—corresponding to Algorithm 2 in Rathee et al. (2020) which, in turn, is based on their Algorithm 1.

That being said, by utilizing 1-out-of-N OT we avoid using AND gates and improve the required communication for ℓ=32-bit inputs by a factor of 1.5 while requiring one more round of communication—we require 1,920 bits and 6 rounds, while Algorithm 2 of Rathee et al. (2020) requires 2,914 bits and 5 rounds.

[62] Rathee et al. 2020, p. 5, Algorithm 1.

[63] Couteau (2018)

3.3.2 Secure Arithmetic Swap

The ability to swap two arithmetic shares based on a (Boolean) choice bit is important for the efficient construction of the arithmetic PMDR protocol. ASWAP implements the following functionality: Let $(\langle x_0\rangle^{\mathcal{A}}, \langle x_1\rangle^{\mathcal{A}})$ be two inputs in arithmetic secret sharing and $\langle b\rangle^{\mathcal{A}}$ a choice bit in Boolean sharing, then ASWAP outputs the pair $(\langle x'_0\rangle^{\mathcal{A}}, \langle x'_1\rangle^{\mathcal{A}}) := (\langle x_b\rangle^{\mathcal{A}}, \langle x_{1-b}\rangle^{\mathcal{A}})$.

While the complexity of both the following naïve implementation and our optimized version is, admittedly, low, PEA uses—to the best of our knowledge—the first efficient implementation, as the "hybrid" multiplication $\langle b\rangle^{\mathcal{B}} \cdot \langle x\rangle^{\mathcal{A}}$ between a Boolean shared bit and an arithmetically shared integer has only been recently constructed[64]. The construction of a batched ASWAP function (Appendix C.4), the security analysis (Appendix C.5), and a correctness proof (Appendix C.6) are given in Appendix C.

[64] Schneider and Tkachenko (2019)

Naïve ASWAP Protocol

The naïve ASWAP function uses four multiplications to calculate:

$$\langle x_i'\rangle^{\mathcal{A}} := (\neg\langle b\rangle^{\mathcal{B}} \cdot \langle x_i\rangle^{\mathcal{A}} + \langle b\rangle^{\mathcal{B}} \cdot \langle x_{1-i}\rangle^{\mathcal{A}}), \quad \forall i \in \{0,1\}.$$

[65] Schneider and Tkachenko (2019)

As described by Schneider and Tkachenko[65], the hybrid multiplication of a Boolean bit with an arithmetic value can be instantiated by two additively correlated OTs. This basic ASWAP protocol requires $8(\kappa+\ell)$ bits of communication in total.

Optimized ASWAP Protocol

[66] Kolesnikov and Schneider (2008)

Inspired by the Boolean "X gate"[66]—a swap function operating on Boolean shares—we design an optimized ASWAP function requiring only one multiplication. This reduces the required communication by a factor of 4 to $2(\kappa+\ell)$ bits. While the X gate is the special case of an ASWAP for bit length $\ell = 1$, it is not easily extendable to values in $\mathbb{Z}_\ell$ with $\ell > 1$. The optimized protocol is shown in Protocol 3.8.

Protocol 3.8: Arithmetic Swap Protocol

Function ASWAP($\langle b\rangle^{\mathcal{B}}, \langle x_0\rangle^{\mathcal{A}}, \langle x_1\rangle^{\mathcal{A}}$):
- $\langle \delta\rangle^{\mathcal{A}} \leftarrow \langle b\rangle^{\mathcal{B}} \cdot (\langle x_1\rangle^{\mathcal{A}} - \langle x_0\rangle^{\mathcal{A}})$
- $\langle x_0'\rangle^{\mathcal{A}} \leftarrow \langle x_0\rangle^{\mathcal{A}} + \langle \delta\rangle^{\mathcal{A}}$
- $\langle x_1'\rangle^{\mathcal{A}} \leftarrow \langle x_1\rangle^{\mathcal{A}} - \langle \delta\rangle^{\mathcal{A}}$
- **return** $(\langle x_0'\rangle^{\mathcal{A}}, \langle x_1'\rangle^{\mathcal{A}})$

Other Application for ASWAP

While developed for PEA's PMDR function, ASWAP is a general building block with versatile applications. One high-impact application might be the implementation of highly efficient arithmetic sorting circuits by combining it with the optimized AGT operation (cf. Section 3.3.1) and constructing sorting networks[67].

[67] Haslop (2020)

3.3.3 Communication of PMDR

To compare the performance of the optimized PMDR protocol using—thanks to the novel AGT and ASWAP protocols—mostly arithmetic secret sharing (denoted as PMDR$^{\mathcal{A}+}$), we translate the MDR algorithm (see Figure 3.2) into a baseline Boolean protocol (denoted as PMDR$^{\mathcal{Y}}$) using Yao's Garbled Circuits and three-halves garbling[68]. In both cases the lengths of integers are fixed to $\ell = 32$—allowing the comparison of 2^{31} genome samples—and $s = 10$ cross-validation steps are performed. The costs of the individual operations are listed in Schneider and Zohner (2013), Demmler, Schneider, and Zohner (2015), and Rosulek and Roy (2021). The final communication cost comparison is shown in Table 3.2.

[68] Rosulek and Roy (2021)

PMDR$^{\mathcal{Y}}$

FOR EACH COMBINATION of L loci and each of 3^L possible combination of alleles, this protocol requires (1) $N-1$ additions for aggregation of allele counts, (2) two multiplications and one comparison for determining the risk category, and (3) one swap operation—implemented as an X gate[69]—to set low and high risk counts in the validation set. The aggregation of validation counts for the accuracy determination requires additional $2 \cdot 3^L - 1$ additions, two multiplications and one comparison. Finally, $s-1$ AND gates combine the results of all cross-validation steps to determine the success of the model in at least one step. In total, for each combination of loci the protocols requires

$$2s(3^L(8\ell^2+s-1)-\ell(4\ell-1))-s+1$$

AND gates. As three-halves garbling requires $1.5\kappa+5$ bit of communication per AND gate, this results in $1{,}394{,}891$ AND gates or 34.34 MB of communication for L=2 loci, and $4{,}347{,}251$ AND gates or 101.05 MB of communication for L=3 loci.

[69] Kolesnikov and Schneider (2008)

PMDR$^{\mathcal{A}+}$

THE PMDR$^{\mathcal{A}+}$ PROTOCOL on the other hand requires one AGT and one ASWAP invocation for each combination of L loci and each of 3^L possible combination of alleles. Afterwards, for each combination of L loci another AGT operation is required. The combination of all cross-validation steps is—as in PMDR$^{\mathcal{Y}}$—is performed in Boolean sharing and requires $s-1$ AND gates. All other operations are performed locally and do not incur any communication cost.

The overall communication cost of PMDR$^{\mathcal{A}+}$ of $s(3^L(\ell(4\kappa+1)+2(\kappa+\ell))+\ell(4\kappa+1))+s-1$ bits—that is 208.8 kB of communication for an interaction depth of L=2 and 585.3 kB for L=3—improve the baseline PMDR$^{\mathcal{Y}}$ by a factor of 164 for $L=2$ and 172 for $L=3$.

Table 3.2: Comparison of the communication costs of the baseline PMDR$^{\mathcal{Y}}$ protocol to the optimized PMDR$^{\mathcal{A}+}$ for interaction depths $L=2$ and $L=3$.

	$L=2$	$L=3$
PMDR$^{\mathcal{Y}}$	34.34 MB	101.05 MB
PMDR$^{\mathcal{A}+}$	208.8 kB	585.3 kB

3.4 Outsourced Data Model

As described in Section 2.3.6, MPC can be performed in an outsourced setting allowing arbitrary many data sources to participate in a computation between M computation parties. This is beneficial for the practical application of PEA in many scenarios. In this section, the computation overhead incurred by employing an outsourced data model with $M=2$ computation parties in PEA is discussed.

3.4.1 Outsourcing of PTuRF & PReliefF

Both PRelief-F and PTuRF share the same outsourcing analysis, as they operate on the same input encoding—for each record each feature is encoded as a 2 bit allele. With a being the number of features and R being the vector of records, $2a|R|$ bits of data must be secret shared for each party. The outsourcing could be implemented in two ways:

1. The outsourcing could be performed completely in Yao's Garbled Circuits, as the data owner sends the keys corresponding to the inputs to the garbler computation party and the corresponding keys to the evaluator computation party. Neither party can infer the clear text input values from the keys. This outsourcing scheme requires 3κ bits of communication for each shared bit, that is $6a|R|\kappa$ bits per data sharing party.

2. Otherwise, the data owners could distribute the inputs as Boolean GMW XOR shared bits between both computation parties. These XOR shares can be converted into Yao's GC without the participation of the data owning party. By transmitting *Pseudorandom Number Generator* (PRNG) seeds to one party the computation can be further reduced, leading to $2a|R|$ bits of communication between a data source and the computation parties and additional $6\kappa a|R|$ bits to re-share the data.

For $a{=}1{,}000$ SNPs and $R{=}10$ records, the first—all Yao's GC—approach requires 1.92 MB of communication between the input owner and the outsourcing servers and the second approach requires 2.5 kB between the input owner and the outsourcing servers and additional 1.92 MB between both servers. The optimal choice depends on the constraints imposed by the network connectivity of the data sources.

3.4.2 Outsourcing of PMDR

PMDR takes 3^L arithmetically shared frequency counts per combination of loci as an input—L being the interaction depth. By expanding a transmitted PRNG seed, an ℓ-bit long input value can be securely transmitted to the computation parties using ℓ bit of communication. By sending a PRNG seed σ_i to computation server $S_{i\neq 0}$, the computation server then can generate the randomness required for $\langle x\rangle_i^A$ and further inputs. Finally, the data source party generates the remaining share $\langle x\rangle_i^A := x - \sum_{i\neq 0}\langle x\rangle_i^A$ and sends it to the other computation party S_0. Note, that the aggregation in the secret sharing process is a non-interactive operation, hence, the communication complexity does not depend on the number of data source parties.

Given s cross-validation steps and a attributes per record, the required communication per data source is $s\cdot 3^L\cdot\ell\cdot\binom{a}{L}$ bits using this scheme. For $a{=}1{,}000$ SNPs, this results in 180 MB of communication for an interaction depth of $L{=}2$ and 60 GB for an interaction depth of $L{=}3$—significantly less than the required communication in a "true" n-party computation (cf. Section 2.3.6).

The required communication for transmitting the outputs tally up to—in comparison nearly negligible—$\binom{a}{L}$ bits per party. For $L{=}2$ this translates to 0.5 MB and for $L{=}3$ to 166 MB (cf. Table 3.5).

The overall very small communication overhead incurred by the outsourced model demonstrates its viability for practical data pooling and analyses over distributed genomic datasets.

3.5 Implementation

Practical Private Epistasis Analysis using MPC (PEA)—our protocol for *Private Epistasis Analysis* (PEA)—is implemented in C++ using the MOTION MPC framework[70]. As the protocol required the implementation of various new building blocks and low-level schemes—for example three-halves garbling[71], $\binom{N}{1}$-OT[72], and Boolean/arithmetic hybrid multiplication[73]—only MOTION provided both the required flexibility to incorporate structural changes to the frameworks architecture and the performance of a low-level, compiled programming language.

[70] Braun et al. (2021)
[71] Rosulek and Roy (2021)
[72] Kolesnikov and Kumaresan (2013)
[73] Schneider and Tkachenko (2019)

3.5.1 Three Halves Make a Whole Garbling

PEA's complete feature selection, as well as some (small) parts of the PMDR algorithm are constructed using Yao's garbled circuits. To optimize the runtime performance of these building blocks, we implemented the current state-of-the-art garbling scheme—*Three-Halves Garbling* (3HG)[74]. This is to our knowledge the first implementation of this garbling scheme. *Three-Halves Garbling* (3HG) sacrifices some computational efficiency to reduce the required communication.

[74] Rosulek and Roy (2021)

When benchmarking our implementation, our 3HG engine achieves AND gate garbling and evaluation rates of $11.2\,\mathrm{M\,s^{-1}}$ and $27.5\,\mathrm{M\,s^{-1}}$. The MOTION implementation of the previous state-of-the-art *half-gate* garbling[75] (cf. Section 2.3) achieves higher rates for both operations—3HG garbles $4.7\times$ slower and evaluates $2.5\times$ slower. This is slightly worse than predicted by Rosulek and Roy [76].

[75] Zahur, Rosulek, and Evans (2015)
[76] Rosulek and Roy (2021)

As a microbenchmark, we evaluated 512 AES circuits in parallel using 5 threads to fully saturate a $10\,\mathrm{Gbit/s}$ network connection given the garbling and evaluation rates. While the benchmark results should be taken with a grain of salt—the implementation of 3HG introduced substantial changes to MOTION's inner workings that might not be accounted for here—3HG achieves a $2.2\times$ speedup compared to half-gate garbling ($0.22\,\mathrm{s}$ vs. $0.5\,\mathrm{s}$).

Detailed profiling of 3HG's performance identifies both its $1.5\times$ higher number of AES invocations and the *substantially* higher degree of branching as the principal bottlenecks. By employing advanced vectorization instructions, such as `_mm256_permutevar8x32_epi32`[77] using AVX2 or `_mm512_permutexvar_epi64`[78] using AVX512, the performance impact of the branching might be mitigated. Being outside the scope of PEA, we leave this as future work.

[77] https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html#text=_mm256_permutevar8x32_epi32

[78] https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html#text=_mm512_permutexvar_epi64

3.6 Performance Evaluation

3.6.1 Test Setup

The performance evaluations of PEA were performed on two servers in a lab environment with Intel Core i9-7960X processors and $128\,\mathrm{GB}$ of RAM each. The

servers were connected via a local 10 Gbit/s network connection with a median latency of 0.2 ms. All reported measurements are averaged over 10 runs. As the protocols in PEA are either highly parallelized or have a constant number of interaction rounds, we expect the network bandwidth to be the most critical parameter.

The experiments use synthetic input data for two reasons: First, as MPC and the circuit generation is independent of the specific inputs, all performance metrics only depend on the structure of the data—for example bit lengths—and not on the data itself. Second, the implemented algorithms perform exact computations, hence, using real, difficult to obtain, and privacy-sensitive datasets brings no discernible advantage or additional insights. The usage of synthetic data is the least privacy intrusive way and follows the principle of "privacy by design".

The PEA benchmarks are performed in two network settings: First, a *Wide Area Network* (WAN) setting and second, a *Local Area Network* (LAN) setting. The descriptions and rationale behind the settings are described in Appendix B, the chosen parameters for bandwidth and latency are given in Table 3.3.

Table 3.3: Network parameters for the experimental evaluation of PEA

Setting	Bandwidth	Latency
WAN	10 Gbit/s	50 ms
LAN	10 Gbit/s	0.2 ms

The benchmarks do not include a "true" n-party setting—that is, with n computation parties—as most practical MPC protocols scale quadratically in the number of parties[79], which—added to the high asymptotic complexity of PEA—quickly renders analyses of real-world workloads infeasible.

[79] Exceptions, like the linearly scaling variation in the SPDZ protocol (Damgard et al. (2012)) are so computationally intensive, that the linear scaling becomes practically beneficial if several thousands of parties participate in the computation.

3.6.2 Performance of PReliefF and PTuRF

The setup and online phase runtimes for both private feature selection algorithms—*Private Relief-F* (PRelief-F) and *Private Tuned Relief-F* (PTuRF)—are shown for both network settings in Figures 3.3 and 3.4. One plot shows the runtimes for varying numbers of records and the other for varying numbers of features. Table 3.4 shows the measurements in tabular form.

As PEA's implementation is optimized for runtime and communication performance, it is not surprising that RAM utilization was a bottleneck in the experiments. Because of RAM exhaustion, PTuRF was not benchmarked across the complete parameters space. The dashed lines show the extrapolated runtimes based on a power-function fit using $f(x) = ax^b + c$ as a functional model. The fit parameters and their uncertainties are listed in Appendix C.7.

After a steep, initial increase in runtime, the expected linear asymptotic dependencies on both the number records and the number of features is visible for PRelief-F and PTuRF. By employing Yao's GC—a constant-round protocol—the impact of high network latencies is mitigated and shows only a moderate impact on the feature selection runtime. While small in comparison to the total runtime, the effect of the network latency is best observed by comparing the setup times between both settings.

The development of Tuned Relief-F[80] was driven by first, improving the robustness in the presence of noisy features and second, to speed up the calculation.

[80] Moore and White (2007)

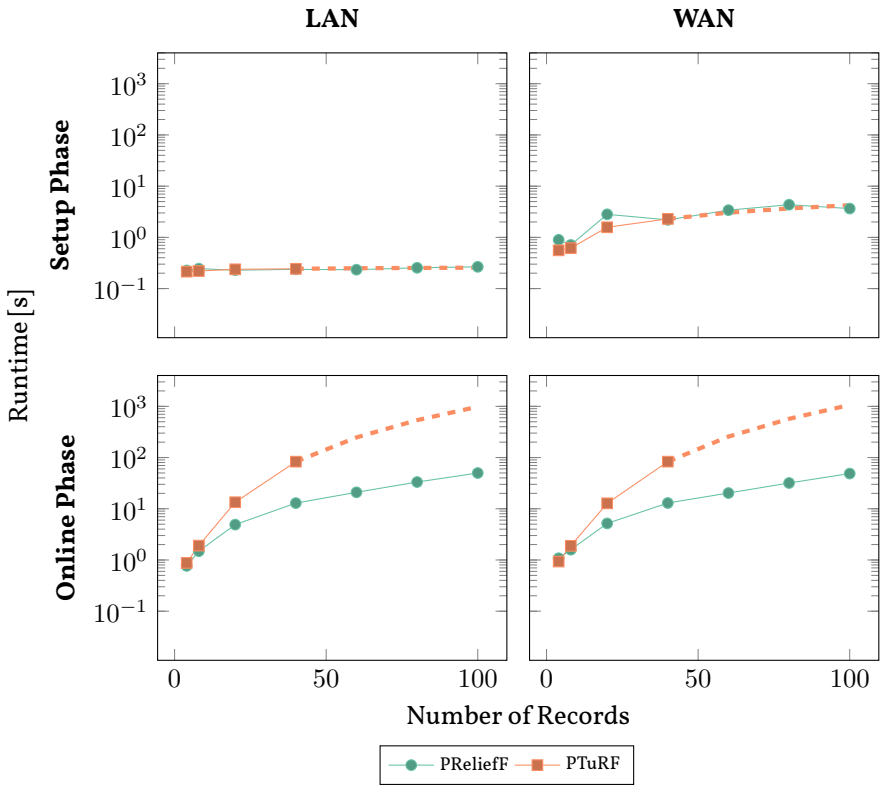


Figure 3.3: Setup and online runtime in seconds of both private feature selection algorithms with varying numbers of records and both network settings.

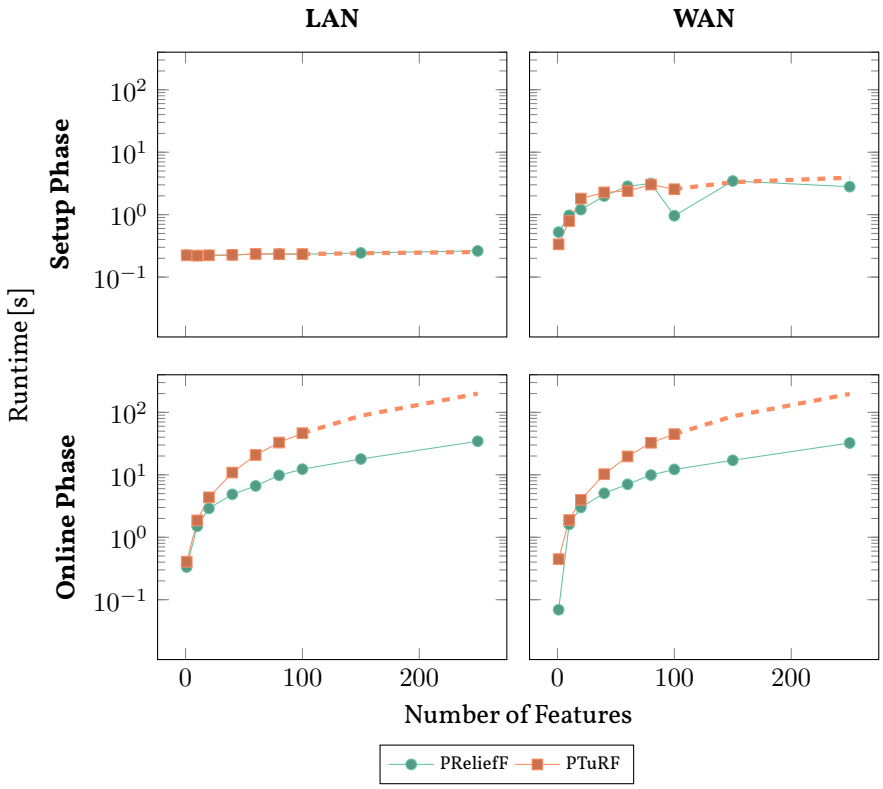


Figure 3.4: Setup and online runtime in seconds of both private feature selection algorithms with varying numbers of features and both network settings.

In the privacy-preserving setting, PTuRF fails to achieve the second goal (while contributing the same advantage to the first as the clear text protocol). Although the number of features is reduced in each iteration, the additional sorting and filtering negates all possible performance gains.

Extrapolated from the parameter dependencies, datasets of sizes encountered in real-world applications—for example $\sim$ 100 records with $\sim$ 10,000 features[81]—can be processed using PRelief-F in under a day.

[81] Chen, X. Zhang, and R. Zhang (2019)

			$\lvert R\rvert$=4	$\lvert R\rvert$=8	$\lvert R\rvert$=20.00	$\lvert R\rvert$=40.00	$\lvert R\rvert$=60.00	$\lvert R\rvert$=80.00	$\lvert R\rvert$=100.00
PRelieff$^{\mathcal{Y}}$	LAN	Setup	0.23 s	0.25 s	0.23 s	0.24 s	0.23 s	0.26 s	0.27 s
		Online	0.77 s	1.49 s	4.90 s	12.91 s	20.96 s	33.26 s	49.87 s
	WAN	Setup	0.90 s	0.71 s	2.82 s	2.19 s	3.40 s	4.34 s	3.65 s
		Online	1.08 s	1.59 s	5.17 s	12.95 s	20.28 s	31.87 s	48.64 s
		Comm.	3.75 MB	9.63 MB	40.98 MB	138.93 MB	294.13 MB	506.45 MB	775.93 MB
		Comm. outs.	3.84 kB	7.68 kB	19.20 kB	38.40 kB	57.60 kB	76.80 kB	96.00 kB
PTuRF$^{\mathcal{Y}}$	LAN	Setup	0.21 s	0.22 s	0.24 s	0.24 s	—	—	—
		Online	0.88 s	1.91 s	13.41 s	83.13 s	—	—	—
	WAN	Online	0.56 s	0.62 s	1.57 s	2.30 s	—	—	—
		Online	0.93 s	1.89 s	12.77 s	83.24 s	—	—	—
		Comm.	4.11 MB	11.13 MB	107.15 MB	510.77 MB	—	—	—
		Comm. outs.	3.84 kB	7.68 kB	19.20 kB	38.40 kB	—	—	—

Table 3.4: Runtimes and communication (incl. outsourcing) for PEA's private Relief-F (PReliefF$^{\mathcal{Y}}$) and private Tuned Relif-F (PTuRF$^{\mathcal{Y}}$) protocols (see Protocol 3.3 and Protocol C.1) filtering $\lvert R\rvert$ records containing 10 SPNs each. Dashes "—" indicate not benchmarked parameter values.

3.6.3 Performance of PMDR

Figure 3.5 and table 3.5 report the performance measurements of PEA's *Private Multifactor Dimensionality Reduction* (PMDR) algorithm for varying numbers of features a, both network models and two interaction depths $L = 2$ and $L = 3$. The exponential influence of the number of interacting loci L is visible both in the runtime and communication size. The latter reaches for L=3, $\lvert R\rvert$=1,000 nearly 100 TB.

For smaller numbers of SNPs or lower interaction depths—for example a=1,000, L=2 or a=100, L=3—PMDR completes in under an hour in both network settings. This demonstrates the importance of the preceding feature selection stage.

PMDR	#comb.	a=10	a=100	a=1,000.00
LAN	L=2	1.71 s	24.85 s	43.08 min
	L=3	3.29 s	33.68 min	—
WAN	L=2	10.88 s	42.71 s	1.04 h
	L=3	10.01 s	47.71 min	—
Comm.	L=2	9.39 MB	1.03 GB	104.29 GB
	L=3	70.23 MB	94.64 GB	97.25 TB
Comm. outs.	L=2	16.20 kB	1.78 MB	180.50 MB
	L=3	129.60 kB	174.83 MB	60.16 GB

Table 3.5: Runtimes and communication (incl. outsourcing) for PEA's Private Multifactor Dimensionality Reduction (PMDR$^{\mathcal{A}+}$) protocol (see Protocol 3.4) with s=10 cross-validation steps using ℓ=32-bit integers, combining L loci on databases containing a attributes—that is SNP's. We report total runtimes only, since PMDR is not optimized for improved online phase times. Experiments that ran longer than a day are excluded and marked with dashes "—".

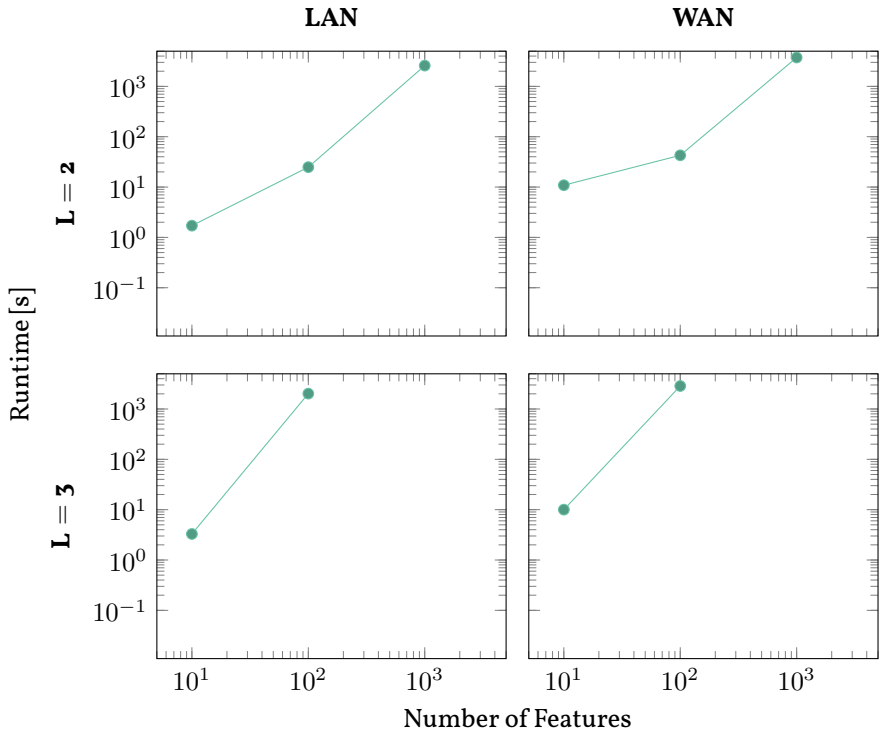


Figure 3.5: Runtimes in seconds for PMDR, for $L = 2$ and $L = 3$. Both interaction depths are benchmarked in two network environments.

3.6.4 Total Performance

As seen before in Section 3.6.3, the exponential complexity of PMDR strongly suggest the combination with a feature selection algorithm. This not only substantially improves the efficiency of the subsequent epistasis analysis, but improves the results by fostering robustness against noisy attributes and class noise[82]—at the cost of revealing the number of filtered features. As discussed by Moore and White (2007), the optimal choice of filtered features is dependent on the amount of noise, the size of the dataset, and even the heritability of the target phenotype. Because of this it is hard to give exact estimates, but TuRF showed over 80 % accuracy while filtering out 95 % of the SNPs[83]—reducing a set of 1,000 SNPs to 50 SNPs.

[82] See Section 3.1.4.

[83] Moore and White (2007)

However, the empirical data in the previous section show that the runtime cost of PRelief-F and PTuRF lead only to performance gains in a composed system for large numbers of features—the slope of PMDR runtimes rise with increasing numbers of features—or interaction depths $L > 2$. In these cases even a moderate reduction of the numbers of features lead to a notable speedup—for a=10,000 features a reduction by 10 % leads to a 20 % improvement, reducing the analyzed combinations from 50 to 40 million.

3.7 Outcome and Prospects

This chapter details the context, design and implementation of PEA, the first secure and privacy-preserving epistasis analysis protocol. To mitigate the super-exponential complexity of unknown-structure, full observability PGM structure reconstruction, PEA implements two MPC protocols for feature selection—*Private Relief-F* (PRelief-F) and *Private Tuned Relief-F* (PTuRF)—and one protocol for

Private Multifactor Dimensionality Reduction (PMDR). These protocols, heuristically calculating the most influential nodes in the Bayesian networks, are important elements for privacy-preserving Epistasis Analyses (EAs), thus enabling scientists to perform inter-institutional medical research otherwise often not possible, due to data protection regulation. Our highly efficient outsourcing algorithm allows the joint computation between arbitrary numbers of parties, while only occurring a (comparatively) minor communication overhead.

To achieve practical runtimes, we designed two new, efficient building blocks for generic arithmetic MPC. PEA's $\binom{N}{1}$-OT[84]-based custom protocol for *Arithmetic Greater Than* (AGT) uses $1.5\times$ less communication at slightly more interaction rounds than the state-of-the-art[85], and Arithmetic Swap (ASWAP), a generalization of the Boolean swap gates by Kolesnikov and Schneider [86] for the arithmetic case, uses $4\times$ less communication than the naïve design.

[84] Kolesnikov and Kumaresan (2013)

[85] Rathee et al. (2020)

[86] Kolesnikov and Schneider (2008)

Furthermore, we present and analyze the first implementation of the novel three-halves garbling[87], which shows a greater slowdown than expected due to a higher degree of branching compared to the prior best garbling scheme. However, the network bandwidth still remains the bottleneck.

[87] Rosulek and Roy (2021)

Lastly our experimental performance evaluations show that our solution for secure and privacy-preserving Epistasis Analyses achieve practical runtimes of under a day for real-world dataset sizes.

Chapter 4
Private Solution to the Kidney Exchange Problem

End-stage renal diseases pose a significant burden on the public healthcare system[1] affecting around 7 % of U.S. adult citizens[2]. With increasing risks due to age, kidney replacement therapy is an ever-increasing aspect of medical care. Apart from chronic dialysis, the cleansing of blood in external machines simulating the functions of the kidneys, kidney transplants are the only viable long term options for those patients.

[1] Thurlow et al. (2021)
[2] Murphy et al. (2016)

Unfortunately the demand significantly exceed the supply of transplants[3], thus finding a willing, living donor is the only timely way to receive a transplant. Even after finding a donor, the projected success of a procedure, i.e., the survival of the recipient, depends on the medical compatibility of donor and recipient, that is immunological and morphological compatibility. To help as many patients as possible, voluntary donors and "their" recipients are often registered in *kidney exchanges*. The biomedical compatibility of all donors and recipients is calculated according to medical evidence-based guidelines and a bipartite *compatibility graph* is constructed, linking donors to possible recipients. "Fairness" in this context means, that if a donor is matched with a recipient, the donor's "original" but incompatible recipient will receive a compatible transplant as well. This corresponds to finding *exchange cycles* in the graph[4]. Regulatory and practical circumstances apply additional constraints, e.g., on the cycle length as all procedures should be performed (almost) simultaneously. The final result is the set of cycles maximizing both graph coverage and the accumulated edge weights, as those weights represent medical compatibility and with that long-term success probability. An overview of the protocol with its "building blocks" is shown in Figure 4.1.

[3] Eurotransplant (2021)
[4] Biró, Klundert, et al. (2021)

As we are dealing with highly sensitive medical health data, this computation should be performed in a decentralized, privacy-preserving manner. The reasoning for that is two-fold: First to limiting the privacy damage inflicted by accidental data leakage or breaches in one institution and second to lessen the legal burdens and supporting compliance of the participating institutions by providing strong privacy guarantees allowing smaller medical facilities to participate in the kidney exchange.

Of course, medical questions are often outside of full algorithmic evaluation. It is important to note, that medical professionals *must* evaluate the results of any algorithmically determined treatment plan and the algorithm should allow the flexible adaptation to situational necessities and updated medical guidelines.

This chapter describes the design and implementation of *Secure and Private Investigation of the Kidney Exchange problem* (SPIKE), a distributed privacy-preserving pro-

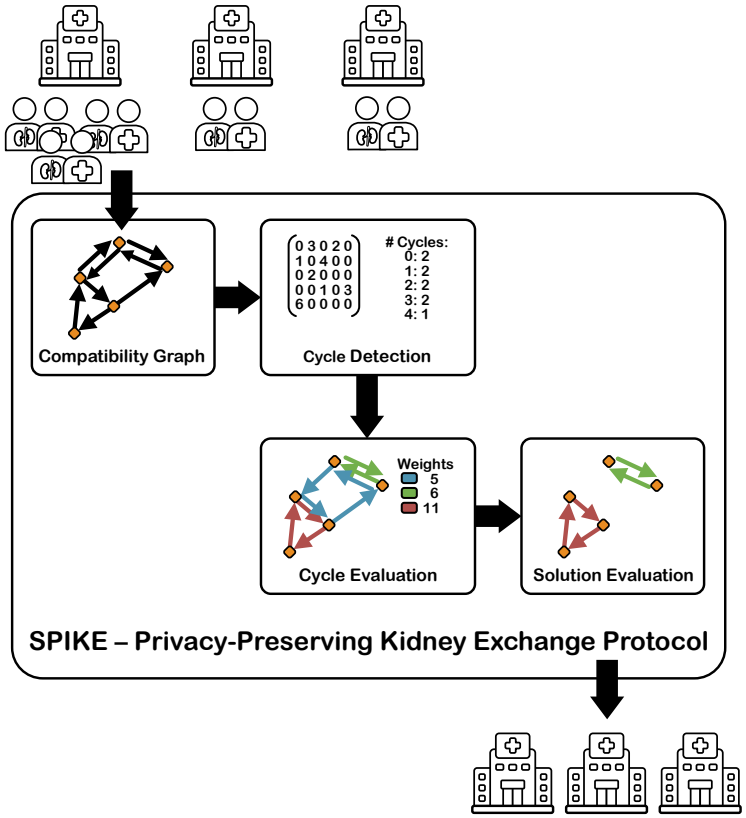


Figure 4.1: Overview of the Privacy-Preserving Kidney Exchange Protocol SPIKE and its algorithmic parts. It calculates the best set of exchange cycles—that is accommodating the most patients while achieving the highest combined success probability—while keeping the patients' data strictly private. This figure was created by the author and used in Birka et al. (2022), licensed under CC-BY.

tocol for solving the *Kidney Exchange Problem* (KEP) assuming the semi-honest security model. It draws upon the work submitted for publication in Birka, T., Hamacher, K., Kussel, T., Möllering, H., Schneider, T. (2022) "SPIKE: Secure and Private Investigation of the Kidney Exchange Problem". *Submitted to BMC Bioinformatics and Decision Making*. The author was deeply involved in nearly every aspect of the described work, notable exception being the implementation done by T. Birka, but otherwise contributing significantly to the design, experimental setup, and manuscript of the publication, as well as supervising T. Birka jointly with H. Möllering. SPIKE improves the medical quality of the results by considering more compatibility factors than the current state-of-the-art[5], namely, age, sex, human leukocyte antigens, and weight. By considering these factors, the risk of failing procedures is reduced increasing the robustness of the solution.

[5] Breuer, Meyer, Wetzel, and Mühlfeld (2020); Breuer, Meyer, and Wetzel (2022)

To demonstrate practical runtimes, the open source implementation[6] of SPIKE is comprehensively empirically evaluated benchmarking runtime and communication costs. Due to the carefully optimized hybrid *Secure Multi-Party Computation* (MPC) protocols using the ABY[7] (cf. Section 2.3.7) framework, we achieve about $30{,}000\times$ speedup over Breuer, Meyer, Wetzel, and Mühlfeld (2020) and $400\times$ over Breuer, Meyer, and Wetzel (2022).

[6] Available under the GNU LGPL v3 license here: https://encrypto.de/code/PPKE

[7] Demmler, Schneider, and Zohner (2015)

4.0.1 Related Works and current State of the Art

The kidney exchange system was introduced in 1991 in South Korea. In Europe, the first kidney exchange program was developed in 1999 in Switzerland[8]. Over

[8] Ellison (2014)

the years many different research groups explored various aspects of the KEP. Most solutions to the KEP are based on *Integer Linear Programming* (ILP), an optimization techniques often applied to graph problems. While there are some works to provide privacy-preserving ILP solver and MPC graph algorithms, the scale of the KEP is—currently—prohibitive for those techniques. Nevertheless, all those works explore the problem space and thus relate to SPIKE. Furthermore, two works represent the current state-of-the-art with regard to privacy-preserving kidney exchange protocols— Breuer, Meyer, Wetzel, and Mühlfeld (2020) and Breuer, Meyer, and Wetzel (2022), both directly interested in the privacy-preserving solution of the KEP.

Privacy-Preserving (I)LP Solver

Most work attempting to solve ILP and related problems in the privacy-preserving domain rely on substituting the discrete nature of *integer* or *mixed* linear programming with the *linear relaxation*, allowing continuous variables and solutions. While the relaxed solution might not be a solution to the integer problem, it gives bounds for an exact solution while transforming the $\mathcal{NP}$-hard ILP problem into a polynomial bound computation. Two early works published MPC-based LP solvers using the *Simplex algorithm*—in a two-party outsourced computation scenario[9] and in a "true" n-party computation[10] (the latter even providing information theoretic security). While being prohibitively slow for real-world dataset and constraint set sizes, other researches improved upon these solutions, e.g., by utilizing a fixed-point transformation to trade accuracy against runtime performance[11]. Lastly, the research linked to the European *SecureSCM* program, researching algorithms for secure, distributed supply chain management, developed efficient solutions based on problem transformation and the design of a domain specific programming language[12]. However, none of these works provide the ILP solution capabilities needed for solving the KEP.

[9] J. Li and Atallah (2006)

[10] Toft (2009)

[11] Catrina and Hoogh (2010)

[12] Schroepfer, Kerschbaum, and Mueller (2010); Kerschbaum et al. (2011); Dreier and Kerschbaum (2011)

Privacy-Preserving Graph Frameworks

The publication of "Graphsc"[13] started the research in MPC-based graph algorithms. The body of work is mostly focused on *graph parallel algorithms*, that is distributed computation on graph structures where vertices perform computations and diffusing the results, potentially for usage in multiple computation rounds. Examples for algorithms implementable in a graph parallel fashion are matrix factorization, histogram calculations, PageRank calculation, and (parallel) *Breadth-First Search* (BFS). The main computation paradigm is the *message passing* with three distinct phases: First, in the *scatter* phase, each vertex distributes—scatters—its own data. Second, in the *gather* phase, each vertex gathers the data scattered by incident vertices and, third, *applies* it to its internal computation state. Hence, this paradigm is sometimes called *SGA* and is used in (parallel) big data architectures such as *MapReduce*. One benefit of oblivious graph structure over *Oblivious Random Access Memory* (ORAM) schemes is the vastly improved performance.

[13] Nayak et al. (2015)

Since Graphsc[14] many improvements have been achieved, for example performance optimizations by allowing differentially private leakage of vertex degree information[15], introducing arithmetic circuits with four-party computation[16], or recently, by replacing secure sort with secure shuffle operations[17].

[14] Nayak et al. (2015)

[15] Mazloom and Gordon (2018)

[16] Mazloom, P. H. Le, et al. (2020)

[17] Araki et al. (2021)

Robustness of Exchange Cycles

The potential cancellation of transplantations after the determination of (approximately) optimal exchange cycles is a major concern in kidney exchange programs. Many reasons can lead to the cancellation of a procedure: from medical professionals overruling the algorithmically determined compatibility, logistical issues, or a donor's consent withdrawal, e.g., because "his" original incompatible recipient already received a deceased donor organ[18].

[18] Pansart et al. (2014)

To mitigate these disruptions, a solution to the KEP must be as *robust* as possible, that means that either the probability of a failing donor-recipient connection is reduced or the algorithm shows resilience towards these disruptions and can recover.

[19] Carvalho et al. (2021)

Carvalho et al.[19] pursue the second solution and propose three policies to allow the mitigation of or recovery from dropouts within an exchange cycle. The first policy penalizes uncertainty in the success of a procedure, thus taking the costs—including opportunity costs—of failing edges into account. The other two policies propose recovery strategies to reduce the impact of failing edges. All presented algorithms are computationally expensive rendering them unsuitable for the MPC adaptation.

[20] Ashby et al. (2017)

Ashby et al.[20] examine the importance of many demographic and (bio) medical factors, such as age, sex, obesity, weight, height, number of HLA mismatches and blood type based on over $230{,}000$ kidney-only transplants. The results of this systematic exploration are published in form of a kidney graft survival probability calculator.

SPIKE increases the robustness by including the most important factor identified by Ashby et al. (2017), thus decreasing the risk of misjudgments in the algorithmic compatibility determination. Furthermore, we follow the recommendation of Pansart et al. (2014) to use cycle length of only two or three to reduce the impact of failing edges. While this reduction in cycle length benefits the logistical challenges of the kidney exchange—according to best practices[21] all procedures of one exchange cycle should be performed simultaneously—the "vertex coverage" of the found solutions decrease[22].

[21] Abraham, Blum, and Sandholm (2007)

[22] Ellison (2014)

Privacy-Preserving Kidney Exchange Protocol

Two publications by Breuer, Meyer, Wetzel, and Mühlfeld [23] describe decentralized privacy-preserving protocols for the KEP in a semi-honest security setting.

[23] Breuer, Meyer, Wetzel, and Mühlfeld (2020); Breuer, Meyer, and Wetzel (2022)

Privacy-preserving KEP with HE. The first *Privacy-Preserving Kidney Exchange Protocol* (PPKEP)[24] is based on *Homomorphic Encryption* (HE)[25], more specifically a threshold variant of the Paillier cryptosystem[26]. Each donor–recipient pair is instantiated as a computing party, effectively creating a HE-based MPC protocol.

[24] Breuer, Meyer, Wetzel, and Mühlfeld (2020)

[25] See Section 2.3.9

[26] Fouque, Poupard, and Stern (2000)

The protocol uses an extensive pre-computation phase to generate an ensemble of all possible exchange graph configurations. Although user configurable, the work examines cycles of length $L = 2$ and $L = 3$. The compatibility graph adjacency matrix is calculated in the joint computation using *Human Leucocyte Antigens* (HLA)-crossmatch and ABO blood type compatibility as edge connection criteria. This resulting adjacency matrix is compared to the pre-computed ensemble and the graph with maximum size is delivered as the protocol's output. Unfortunately, the runtimes of the protocol prohibit the real-world application. Due to the protocol's exponential runtime complexity with regard to the number of pairs, the calculation takes 14 s for two pairs and 13 h for nine pairs—the maximum number of pairs benchmarked in the publication.

Privacy-preserving KEP with Shamir's Secret Sharing. Concurrently with the development of SPIKE, Breuer, Meyer, and Wetzel [27] introduced a PPKEP for *crossover kidney exchanges*—a variation searching for the *one* best exchange pair for each pair, effectively evaluating cycles of length $L = 2$. For that it uses the graph matching algorithm by Pape and Conradt [28]. This limitation enables a more efficient protocol design, leading to polynomial runtime complexity. The system is complimented by protocols for an online mode, allowing the addition and removal of donor–recipient pairs without complete recalculation of the exchange graph. The PPKEP is implemented using the MP-SPDZ MPC framework[29] and its implementation of semi-honest *Shamir's Secret Sharing* (SSS). It reduces the runtime for 15 pairs and cycle length $L = 2$ from 8.5 h in Breuer, Meyer, Wetzel, and Mühlfeld (2020) to 30 min.

[27] Breuer, Meyer, and Wetzel (2022)

[28] Pape and Conradt (1980)

[29] Keller (2020)

Our privacy-preserving KEP protocol SPIKE allows freely configurable cycle length, results in medically better (and robuster) solutions by including four additional biological factors, and outperforms both state-of-the-art protocols in comparable parameter and network settings by several orders of magnitude. The increased performance is achieved by efficiently combining three MPC techniques and carefully designing optimized circuits (described in Section 4.2).

4.1 Medical Considerations for Kidney Transplantations

The first successful surgical replacement of a dysfunctional kidney was achieved 1954[30]. One major factor for the positive outcome of this procedure was, that the patient had an identical twin brother voluntarily donating one functioning organ[31]. Since then, new procedure methods and drug treatments allow the transplant between unrelated donors and recipients, however, many factors influence the medical compatibility and survival rate post transplant.

[30] Hatzinger et al. (2016)

[31] Leeson and S. P. Desai (2015)

Patients in need of a kidney replacement treatment suffering from certain diseases—HIV, hepatitis B and C, cytomegalie, and the Eppstein-Barr virus, for example—can only receive organs from a donor with the same condition. Patients requiring both kidney and liver transplants are most of the time in a special waiting list, as it is beneficial to replace both organs simultaneously. Furthermore, anatomical considerations come into play, like the exact position of vein connections. Most of these aspects must be evaluated by medical specialists on

a case-by-case basis, however some criteria may be evaluated algorithmically to generate a set of potential candidates to be evaluated by experts.

Human Immune System

MOST CRITERIA SPIKE evaluates to determine the medical compatibility between donors and recipients are related to a possible immunological rejection of the transplant. Every human is in possession of two immune systems, an *innate*, static system which is fully developed at birth, and an *adaptive*, dynamic system which evolves during the lifetime due to the exposure to *pathogens*[32]—harmful substances, organisms and viruses. Special types of white blood cells—*lymphocytes*—are primarily responsible for the adaptive immune system's function. Lymphocytes detect exogenous substances, potential pathogens, by special, identifying molecular structures on the surface of those substances—*antigens*. By producing *antibodies*, antigen-specific molecules that can attach to those structures, they prevent the pathogens from docking, thus inhibiting their harmful effect[33]. However, antigens do not only exist on pathogens but are naturally produced endogenous molecules as well.

[32] And, in pathological cases, due to the exposure to endogenous substances—leading to autoimmune disorders.

[33] Alberts et al. (2002)

4.1.1 Prohibitive Immunological Compatibility

SPIKE follows evidence-based guidelines to assess the immunological compatibility between donors and recipients. While most immunological criteria gradually influence the probability of a successful transplant, SPIKE uses one assessment as a compatibility prohibitive factor: a positive HLA crossmatch.

Human Leukocyte Antigens (HLA) crossmatch

THE IMMUNOLOGICAL "FINGERPRINT" recognized as normal, the "default" antigens, is genetically determined. One important group of antigens in that regard, the *Human Leucocyte Antigens* (HLA), are genetically encoded in the *Major Histocompatibiliy Complex* on the sixth chromosome (with one exception encoded on chromosome 15). The HLA are grouped into classes and further split into groups according to their loci[34] and site of expression[35]. Only classes I and II encode HLA and are of interest for transplantation immunology.

[34] Sung (2007)

[35] Nguyen (2021)

SPIKE assesses the general compatibility between recipients and donors by performing an *HLA crossmatch*—the matching of the donor's human leukocyte *antigens* to the recipient's corresponding human leukocyte *antibodies*[36]. A positive HLA crossmatch means, that the transplantation would cause a severe immune reaction including possible allograft rejection and death[37]. While an accompanying treatment with modern immunosuppressant drugs might provide a chance of successfully performing such a procedure[38], such specialized cases are not in scope of an automated algorithmic evaluation.

[36] Eurotransplant (2018)

[37] Lefaucheur et al. (2010); Ntokou et al. (2011)

[38] Santos et al. (2014)

As recommended by Eurotansplant's guidelines[39], SPIKE considers the HLA groups most frequently screened for kidney replacement therapy[40]: HLA-A, -B, and -DR. Additionally, the HLA-DQ antigens are considered, as they are linked to post-operative antibody-mediated rejection events[41]. The list of HLA considered in the default configuration is shown in Table 4.1, although this list is user configurable.

[39] Eurotransplant (2018)

[40] Eurotransplant (2018)

[41] Leeaphorn et al. (2018)

Class I			Class II	
HLA-A	HLA-B		HLA-DR	HLA-DQ
A23	B38	B60	DR11	DQ5
A24	B39	B61	DR12	DQ6
A25	B44	B62	DR13	DQ7
A26	B45	B63	DR14	DQ8
A29	B49	B64	DR15	DQ9
A31	B50	B65	DR16	
A32	B51	B71	DR17	
A33	B52	B72	DR18	
A34	B54	B75		
A66	B55	B76		
A68	B56	B77		
A69	B57			
A74	B58			

Table 4.1: HLA assessed in SPIKE's donor–recipient compatibility testing.

4.1.2 Match quality estimation

In addition to the prohibitive compatibility estimation using HLA crossmatching, a multitude of other factors influence the probability of allograft rejection. To generate a medically more meaningful, gradual compatibility estimation than using HLA crossmatching alone, we process additional factors, based on the empirical findings of Ashby et al.[42]:

[42] Ashby et al. (2017)

(i) *HLA match.* Not only the human leukocyte antigen/antibody combination influences the probability of antibody-mediated organ rejection, but the similarity of the donor–recipient "HLA footprint" as well. If the donor expresses a subset of the recipient's HLA, the probability of a successful organ transplant is increased. The higher the number of "mismatches", the higher the risk that the recipient will develop antibodies against the donors HLA in the future[43]. HLA mismatches are not an exclusion criterion. Treated with immunosuppressants the rejection probability can be lowered. However, the usage of immunosuppressants itself is linked to possible adverse health effects[44]. Mismatches involving the HLA-DQ group are particularly linked to antibody-mediated rejections[45].

[43] Opelz and Döhler (2012)

[44] Opelz (1997); Lim et al. (2012)

[45] Leeaphorn et al. (2018)

As each HLA group is encoded in one locus, each person can only inherit up to two HLA per group[46], meaning that per group at most two mismatches can occur. Having no mismatches is a very rare case, usually only occurring in twin donor–recipient pairs. We group the number of mismatches into four groups: 0 mismatches, 1 to 2 mismatches, 3 to 4 mismatches, and $5+$ mismatches[47]. The last group shows a more than 6 % cumulative risk for death with a functioning graft during the first year.

[46] Nguyen (2021)

[47] Opelz and Döhler (2012)

(ii) *ABO blood type.* The blood type in the ABO system is dependent on the presence or absence of two different antigens on the surface of the red blood cells leading to four combinations. The relative frequency of the blood types varies across different populations. The absence of both type A and type B antigens mark blood type O (a universal donor), the presence of both mark blood

type AB (a universal recipient), and the presence of only one mark blood type A and B, respectively. The immune reaction caused by the contact of incompatible blood results in blood clumping and possibly failing transplants—the compatibility between types is shown in Table 4.2. While 43 systems describing blood types are listed[48] by the International Society for Blood Transfusion[49]—the most important ones being ABO, Rh, and Kell system—SPIKE only considers the (most important) ABO blood type.

[48] As of 06.04.2022

[49] ISBT Working Party for Red Cell Immunogenetics and Blood Group Terminology (2021)

By draining all blood residues and pre-treating the donor organ, ABO incompatible organ transplants are feasible. In the first year after the procedure they show a higher risk of adverse reactions such as severe viral infections, post-operative bleeding, or antibody-mediated rejections. However, excluding this first year, long-term survival rates are comparable to ABO compatible donor–recipient pairs[50].

[50] Weerd and Betjes (2018)

Table 4.2: ABO blood transfusion compatibility

Blood Group	Can Receive From	Can Donate To
O	O	O, A, B, AB
A	O, A	A, AB
B	O, B	B, AB
AB	O, A, B, AB	AB

(iii) *Age.* Age disparity between organ donors and recipients influence transplant survival rates[51]. By observing the outcomes in all combinations of two age groups—*junior* participants aged below 55 years and *senior* participants aged 55 years or above—intra-categorical transplant showed the highest survival rates. Junior donors and senior recipients showed the next best outcomes and, finally, pairings with senior donors and junior recipients fared worst.

[51] Waiser et al. (2000)

(iv) *Sex.* Zhoua et al.[52] examined the influence of donor and recipient sexes on the transplant failure probabilities. The lowest survival rates witnessed for male recipient receiving female donor organs. The best chances for a successful transplant were observed for same-sex donor–recipient pairs, closely followed by female recipients receiving male donor organs.

[52] Zhoua et al. (2013)

(v) *Weight.* Clinical observations show higher allograft loss rates for recipients who receive an organ from a significantly lighter donor[53]. According to El-Agroudy et al.[54], this is causally linked as the donated kidney of a lighter donor might be unable to support sufficient organ function in a heavier recipient's body.

[53] Miller et al. (2017)

[54] El-Agroudy et al. (2003)

4.2 Design and Implementation of SPIKE

The formal objective for SPIKE is to fulfil the ideal functionality shown in Figure 4.2 in a provably secure fashion, while satisfying additional functional requirements. The ideal functionality to be translated into a MPC protocol assumes a perfectly ideal *Trusted Third Party* (TTP). Hospitals send their patients' medical and demographic data to the *Trusted Third Party* (TTP) which calculates exchange cycles that a) include the most pairs and b) exhibit the highest (aggregated) probability for transplant success. Finally, the TTP returns for each recipient the information about their donor to the respective medical institution. Note, that the

evaluation by medical experts is—of course—still required. We strive to accelerate the process, hence, help in fairly maximizing the available resources such as medical personnel, time, and donor organs.

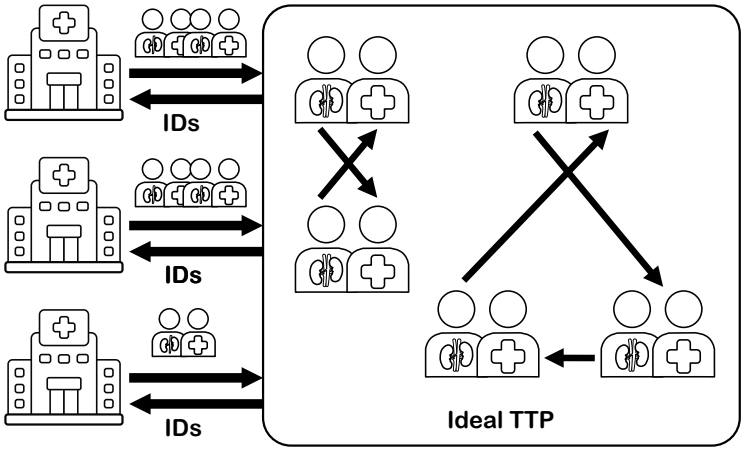


Figure 4.2: Ideal Functionality for a secure privacy-preserving Kidney Exchange Problem (KEP). This figure was created by the author and used in Birka et al. (2022), licensed under CC-BY.

Functional Requirements

AS SAID ABOVE, the list of functional requirements for a successful PPKEP includes more than the implementation of the ideal functionality. We propose the following requirements for a secure privacy-preserving KEP protocol:

- **Privacy.** The privacy-preserving KEP protocol must implement the same functionality as described in the ideal functionality without relying on a TTP—that is it must not leak any information beyond what can be inferred from the output.
- **Efficiency.** The privacy-preserving KEP protocol must achieve practical efficiency with respect to communication and computation. It must be able to run typical workloads in appropriate time on (the domain's) standard server hardware.
- **Decentralization.** The sensitive medical data of donors and recipients must not leave the respective medical institutions. The privacy-preserving KEP protocol must be conducted fully decentralized. This complies with the data minimization principle.
- **Adaptability for Medical Experts.** The privacy-preserving KEP protocol must be adaptable for medical experts to react to case-specific requirements and updated treatment guidelines. The protocol must be easily extendable to new factors and HLA groups while allowing the adjustment of their selection and relative importance.

SPIKE is designed to fulfill all above requirements.

4.2.1 Protocol Overview

SPIKE performs four separate algorithmic parts as displayed in Figure 4.1. In addition to conforming to the "separation of concerns" principle[55], this allows

[55] Dijkstra (1982)

storing (secret shared) intermediate values to reduce memory consumption and speeding up parts of the calculation. It is possible, for example, to calculate the best solution for multiple cycle lengths without re-calculating the compatibility graph. Due to this design, SPIKE is able to operate in a dynamic setting similar to the online system mode in Breuer, Meyer, and Wetzel (2022).

The first phase—the *compatibility matching*—generates the (weighted) adjacency matrix of the compatibility graph by evaluating the donors and recipients of all pairs according to the six biomedical factors introduced in Section 4.1. The second phase—*cycle detection*—calculates the total number of occurring cycles in the graph. Next, the *cycle evaluation* phase calculates the combined weight—that is transplant success probability—for each cycle and, finally, the forth phase—*solution evaluation*—finds the best set of (vertex-) disjoint cycles.

4.2.2 Compatibility Matching

The construction of the compatibility graph is performed by calculating the pairwise medical compatibility as a weighted sum (see Protocol 4.1). The evaluation of each medical factor—HLA mismatches, ABO, age, sex, and weight—are described in Subprotocols D.1 to D.5 in Appendix D.1. By weighting the individual contributions, medical examiners can exclude certain criteria and modify their individual relative importance. Note, that the circuit is built independently of the actual inputs. This implies, that the edge weights must be calculated for every vertex pair—that is $|\mathsf{pairs}| \cdot (|\mathsf{pairs}| - 1)$ times as generally $\mathsf{compG}_{ij} \neq \mathsf{compG}_{ji}$ and self loops are forbidden by construction. Depending on the result of the HLA crossmatch (Subprotocol 4.1)—the prohibitive compatibility factor—the respective entry in the adjacency matrix is set to the calculated weight or zero. The weighted sum evaluation contains five multiplications and additions per adjacency graph entry, hence, this protocol is evaluated in Arithmetic Sharing ($\mathcal{A}$) requiring conversions for the results of the compatibility evaluations.

Protocol 4.1: Protocol `computeCompatibilityGraph` computes the adjacency matrix of the compatibility graph. Note, that all compatibility factors are evaluated regardless of the HLA crossmatch status.

Function `computeCompatibilityGraph`$(\langle \mathsf{pairs} \rangle^{\mathcal{B}}, \langle \mathsf{w} \rangle^{\mathcal{A}})$**:**
  $\langle \mathsf{compG} \rangle^{\mathcal{A}} \leftarrow \mathsf{matrix} \in \{\langle 0 \rangle^{\mathcal{A}}\}^{|\mathsf{pairs}| \times |\mathsf{pairs}|}$
  for $i = 1 \dots |\mathsf{pairs}|$ **do**
    for $j = 1 \dots |\mathsf{pairs}|$ **do**
      $d \leftarrow \mathsf{pairs}[i].d$; `// Extract donor`
      $r \leftarrow \mathsf{pairs}[j].r$; `// Extract recipient`
      $\langle \mathsf{edge\_w} \rangle^{\mathcal{A}} \leftarrow \langle 1 \rangle^{\mathcal{A}} +$
        $\langle w \rangle^{\mathcal{A}}[0] \cdot \mathtt{b2a}(\mathtt{evalHLA}(\langle d.\mathsf{hla} \rangle^{\mathcal{B}}, \langle r.\mathsf{hla} \rangle^{\mathcal{B}})) +$
        $\langle w \rangle^{\mathcal{A}}[1] \cdot \mathtt{b2a}(\mathtt{evalABO}(\langle d.\mathsf{bg} \rangle^{\mathcal{B}}, \langle r.\mathsf{bg} \rangle^{\mathcal{B}})) +$
        $\langle w \rangle^{\mathcal{A}}[2] \cdot \mathtt{b2a}(\mathtt{evalAge}(\langle d.\mathsf{a} \rangle^{\mathcal{B}}, \langle r.\mathsf{a} \rangle^{\mathcal{B}})) +$
        $\langle w \rangle^{\mathcal{A}}[3] \cdot \mathtt{b2a}(\mathtt{evalSex}(\langle d.\mathsf{sex} \rangle^{\mathcal{B}}, \langle r.\mathsf{sex} \rangle^{\mathcal{B}})) +$
        $\langle w \rangle^{\mathcal{A}}[4] \cdot \mathtt{b2a}(\mathtt{evalWeight}(\langle d.\mathsf{weight} \rangle^{\mathcal{B}}, \langle r.\mathsf{weight} \rangle^{\mathcal{B}}))$
      $\langle \mathsf{compG} \rangle^{\mathcal{A}}[i][j] \leftarrow \mathtt{b2a}(\mathtt{matchHLA}(\langle d.\mathsf{hla} \rangle^{\mathcal{B}}, \langle r.\mathsf{ahla} \rangle^{\mathcal{B}}) > \langle 0 \rangle^{\mathcal{B}}$ **?**
        $\mathtt{a2b}(\langle \mathsf{edge\_w} \rangle^{\mathcal{A}})$ **:** $\langle 0 \rangle^{\mathcal{B}})$
  return $\langle \mathsf{compG} \rangle^{\mathcal{A}}$

While the HLA crossmatch functionality in Subprotocol 4.1 is straight forward, two opportunities for optimized circuit generation arise: All $|\mathsf{HLA}|$ `AND` gates in the first "loop" can be vectorized in a single *Single Instruction Multiple Data* (SIMD)

instruction. Furthermore, the fold of the resulting bit vector is constructed as a binary tree to reduce the circuit's multiplicative depth from $|\mathsf{HLA}|+1$ to $\log_2(|\mathsf{HLA}|)$. The output is inverted to align the boolean value to the semantic meaning. Due to the shallow construction of the circuit it is evaluated using Boolean GMW ($\mathcal{B}$).

```
1 Function matchHLA(⟨HLA_d⟩^B, ⟨aHLA_r⟩^B):
2     ⟨comp⟩^B ← [⟨0⟩^B]^|HLA|
3     for i = 1 ... |HLA| do // SIMD
4         ⟨comp⟩^B[i] ← ⟨hla_d[i]⟩^B ∧ ⟨ahla_r[i]⟩^B
5     ⟨combined⟩^B ← ORTree(⟨comp⟩^B)[1]
6     return ¬⟨combined⟩^B
```

Subprotocol 4.1: The `matchHLA` subprotocol determines general compatibility between donor and recipient by performing an HLA crossmatch.

4.2.3 Cycle Computation

To increase the efficiency of the following algorithmic parts of SPIKE, it is important to calculate (and reveal) the total number of cycles in the given compatibility graph. This provides a—in most cases—much smaller upper bound for loops iterating over cycles compared to the theoretical maximum number of cycles. This calculation is performed in Protocol 4.2.

```
1 Function determineNumberCycles(⟨compG⟩^A):
2     ⟨compG⟩^B ← a2b(⟨compG⟩^A)
3     ⟨uG⟩^A ← removeWeights(⟨compG⟩^B)
4     ⟨cG⟩^A ← pow(⟨uG⟩^A, cLen)
5     ⟨|cycles|⟩^A ← ⟨0⟩^A
6     for i = 1 ... |pairs| do
7         ⟨|cycles|⟩^A ← ⟨|cycles|⟩^A + ⟨cG⟩^A[i][i]
8     return ⟨|cycles|⟩^A
```

Protocol 4.2: `determineNumberCycles` calculates the number of cycles existing in the compatibility graph.

The technique for cycle count computation by calculating powers of the adjacency matrix (cf. Section 2.1.3) only works with unweighted—that is binary—adjacency matrices. Hence, the weighted matrix is stripped of its weights using Subprotocol D.6 and then raised to the cLen-th power. For the exponentiation a naïve matrix multiplication is used, as this algorithmic part's performance—even though exhibiting a cubic runtime complexity—is small compared to the following two parts (cf. Section 4.3.2). The number of cycles containing the respective vertex are encoded in the diagonal elements of the resulting matrix, hence the trace contains an upper bound for the total number of cycles in the graph. As a result of duplicate counting—cyclically shifted loops[56] are counted multiple times, even though enumerating the same cycle—it is only an upper bound. The duplicates are pruned later. Note, that even though the trace calculation could have been performed in a binary tree structure the costs associated with the additions are already negligible in $\mathcal{A}$ sharing.

[56] For example cycles (A,B,C) and (B,C,A) are the same, while (A,C,B) differs from those.

4.2.4 Cycle Evaluation

The third phase is concerned with filtering out all duplicate cycles and finding the most promising (unique) cycles in the graph—based on transplant success probability.

Protocol 4.3: The central protocol of SPIKE's third phase, `evaluateCycles`, removes duplicate cycles and finds the most promising cycles in the graph.

Function `evaluateCycles`($\langle \mathsf{compG} \rangle^{\mathcal{Y}}$)**:**
- $\langle \mathsf{allCycles} \rangle^{\mathcal{Y}}, \mathsf{cCycle} \leftarrow \emptyset$
- **for** $i = 1 \dots |\mathsf{pairs}|$ **do**
  - cCycle.`append`(i)
  - $\langle \mathsf{allCycles} \rangle^{\mathcal{Y}} \leftarrow$ `findCycles`($\langle \mathsf{compG} \rangle^{\mathcal{Y}}$, cCycle, $\langle \mathsf{allCycles} \rangle^{\mathcal{Y}}$, $\langle \mathsf{weight} \rangle^{\mathcal{Y}}$, $\langle \mathsf{valid} \rangle^{\mathcal{Y}}$)
  - cCycle.`remove`()
- $|\mathsf{allCycles}| \leftarrow$ `totalCycles`()
- $\langle \mathsf{sortedCycles} \rangle^{\mathcal{Y}} \leftarrow$ `kNNSort`($\langle \mathsf{allCycles} \rangle^{\mathcal{Y}}$, $|\mathsf{cycles}|$)
- $|\mathsf{unique}| \leftarrow \lfloor \frac{|\mathsf{cycles}|}{\mathsf{cLen}} \rfloor$
- $\langle \mathsf{filteredCycles} \rangle^{\mathcal{Y}} \leftarrow$ `removeDuplicates`($\langle \mathsf{sortedCycles} \rangle^{\mathcal{Y}}$)
- **return** $\langle \mathsf{filteredCycles} \rangle^{\mathcal{Y}}$

Protocol 4.3 is the central protocol of SPIKE's cycle evaluation phase. First it calculates recursively the (aggregated) cycle weight for all cycles and all "start" vertices (Subprotocol 4.2). This resulting list of all (weighted) cycles still contains duplicate cycles. As the de-duplication of cycles (Subprotocol D.8) is computationally expensive, the set of cycles is first partially sorted and reduced to the k "best" cycles with the largest combined edge weights (Subprotocol D.7)[57].

[57] The `kNN` circuit is inspired by Järvinen et al. (2019).

The first subprotocol—Subprotocol 4.2—creates a list of *all* cycles in the graph—duplicates included—with length cLen together with their combined edge weight. This cycle weight is a measure of the cycle's success probability. Note, that no normalization is required for this weight, as the possible maximum weight is only dependent on the length of the cycles and this length is the same for all cycles. This means, that all weights are relatively comparable and in the same scale without the need of a (computationally expensive) division.

Without being able to rely on often employed graph traversal techniques such as vertex coloring, SPIKE's MPC protocol must exhaustively inspect all paths of length cLen, checking whether it is a "valid" cycle—that is a closing edge exists—after $\mathsf{cLen} - 1$ edges. For best efficiency, this circuit is called recursively with parallel execution for each starting vertex. However, this requires to pass each call the index of the current cycle cCycle, the current secret shared weight $\langle \mathsf{weight} \rangle^{\mathcal{Y}}$, a secret shared edge counter $\langle \mathsf{valid} \rangle^{\mathcal{Y}}$ indicating when the desired cycle length is reached, and the output vector $\langle \mathsf{allCycles} \rangle^{\mathcal{Y}}$ to which all valid cycles and their weights are appended. As this subprotocol is somewhat complex, we will explain it in more detail.

The first check testes, whether the currently examined cycle already reached the desired length. If it has, the weight of the last edge is added to the cycle weight, and it is checked, whether the considered, closing edge is present in the graph. Invalid, open "cycles"—better called *paths*—are given a cycle weight of 0. As they do not contribute to the solution's optimization metric, they are, thus, never considered again. After appending the cycle to the $\langle \mathsf{allCycles} \rangle^{\mathcal{Y}}$ vector, the previous steps—the addition of the weight and counting of the edge—are reverted to prepare all parameters for the next call analyzing a different edge.

```
1  Function findCycles(⟨compG⟩^𝒴, cCycle, ⟨allCycles⟩^𝒴, ⟨weight⟩^𝒴, ⟨valid⟩^𝒴):
2      if |cCycle| == cLen then
3          ⟨weight⟩^𝒴 ← ⟨weight⟩^𝒴 + ⟨compG⟩^𝒴[cLen − 1][0]
4          ⟨valid⟩^𝒴 ← ⟨compG⟩^𝒴[cLen − 1][0] > ⟨0⟩^𝒴 ?
5                      ⟨valid⟩^𝒴 + ⟨1⟩^𝒴 : ⟨valid⟩^𝒴
6          ⟨addC⟩^𝒴 ← ⟨cLen⟩^𝒴 == ⟨valid⟩^𝒴
7          ⟨cWeight⟩^𝒴 ← ⟨addC⟩^𝒴 ? ⟨weight⟩^𝒴 : ⟨0⟩^𝒴
8          ⟨cycle⟩^𝒴 ← ⟨cCycle⟩^𝒴
9          ⟨allCycles⟩^𝒴.append((⟨cWeight⟩^𝒴, ⟨cycle⟩^𝒴))
10         revert()
11         ⟨weight⟩^𝒴 ← ⟨weight⟩^𝒴 − ⟨compG⟩^𝒴[cLen − 1][0]
12         ⟨valid⟩^𝒴 ← ⟨compG⟩^𝒴[cLen − 1][0] > ⟨0⟩^𝒴 ?
13                     ⟨valid⟩^𝒴 − ⟨1⟩^𝒴 : ⟨valid⟩^𝒴
14     else
15         if cCycle.contains(i) then
16             continue
17         else
18             ⟨weight⟩^𝒴 ← ⟨weight⟩^𝒴 + ⟨compG⟩^𝒴[−1][i]
19             ⟨valid⟩^𝒴 ← ⟨compG⟩^𝒴[−1][0] > ⟨0⟩^𝒴 ?
20                         ⟨valid⟩^𝒴 + ⟨1⟩^𝒴 : ⟨valid⟩^𝒴
21             cCycle.append(i)
22             ⟨allCycles⟩^𝒴 ← findCycles(⟨compG⟩^𝒴,
23                         cCycle, ⟨allCycles⟩^𝒴, ⟨weight⟩^𝒴, ⟨valid⟩^𝒴)
24             cCycle.remove()
25             revert()
26             ⟨weight⟩^𝒴 ← ⟨weight⟩^𝒴 − ⟨compG⟩^𝒴[−1][i]
27             ⟨valid⟩^𝒴 ← ⟨compG⟩^𝒴[−1][0] > ⟨0⟩^𝒴 ?
28                         ⟨valid⟩^𝒴 − ⟨1⟩^𝒴 : ⟨valid⟩^𝒴
29     return ⟨allCycles⟩^𝒴
```

Subprotocol 4.2: Subprotocol `findCycles` performs the "heavy lifting" by recursively evaluating the "cycleness" of all paths of length cLen and calculating an aggregate cycle weight in case of a closed loop.

Cycles that have not reached the desired length yet are first checked whether the current vertex is already part of the cycle. This is possible, because all paths of length cLen are evaluated. As we only allow vertex disjoint cycles, these cycles are discarded. Otherwise, the new vertex is added, the cycle weight is increased by the weight of the edge between the last vertex and the current, and $\langle \mathsf{valid} \rangle^{\mathcal{Y}}$ is set according to whether or not the edge exists in the graph. Next, a recursive call to `findCycles` happens to evaluate the subsequent possible edges. As in the last paragraph, the "state" of all parameters is reverted to allow the evaluation of the other possible edges. Finally, after all calls succeeded, $\langle \mathsf{allCycles} \rangle^{\mathcal{Y}}$ contains all cycles and their associated cycle weights, including cyclic rotations of the vertices.

The removal of duplicate cycles (Subprotocol D.8 in Appendix D.1) works efficiently, because both cLen and |cycles|, the total number of cycles in the graph, calculated in part two[58] of SPIKE, are public knowledge. It reduces the vector of all cycles to $\#\mathsf{unique} = \lfloor \frac{\#\mathsf{cycles}}{\mathsf{cLen}} \rfloor$ unique cycles where the first k elements are sorted according to decreasing cycle weight—that is the highest cycle weight in the first position. The public revelation of the total number of cycles |cycles| is considered acceptable, as it only reveals a very high-level property, aggregated over all vertices, thus, not revealing information regarding the compatibility graph's topology[59].

[58] See Protocol 4.2

[59] Except for fully connected or empty graphs—scenarios not at risk of leaking sensitive data.

The recursive execution and large amount of branching leads to multiplicatively deep circuits. For example, `findCycles` (Subprotocol 4.2) exhibits a depth of order $\mathcal{O}(|\mathsf{allCycles}| \times |\mathsf{cycles}| \times \mathsf{cLen})$. All (sub-)protocols in this algorithmic part of SPIKE exhibit a high depth, which is why this complete part is evaluated using Yao's Garbled Circuit ($\mathcal{Y}$).

4.2.5 Solution Evaluation

In the last part of SPIKE, the found cycles are combined to a complete solution with locally optimal combined weights. The calculation of a *globally* optimal solution is proven to be a $\mathcal{NP}$-hard problem[60], hence today computationally out of reach for numbers of participants relevant in practical application. Donors and recipients can only participate in one exchange cycle, that means that the solution set may only include vertices with either degree zero or two.

[60] Biró and Cechlárová (2007)

Protocol 4.4 describes this functionality. It delegates the determination whether a potential cycle contains already included vertex to `disjointSet` (Subprotocol D.10) and the choice of the best solution from the set of candidates to `findMaximumSet` (Subprotocol D.11). The performed tasks are mostly comparison-based "bookkeeping", hence, the efficient comparisons in $\mathcal{Y}$ are used.

4.2.6 Complexity Assessment

Table 4.3 shows the asymptotic runtime complexity for the four algorithmic phases of SPIKE. Overall, the runtime complexity is most importantly dependent on the number of pairs |pairs|, the number of considered HLA |HLA|, the length of cycles cLen, and the number of unique cycles |cycles|. Note, that the number of

```
1  Function evaluateSolution(⟨filteredCycles⟩^𝒴):
2      ⟨sets⟩^𝒴 ← ∅
3      ⟨weights⟩^𝒴 ← ∅
4      ⟨dummyC⟩^𝒴 ← {⟨|pairs|⟩^𝒴}^cLen
5      for i = 1 ... |unique| do
6          ⟨tempSet⟩^𝒴 ← ∅
7          ⟨tempSet⟩^𝒴.append(⟨filteredCycles⟩^𝒴[i][2])
8          ⟨weight⟩^𝒴 ← ⟨filteredCycles⟩^𝒴[i][1]
9          counter ← 1
10         for j = 1 ... |unique| do
11             if i == j then
12                 continue
13             ⟨cCycle⟩^𝒴 ← ⟨filteredCycles⟩^𝒴[j][2]
14             ⟨disjoint⟩^𝒴 ← disjointSet(⟨tempSet⟩^𝒴,
15                                ⟨cCycle⟩^𝒴)
16             ⟨vertices⟩^𝒴 ← ∅
17             ⟨vertices⟩^𝒴.append(⟨disjoint⟩^𝒴 ?
18                                ⟨cCycle⟩^𝒴 : ⟨dummyC⟩^𝒴)
19             ⟨weight⟩^𝒴 ← ⟨disjoint⟩^𝒴 ? ⟨weight⟩^𝒴 : ⟨0⟩^𝒴
20             ⟨tempSet⟩^𝒴.append(⟨vertices⟩^𝒴)
21             counter ← counter + 1
22         ⟨sets⟩^𝒴.append(⟨tempSet⟩^𝒴)
23         ⟨weights⟩^𝒴.append(⟨weight⟩^𝒴)
24     return findMaximumSet(⟨sets⟩^𝒴, ⟨weights⟩^𝒴)
```

Protocol 4.4: Protocol `evalSolution` combines possible cycles while checking for cycle disjointedness and returns the set of compatible cycles with the highest combined weight.

cycles is indirectly dependent on the number of pairs, however by revealing the upper bound of cycles in the graph in phase two (cf. Protocol 4.2), we are not required to evaluate all $\binom{n}{L}\frac{(L-1)!}{2}$ theoretically possible number of cycles[61]. Also note, that in the default configuration, |HLA| is 50.

[61] The proof of this equation is straight forward: let C_i^L be a cycle of length $L =$ cLen starting (and stopping) at vertex i: $C_i^L = (i \to [j \to k \to \ldots \to \ell] \to i)$, where i, k, ℓ are the other vertex indices in the cycle. Note, that the first and last vertex are identical. In the square bracket are $L-1$ vertices. Hence, there are $(L-1)!/2$ possibilities to cyclically shift the vertices in the square brackets. Finally, there are $\binom{n}{L}$ possible ways to choose all p vertices of a cycle, each having $(L-1)!/2$ cyclic shift variations.

	Name	Protocol	Time Complexity
Part 1	Compatibility Matching	Subprotocol 4.1	$\mathcal{O}(\lvert\mathsf{HLA}\rvert)$
		Subprotocol D.1	$\mathcal{O}(\lvert\mathsf{HLA}\rvert)$
		Subprotocol D.2	$\mathcal{O}(1)$
		Subprotocol D.3	$\mathcal{O}(1)$
		Subprotocol D.4	$\mathcal{O}(1)$
		Subprotocol D.5	$\mathcal{O}(1)$
		Protocol 4.1	$\mathcal{O}(\lvert\mathsf{pairs}\rvert^2 \times \lvert\mathsf{HLA}\rvert)$
Part 2	Cycle Computation	Subprotocol D.6	$\mathcal{O}(\lvert\mathsf{pairs}\rvert^2)$
		Protocol 4.2	$\mathcal{O}(\mathsf{cLen} \times \lvert\mathsf{pairs}\rvert^3)$
Part 3	Cycle Evaluation	Subprotocol D.9	$\mathcal{O}(1)$
		Subprotocol 4.2	$\mathcal{O}(\lvert\mathsf{pairs}\rvert^{\mathsf{cLen}})$
		Subprotocol D.7	$\mathcal{O}(\lvert\mathsf{cyclesSet}\rvert \times \mathsf{k} \times \mathsf{cLen})$
		Subprotocol D.8	$\mathcal{O}(\lvert\mathsf{cycles}\rvert^2)$
		Protocol 4.3	$\mathcal{O}(\lvert\mathsf{pairs}\rvert^{\mathsf{cLen}})$
Part 4	Solution Evaluation	Subprotocol D.10	$\mathcal{O}(\lvert\mathsf{cycles}\rvert \times \mathsf{cLen})$
		Subprotocol D.11	$\mathcal{O}(\mathsf{cycles}\rvert^2)$
		Protocol 4.4	$\mathcal{O}(\lvert\mathsf{cycles}\rvert^3 \times \mathsf{cLen}^2)$

Table 4.3: Complexity Assessment of all (sub-)protocols composing the SPIKE PPKEP.

Composing all individual protocols and subprotocols to perform SPIKE's full functionality, SPIKE's overall asymptotic runtime complexity becomes

$$\mathcal{O}(|\mathsf{pairs}|^2 \times |\mathsf{HLA}| + \mathsf{cLen} \times |\mathsf{pairs}|^3 + |\mathsf{cycles}|^3 \times \mathsf{cLen}^2).$$

4.3 Performance Experiments

To gauge the practicality of SPIKE, we evaluate the performance metrics of SPIKE in experiments simulating a variety of application scenarios. Lastly, we examine SPIKE's runtime performance in comparison with the current state-of-the-art and inspect the runtime cost incurred by the extended medical compatibility matching.

4.3.1 Test Setup

We benchmarked SPIKE's performance in a lab environment using two servers with Intel Core i9-7960X CPUs, 128 GiB RAM each and a local 10 Gbit/s network connection with a median latency of 0.2 ms. All reported measurements are averaged over 10 runs.

SPIKE is designed for the application in different environments. This is reflected in the benchmarks by providing experimental data for two different network settings: *Wide Area Network* (WAN) and *Local Area Network* (LAN). Additionally, SPIKE is compared to the runtime performance of the current state-of-the-art[62]. As the software implementation of their work was not available at the time of writing, we replicated their published network setting with 1 Gb/s bandwidth and 1 ms of latency and compared SPIKE with the published runtimes. The details and reasoning behind the WAN and LAN settings are given in Appendix B, the used network parameters are shown in Table 4.4.

[62] Breuer, Meyer, Wetzel, and Mühlfeld (2020); Breuer, Meyer, and Wetzel (2022)

Table 4.4: Network parameters for the experimental evaluation of SPIKE

Setting	Bandwidth	Latency
WAN	100 Mbit/s	100 ms
LAN	10 Gbit/s	0.2 ms
Comparison	1 Gbit/s	1 ms

4.3.2 Performance Benchmarks

The total runtime of SPIKE for varying numbers of pairs, the two described network settings, and cycle length $L = 2$ and $L = 3$ are shown in Figure 4.3. The full results are shown in Tables D.2 to D.7 in Appendix D.2.

For longer cycles—$L \geq 3$—RAM usage limited the experimental evaluation. The benchmarks were performed up to RAM exhaustion and then extrapolated based on a power-law fit of the observed experimental data[63]. The extrapolated graphs are drawn with a dashed line. The fit coefficients are shown in Appendix D.3. The sudden runtime increase for $L = 3$ between 12 and 13 can be explained by the occurrence of memory swapping with the associated performance hit.

As expected, a polynomial relationship between the number of pairs and the overall runtime can be observed. For $L = 2$ the calculation of 40 participating donor–recipient pairs finishes in under 4 min in the *LAN* setting, and in under 40 min in the *WAN* setting. This increase of around one order of magnitude between a high-performance network and a severely degraded connection demonstrates the real-world applicability of SPIKE across multiple deployment settings and even with

[63] All fits used the Trust-Region algorithm and the model function $f(x) = ax^b + c$ with fit-determined parameters a, b and c. The model is based on the complexity assessment in Section 4.2.6.

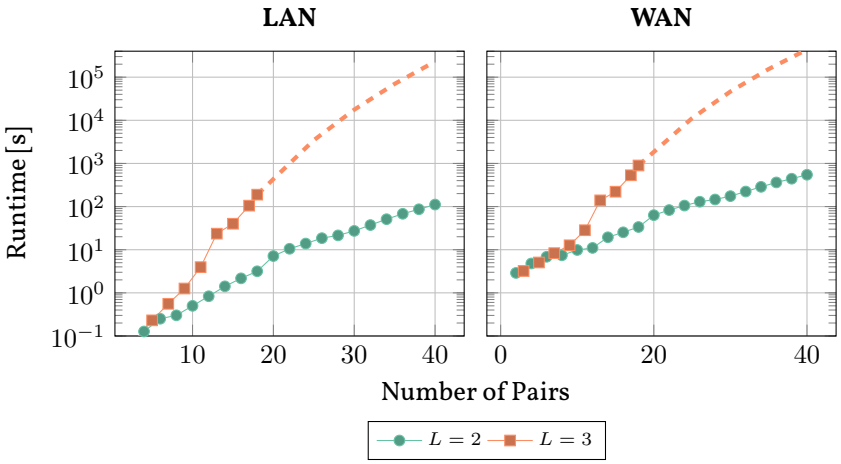


Figure 4.3: Overall runtime of SPIKE for cycle lengths $L = 2$ and $L = 3$ in both network scenarios. The dashed line shows the extrapolated power function for $L = 3$.

residential internet connections. For larger cycle lengths ($L = 3$) the runtime increases significantly. Even then, the computation completes in around 1 h for 25 pairs.

The separation of the overall runtime for $L = 2$ into the individual algorithmic parts' contribution in Figure 4.4 shows, that medical compatibility testing, graph creation, and the computation of the number of cycles quickly become negligible compared to cycle and solution evaluation. Note the logarithmic scale on the y-axis. In an online mode of operation, where setup and online phase are executed separately, a 134 % performance increase can be achieved, as only the online phase times are relevant in this scenario.

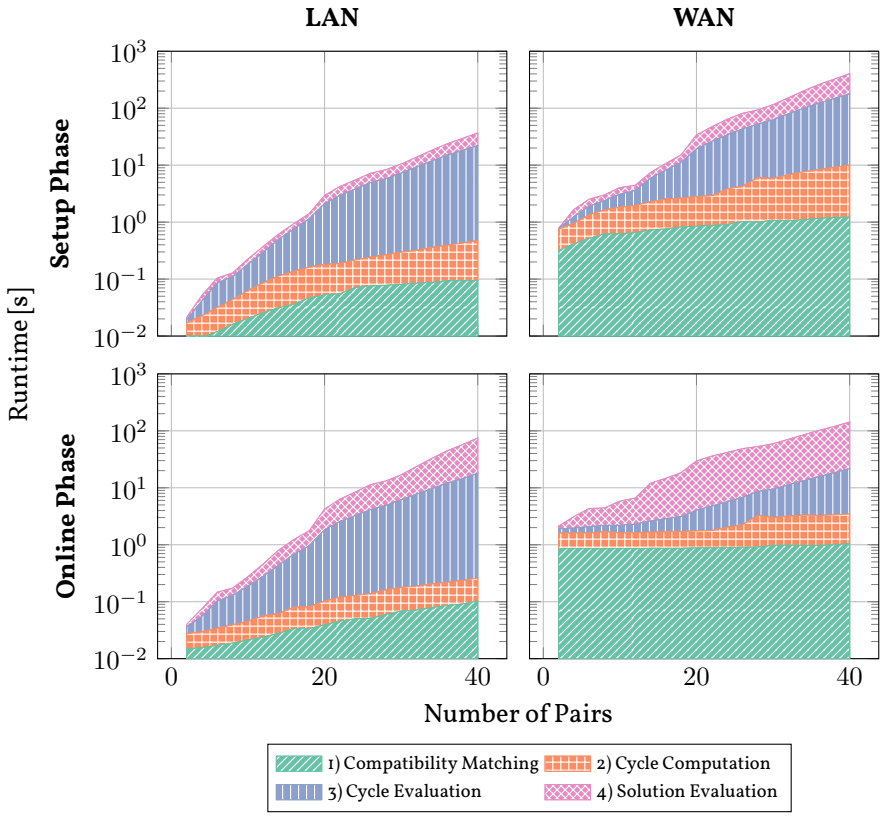


Figure 4.4: Runtime composition of SPIKE for $L = 2$ separated by algorithmic parts, protocol phase, and network setting.

Comparison to State-Of-The-Art

FIGURE 4.5 SHOWS THE overall runtimes of SPIKE for $L = 2$ and $L = 3$ in comparison to the two implementations from BREUER, MEYER, WETZEL, AND MÜHLFELD [64] using the same network setting as published—1 Gb/s bandwidth and 1 ms of latency. The first state-of-the-art implementation Breuer, Meyer, Wetzel, and Mühlfeld (2020) uses a Pallier-based Threshold HE scheme and allows the arbitrary choice of cycle lengths. The displayed runtimes correspond to $L = 3$, the highest benchmarked cycle length in the original publication. The second implementation— Breuer, Meyer, and Wetzel (2022)—only allows $L = 2$ and is based on a three-party semi-honest *Shamir's Secret Sharing* (SSS) MPC protocol implemented using the MP-SPDZ[65] framework. As both implementations are not publicly available, the performance data are taken from the referenced publications.

[64] Breuer, Meyer, Wetzel, and Mühlfeld (2020); Breuer, Meyer, and Wetzel (2022)

[65] Keller (2020)

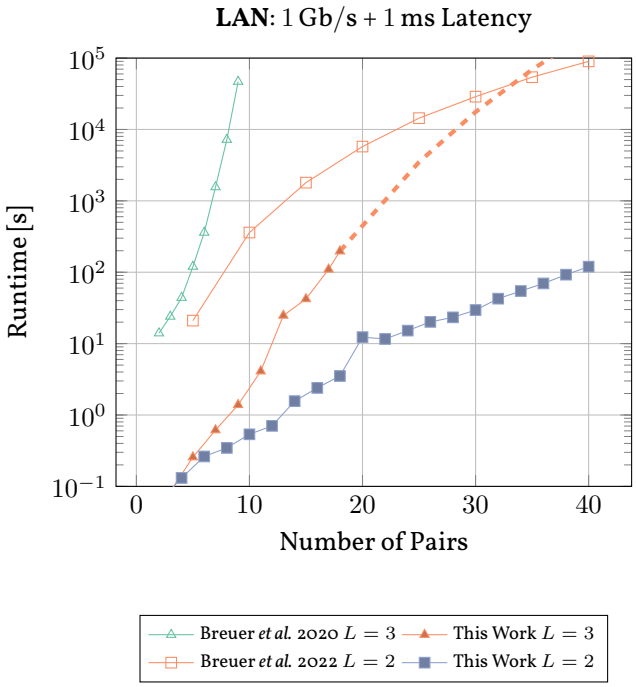


Figure 4.5: Runtime comparison between this work with cycle lengths $L = 2$ and $L = 3$, and both Breuer, Meyer, Wetzel, and Mühlfeld (2020) ($L = 3$) and Breuer, Meyer, and Wetzel (2022) ($L = 2$). All measurements use a LAN network setting with 1 Gb/s bandwidth and 1 ms latency. The dashed line shows the extrapolated power function for our algorithm at $L = 3$.

SPIKE and the SSS based state-of-the-art follow a polynomial pairs–runtime relation, while the HE-based implementation scales exponentially. For $L = 3$ and 9 pairs—the maximum number of pairs benchmarked in the original publication—SPIKE achieves a $29{,}828\times$ speedup. Compared to the SSS-based protocol and $L = 2$, SPIKE achieves a $414\times$ performance improvement.

Runtime Impact of Additional Medical Factors

TO IMPROVE THE robustness of the calculated solution—that is lower the probability of a failing edge—SPIKE compares more medical criteria for compatibility determination than all other privacy-preserving kidney exchange protocols[66]. Even though the performance impact of the compatibility assessment and graph generation is negligible compared to the cycle and solution evaluation (cf. Figure 4.4), we experimentally compared the runtime difference between the two compatibility criteria used by BREUER, MEYER, WETZEL, AND MÜHLFELD

[66] See Section 4.1.2

and SPIKE's full set of six criteria. As the graphs of the results displayed in Figure 4.6 show, both curves assume nearly the same slope after a short "transient phase" for the full set of medial factors and small numbers of pairs. The "baseline" runtime caused by the network delay is clearly observable in the graph of the *WAN* network setting.

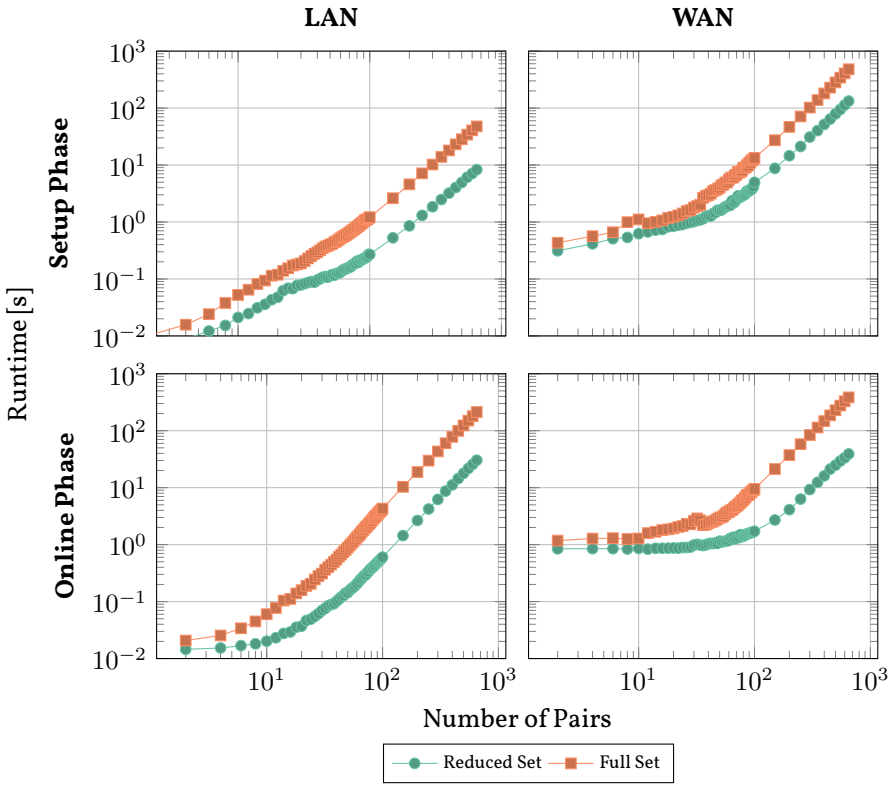


Figure 4.6: Runtimes of the compatibility matching for $L = 2$ and both the reduced set of criteria—as implemented by Breuer, Meyer, Wetzel, and Mühlfeld (2020)—and SPIKE's full set of criteria. The remaining algorithmic parts are not affected by this choice.

4.4 Privacy- and Security Setting

SPIKE is implemented using the ABY[67] framework for secure Two-Party Computation. It provides semi-honest security—a corrupted party is assumed to still follow the protocol while trying to learn secret information—preventing against two major security concerns: First, the inadvertent disclosure of sensitive data or curious personnel. Second, in case of a security incident at one of the participating parties only this party's data can be exfiltrated, compared to a centralized computation where *all* data might be disclosed.

[67] Demmler, Schneider, and Zohner (2015)

The outsourcing scenario allows arbitrary numbers of data sources performing the computation via two computation parties. As described in Section 2.3.6, this adds a *non-collusion* assumption to the security model—if both computation parties collude the security of the complete system is compromized—however, no such assumption exists for the data sources. The privacy guarantees hold with any numbers of colluding or maliciously adversarial data sources. Of course, the computation of a correct result relies on the truthful execution of the protocol and malformed inputs might (very likely) lead to incorrect results. For details regarding MPC's place in a comprehensive data protection model please see Section 2.2.

Semi-honest security is certainly not sufficient for every application—for example the European Data Protection Board recommends active security for computations with parties in different jurisdictions[68]. However, for the intended setting of SPIKE, namely the joint computation among fixed, national medical institutions, semi-honest behavior and non-collusion can be enforced by law and policy.

[68] European Data Protection Board (2021)

The primary security concern is the privacy of all patient's data and their association to the numerous data sources. The *number* of donor–recipient pairs and an upper bound of the total number of cycles in the graph are treated as public information, substantially increasing the performance of the system. If in some case the total number of patients is considered sensitive, it could be obfuscated by padding each party's input to a fixed length using "dummy" entries—incurring the expected performance hit due to a larger number of pairs than functionally necessary.

4.5 Outcome and Prospects

With SPIKE we introduced a protocol for the privacy-preserving solution of the kidney exchange problem. Empirical data support the feasibility and applicability of the protocol for real-world applications. SPIKE meets all proposed functional requirements with regard to security and privacy, efficiency, decentralization, and adaptability, flexibly supporting medical professionals in the fulfillment of their tasks.

More specifically, SPIKE enables the periodic batch-processing for cycle lengths of $L = 3$ with large numbers of donors and recipients involved. SPIKE achieves a $30{,}000\times$ and $400\times$ speedup compared to the current state-of-the-art[69] for cycle lengths of $L = 3$ and $L = 2$, respectively, completing within a total runtime of under 4 min for 40 pairs at $L = 2$ and around 1 h for 25 pairs at $L = 3$. The real-world feasibility with regard to performance holds even under suboptimal network connectivities, such as residential internet connections, allowing local residential nephrology experts and regional hospitals to directly participate in kidney exchanges. This could considerably increase the access and the quality of rural kidney replacement care. As the protocol still requires substantial communication sizes, the participation in a two-party computation using metered or cellular data connections is not recommended. However, the participation as a data source in an outsourced SPIKE computation over metered or cellular connections is reasonable.

[69] Breuer, Meyer, Wetzel, and Mühlfeld (2020); Breuer, Meyer, and Wetzel (2022)

Privacy of patients' medical and demographic data is the primary objective. SPIKE's security guarantees rely on formally defined assumptions of the hardness of mathematical problems, protecting the input data using state-of-the-art provably secure cryptographic techniques. By fully decentralizing the computation the privacy risk regarding data leaks both due to accidental data disclosure as well as security incidents at one of the participating parties is substantially reduced. This holds especially in comparison to central data repositories and trust agencies.

As transplantation medicine is a complex and fast evolving field, the algorithmic evaluation must allow the flexible adjustment based on expert assessments. SPIKE allows the weighting of the individual compatibility assessment subprotocols and the modification and extension of considered HLA to react to specific case requirements or updated evidence-based guidelines. By providing an open source implementation and clear architectural boundaries, the addition of user-defined criteria and functionalities is easily achievable.

While meeting all formal requirements, SPIKE leaves room for further improvements and novel research opportunities. Due to the protocol's optimization for runtime performance, the high memory consumption is expected. Furthermore, larger medical institution are usually able to scale hardware without issues, nevertheless, this is not ideal for residential experts and regional hospitals. By developing algorithms for graph cluster batch processing and space-optimized data structures, the memory utilization may be substantially lowered. The adoption of recent MPC-based graph analysis and -structure approaches— Araki et al.[70], for example—might allow the inclusion of more advanced and more efficient graph algorithms for cycle detection, such as the Bellman-Ford[71] or Floyd-Warshall[72] algorithms. For $L = 2$ maximum matching algorithms, such as the Hungarian Method[73], the Blossom[74] algorithm, or the Hopcroft-Karp[75] algorithm could be pursued. Lastly, for real-world adoption widespread medical standards, such as HL7 FHIR R4[76], the addition of audit- and authentication capabilities, the development of deployment packages, and full (legal) documentation must be pursued. However, as an academic research project resulting in a prototypical software artifact, this is not in the scope of this work.

By advancing the state-of-the-art in privacy-protecting graph applications in the field of kidney transplantation medicine, we hope to increase the quality of—and in rural areas the access to—possibly live changing procedures.

[70] Araki et al. (2021)

[71] Bellman (1958); Ford Jr (1956)

[72] Floyd (1962); Warshall (1962)

[73] Kuhn (1955)

[74] Edmonds (1965)

[75] Hopcroft and Karp (1973)

[76] https://www.hl7.org/fhir/R4/

CHAPTER 5
Secure Record Linkage

Data quality is an important requirement of meaningful data analysis. While assuring high data quality in local data is a strenuous task, it becomes even more challenging for distributed datasets located in multiple owner domains. Distributed analysis in the biomedical field suffer from this issue in particular, as most data transfer is strictly regulated.

One basic task of data quality assurance—de-duplication—is known as *record linkage* in medical informatics, as records belonging to the same human being are being linked between databases or institutional boundaries. The features used for identifying the individual—*Identifying Personal Data* (IDAT)—are themselves protected under various legal regulations, such as *Health Insurance Portability and Accountability Act* (HIPAA) or the *General Data Protection Regulation* (GDPR). Hence, either substantial legal contracts for independent trustee operations or privacy-preserving analysis protocols must be used to mitigate any re-identification risk. Of course, central fiduciary repositories constitute a "single point of failure"—data exfiltration due to a security incident put the privacy of all held records from multiple data owners at risk. Hence, our focus is on privacy-preserving decentralized record linkage algorithms.

Record linkage in rare disease analyses

WHILE RECORD LINKAGE often is used as a method for data quality assurance, it is an important prerequisite for the aggregation of vertically partitioned data[1] as well. Some fields of research rely predominantly on some form of record linkage. One example is the field of rare diseases. One of the more than $6{,}000$ rare diseases, that is diseases with a prevalence of under 5 in $100{,}000$ (based on the definition of the European Union[2]), affect, albeit individually rare, around 5 % of the German population[3]. The medical research in this field faces multiple challenges caused by data sparsity. One obvious challenge is, that for most studies patient records from multiple institutions must be aggregated, as a single hospital most certainly does not observe enough cases for valid statistical analysis. Due to the difficulties in diagnosing rare diseases, most patients were registered and treated in multiple hospitals leading to notable patient overlaps in registries and hospital information systems. Record linkage and the related record-linkage based set intersection cardinality is used to exclude the duplicates in those joint study databases to reduce the possible bias made worse by the small cohort sizes. Additionally, it can be used as an analysis in itself, for example examining the "trajectory" of patients through the public health system and their path to a sufficient treatment.

[1] Vertical partitioned data include different attributes of the same patients in multiple datasets. The opposite is horizontal partitioning, where the full set of attributes are present in each database but for different patients. A fitting visualization is a tabular view of the dataset where the partition direction designation indicates, whether the datasets are "sliced" between rows or columns.

[2] Commission of the European Communities (2008)

[3] Bundesgesundheitsministerium (2021)

This chapter of the dissertation draws upon work published in STAMMLER, S., KUSSEL, T., SCHOPPMANN, P., STAMPE, F., TREMPER, G., KATZENBEISSER, S., HAMACHER, K., LABLANS, M. (2020) "Mainzelliste SecureEpiLinker (MainSEL): Privacy-Preserving Record Linkage using Secure Multi-Party Computa-

tion". *Bioinformatics*, and Kussel, T., Brenner, T., Tremper, G., Schepers, J., Lablans, M., Hamacher, K. (2022) "Record Linkage based Patient Intersection Cardinality for Rare Disease Studies using Mainzelliste and Secure Multi-Party Computation". *Submitted to BMC Journal of Translational Medicine*. The author was deeply involved in all aspects of both publications contributing significantly to the design, implementation, experimental setup, and manuscript of the first publication, and being the primary contributing author in all those aspects of the latter.

Record Linkage against "Pure" Set Intersection

The most efficient way to determine common patients or calculate the number of common patients in a privacy-preserving fashion are specialized *Private Set Intersection* (PSI) and *Private Set Intersection Cardinality* (PSI-C) protocols, for example Pinkas et al. (2018) and De Cristofaro, Gasti, and Tsudik (2012). However, while very efficient, these protocols are only applicable to perform the deduplication using a trans-institutional unique *master patient index*. In some countries, unique identifiers such as the social security number or the patient's health insurance number could be used. In Germany, however, only statutory health insured patients are issued such an insurance ID—less than 90 % of all patients[4]. While slower than those specialized PSI protocols, our record-linkage based protocols utilize the full set of identifiable patient data to achieve optimal record linkage quality—as discussed in Section 5.5.1—while maintaining privacy.

[4] GKV-Spitzenverband (2021)

Record linkage as a graph problem

Considering the features as vertices in a multipartite graph, connected according to the feature combinations of the patients IDAT, we strive to find subgraph isomorphisms between multiple graphs. These multiple graphs are arranged as a peer-to-peer multi-layer "meta" graph. This already is a hard computational problem, exhibiting superpolynomial asymptotic complexity. To make matters more challenging, real-world data quality makes it impractical to find exact subgraphs. People change their last names (e.g., due to marriages), birthdays and birth months might be swapped, and—of course—the data entry is prone to transcription errors. To enable linking between databases with thousands of noisy, incomplete records nevertheless, we transform the problem to a measure problem and implement privacy preserving algorithms to calculate feature wise, similarity measures which are combined to a record similarity measure.

The MainSEL Record Linkage System

In "usual" record linkage operation, correlated (second order) pseudonyms are generated according to the match status. These pseudonyms can be used for (later) data transfer processes or as an analysis in itself, e.g., to track a patient's trajectory through medical institutions (with consent). One "aggregate" statistic resulting from record linkage processes useful in other research questions is the *cardinality* of matches—the set intersection cardinality based on probabilistic, fuzzy matching. We implemented both modes of operation extending the well-known pseudonymization framework Mainzelliste [5] and achieved substantially stronger privacy guarantees compared to other record linkage systems, practical runtimes by careful optimization of the developed *Secure Multi-Party Computation* (MPC) protocols, and a polished, easy to deploy open source software

[5] Lablans, Borg, and Ückert (2015)

package—*Mainzelliste Secure EpiLinker* (MainSEL). This software was tested and vetted throughout multiple medical use cases and its nationwide rollout is discussed for the next funding phase of the *(German) Medical Informatics Initiative* (MI-I).

5.1 Related Works

Record linkage algorithms are an established field of scientific research for over fifty years[6]. With progressing digitalization, data protection concerns have become the focus of discourse, shifting the research endeavors to *Privacy-Preserving Record Linkage* (PPRL) techniques. Additional areas of research are scalability of record linkage systems[7] and the inclusion of additional data types, such as clinical and genomic data[8].

[6] Fellegi and Sunter (1969)

[7] Vatsalan, Sehili, et al. (2017); Rohde et al. (2021)

[8] Baker et al. (2018)

Many established PPRL implementations[9] employ a (central) comparison of Bloom filters containing hashes of the IDAT or hashes of combined fragments of IDAT[10]. Unfortunately, both the basic cryptographic method—Bloom filters of hashed IDAT and *Hash-based Message Authentication Codes* (HMAC)—and the centralized system architecture are vulnerable. As any centralized approach the integrity of the *Trusted Third Party* (TTP) is a single point of failure with the potential to compromise the privacy of *all* data records. Furthermore, frequency analysis and cryptanalysis vulnerabilities against Bloom filter-based PPRL systems are described in the literature[11]. While recent developments claim resistance against those known attacks[12], the development of previously unknown exploits has to be expected[13].

[9] Lablans, E. Schmidt, and Ückert (2018); Heidt, Hund, and Fegeler (2021)

[10] Schnell, Bachteler, and Reiher (2009); Vatsalan, Christen, and Verykios (2013)

[11] Kuzu et al. (2011); Vatsalan, Sehili, et al. (2017); Christen et al. (2017)

[12] Schnell and Borgs (2018)

[13] Zabicki and Ellis (2017)

An alternative to Bloom filter-based PPRL algorithms are MPC-based techniques—starting with Laud and Pankova 's[14] entry to the 2017 iDASH competition[15]—only evaluating exact matches between records—to the previous state-of-the-art by Lazrig et al.[16] and *Mainzelliste Secure EpiLinker* (MainSEL) described in this chapter. This class allows the joint record linkage without central component with clearly defined security guarantees, possibly—legal analysis pending—even record linkage without explicit patient consent, e.g., using records of deceased patients.

[14] Laud and Pankova (2018)

[15] http://www.humangenomeprivacy.org/2017/

[16] Lazrig et al. (2018)

For a recent survey of PPRL solutions, see Gkoulalas-Divanis et al. (2021)[17], for a discussion on the relationship between cryptographic security and *Statistical Disclosure Control* (SDC) for record linkage see X. He et al. (2017)[18].

[17] Gkoulalas-Divanis et al. (2021)

[18] X. He et al. (2017)

Related to parts of the work described here are *Private Set Intersection Cardinality* (PSI-C) protocols[19]. While highly efficient and optimized, PSI and PSI-C algorithms enable only the calculation of exact intersections, rendering them unsuitable for noisy real-word patient datasets. The PPRL-based private set intersection cardinality in this chapter utilizes all error-tolerance mechanisms employed in probabilistic record linkage.

[19] De Cristofaro, Gasti, and Tsudik (2012); Kolesnikov, Kumaresan, et al. (2016); Kales et al. (2019)

5.1.1 Comparison to State-of-the-Art

Lazrig et al.[20] published a PPRL system based on a MPC Dice-coefficient similarity comparison of Bloom filters—following Schnell, Bachteler, and Reiher [21]. While this methodology for fault-tolerant matching is similar to MainSEL's approach—Mainzelliste's record linkage is based on a combination of ideas from Schnell, Bachteler, and Reiher (2009) and Contiero et al. (2005)[22]—MainSEL differs fundamentally in two key aspects.

[20] Lazrig et al. (2018)

[21] Schnell, Bachteler, and Reiher (2009)

[22] Contiero et al. (2005)

First, Lazrig et al. use four Bloom filters combining different fields of the IDAT or fractions thereof. These combinations are determining using expert knowledge to estimate the most probable encountered error scenarios. For comparison, MainSEL follows a much more general approach of comparing *every* IDAT field in a separate Bloom filter without relying on pre-determined (fragmentary) field combinations. Additionally, this enables MainSEL to deterministically and reliably handle missing field values and erroneously interchanged fields—the latter by introducing configurable exchange groups.

Second, the PPRL solution of Lazrig et al. does *not* perform the whole record linkage operation in a MPC setting. Only the calculation of the Bloom filter comparisons is performed as a multi-party computation, especially the post-processing, that is the match-classification based on the *publicly reviled* similarity scores, is performed publicly, potentially leaking information[23]. MainSEL performs *all* operations in a secure and privacy-preserving fashion, considering only the total number of records as public information. Note, that the MPC sections of Lazrig et al. (2018) constitute a subset of MainSEL's functionality, specifically Circuit 5.5.

[23] The knowledge of the per-filter scores makes the identification of partial information possible. For example by observing, that only Bloom filters containing the last name do not match one can infer that the same patient with a different last name is registered in the other database, not only leaking this information but possibly maritial status, etc.

To these two major differences in the principles of operation, many details differ between the two PPRL solutions. MainSEL implements a novel tie-solving order to choose the best matching candidate when multiple records exhibit the same similarity score. This order is stable and semantically meaningful in the presence of empty fields. Furthermore, MainSEL allows the configuration of the used comparison mechanism. The probabilistic Dice-Bloom comparison is useful for fields containing strings but inappropriate for numeric types such as integers or floats. MainSEL's usage of the ABY[24] MPC framework[25] results in a very adaptable and extendable implementation providing four protocol variants, whereas Lazrig et al. implements a custom Yao's Garbled Circuits protocol. Lastly, the solution of Lazrig et al. compares each individual Bloom filter's similarity to a threshold and assumes a match if at least one similarity exceeds the threshold. The record-score aggregation in MainSEL weighing each individual field allows not only statistically more meaningful similarities but provides flexibility to employ MainSEL in different applications and with (semantically) different record types.

[24] Demmler, Schneider, and Zohner (2015)

[25] see Section 2.3.7

In contrast to Lazrig et al., for MainSEL we chose to not implement blocking techniques to reduce the computational workload. A result of this decision is the extremely high privacy level achieved by MainSEL, as many blocking techniques—especially blocking based on *Locality-Sensitivity Hashing* (LSH)

and *Differential Privacy* (DP)—are not composable with MPC techniques without jeopardizing the security level[26]. Even using DP-based blocking in a local preprocessing stage[27]—the approach chosen by LAZRIG ET AL.—fail to achieve strong overall security guarantees and might leak sensitive information. See X. HE ET AL.[28] for a detailed discussion and security proofs of the composition of DP and MPC in a record linkage context.

[26] For DP this is intuitively understandable, as MPC is concerned with finding *exact* results of a given calculation, while DP aims to only reveal (bounded) approximate results.

[27] Inan et al. (2010)

[28] X. He et al. (2017)

All this makes MainSEL's security guarantees and match quality unique. For an empirical analysis of MainSEL's record linkage quality—including a comparison to Lazrig et al. (2018)—see Section 5.5.1. By extending Mainzelliste, MainSEL provides not only a record linkage solution, but a full pseudonymization and ID management solution.

5.2 Record Linkage

The process of finding duplicates in one or between many databases is called *record linkage*. Most of the time the term is used in a medical informatics setting to describe the identification of duplicate patients based on IDAT such as name, birthdate, or address. However, the process is extendable to arbitrary types of data, as long as some meaningful similarity measure can be defined (see Section 5.7). The record linkage process can be divided into two phases: First, a similarity calculation between all records and second, the classification of the compared records as duplicates (*match*) or unrelated (*non match*)[29].

[29] Fellegi and Sunter (1969)

As described in the introductory text of this chapter, the record linkage problem can be modeled as a graph problem of finding probabilistic subgraph isomorphisms on multipartite (hyper-)graphs. Possibly interchanged vertices, such as first-, last-, and surname, build a fully connected clique.

Let A be a $n \times n$ adjacency matrix and B a $m \times m$ (sub) graph, then the optimization problem

$$P_{\text{EX}} = \arg\min_{P} ||B - PAP^T||_2^2$$

is called the *subgraph isomorphism problem* with the $m \times n$ (pseudo) permutation matrix P_{EX}. This optimization problem finds exact subgraphs. M. SCHMIDT ET AL.[30] developed a probabilistic subgraph isomorphism algorithm—"SICOR"—as a *ribonucleid acid* (RNA) similarity algorithm. The abstract nature of the underlying mathematical problem however, allows the application to other graph structures. SICOR employ a convexly relaxed problem statement transforming the permutation matrix P_{EX} into a pseudo bi-stochastic matrix S of the same dimension that is found via local optimization methods. The defined similarity measure uses the re-projection of S to the space of the (pseudo) permutation matrix P_{EX}. Unfortunately, the runtime performance of the SICOR algorithm is prohibitive for the usage in a MPC setting, as the clear text analysis of graphs with 200 vertices and a subgraph size of 10 vertices takes tens to hundreds of seconds. The estimated graph sizes in real-world applications are around $10{,}000$ vertices and small subgraphs with around 10 vertices—8 to be precise, according to the fields defined by Mainzelliste.

[30] M. Schmidt et al. (2020)

To find a practical solution note, that the graph structure represents a fully normalized structure of a (relational) database. The main benefits of the graph model over such a record-based "database view" are:

- The resilience against interchanged data fields, as they are considered unordered by forming a fully connected clique, and
- the independence from the actual included data types, as the process is defined on an abstract data structure.

[31] Contiero et al. (2005)

The *EpiLink* record linkage algorithm[31] is able to include both aspects by including all permutations between defined fields into the similarity calculation (*Exchange groups*, cf. Section 5.2.1) and by being easily extendable to include similarity measures between other data types. This way we can transform the graph problem into a record-wise measure problem, a concept useful for the efficient inclusion of other graph structures in Section 5.7.

Formal objective statement

THE PRINCIPAL OBJECTIVE of record linkage is to determine the similarities between a record x and a dataset with N records $\{y^j\}_{0\leq j<N}$ (abbreviated $\{y^j\}$), and output the best matching database record. The similarity between two records is given by a function $S(x, y)$, resulting in a similarity score between 0 and 1. A similarity of 1 signifies identical records, 0 complete independence. The similarity score of the best matching entry in the database is then compared to two thresholds $0 < T_1 \leq T_2 \leq 1$. Records with scores below the first threshold are considered distinct and thus classified as non matches. For scores above the *second* threshold, the two entries are classified as a likely *match*. If the threshold falls between the two thresholds[32] the records are marked as a *tentative match*. This primary functionality is called bestMatch$(x, \{y^j\})$.

[32] For brevitie's sake we will consider only one threshold T from here on, as the extension of all formulas and circuits to more than one threshold value is trivial.

To enable, for example, the privacy-preserving cohort size estimation, we introduce a second functionality. This functionality *counts* the number of records classified as matches in a bestMatch$(\{x^k\}, \{y^j\})$ comparison between two datasets—the *match-cardinality* or *intersection-cardinality*. We denote this functionality with matchCardinality$(\{x^k\}, \{y^j\})$.

In conclusion, we care about the following two functionalities:

$$\begin{aligned}\text{bestMatch}(x, \{y^j\}) := (j^*,\ S(x, y^{j^*}) > T) \\ \in \{0, \ldots, N-1\} \times \{0, 1\}, j^* := \underset{0\leq j<N}{\arg\max}(S(x, y^j))\end{aligned} \tag{5.1}$$

$$\begin{aligned}\text{matchCardinality}(\{x^k\}, \{y^j\}) := |\{k : \exists j : S(x^k, y^j) > T\}| \\ \in \{0, \ldots, \min(M, N)\}\end{aligned} \tag{5.2}$$

To be a useful advancement for real-world applications in (bio)medical research, the following constraints must be satisfied:

- The sensitive patient data must be protected, no information allowing the re-identification of the patient may leave the data owner's network.
- No trusted party should be used for record linkage. This constraint simplifies the patient consent process as well and, thus, allow recruiting more patients for the research for rare diseases.
- Even in the case of an IT security incident in one of the participating party's protected networks, the patients' data of all other institutions must remain private.
- The proposed method must exhibit high precision and full coverage of dataset comparison in order to operate satisfactory in sparse data environments, e.g., the field of rare diseases.

Today's commonly used Bloom filter-based solutions fail to meet the first three requirements, while non-record linkage based MPC private set intersection algorithms cannot provide the probabilistic comparison required for sufficient precision on noisy real-world datasets. We approach these challenges by designing a method for record linkage and record linkage-based patient intersection using MPC.

5.2.1 Match Classification

For its local data de-duplication, Mainzelliste uses a record linkage algorithm, which is inspired by the EpiLink software[33] and resembles a threshold-based similarity join[34]. To achieve the best compatibility within the German medical research ecosystem, we implemented the same algorithm for MainSEL.

[33] Contiero et al. (2005)
[34] Cohen (2000)

The EpiLink algorithm subdivides the record similarity calculation into two parts. First, the similarity of each field is calculated. Subsequently, the similarity score $S(x, y)$ for the two records x and y is the normalized weighted sum of the individual field similarities. Both field similarities and record similarity scores lie between 0 and 1:

$$S(x,y) := \overbrace{\sum_{i \in I} \delta_{i,i} w_i \operatorname{sim}_i(x_i, y_i)}^{s(x,y):=} \Big/ \overbrace{\sum_{i \in I} \delta_{i,i} w_i}^{w(x,y):=} . \tag{5.3}$$

Both records x and y have $n = |I|$ field values x_i and y_i, each, for $i \in I$, where I is the field index set. $\delta_{i,j}$ is 1 if both fields x_i and y_j are non-empty and 0 otherwise. The field similarity of fields i are calculated using the functions sim_i, which will be described in Section 5.2.2. Following Contiero et al.[35], the weights are chosen using the error rate e_i and average frequency of values f_i, according to the formula $w_i = \log((1 - e_i)/f_i)$. Those values are statistically derived once for a (gold standard) set of fields and then fixed. The values used in this work are listed in Appendix E.2.

[35] Contiero et al. (2005)

The ability to weight field similarities allows researchers to reflect the state of data quality in the used datasets and enable flexible adaptation to specific use cases. Based on user-configurable thresholds, the records are determined to match or tentatively match according to their similarity score.

Equation (5.3) introduces the definitions $s(x, y)$ and $w(x, y)$ for the numerator and denominator—called the *field-weight* and *weight component* of a (partial) score–because often individual processing is required, especially when describing the MPC solution. As divisions are computationally expensive—even more so in MPC—the actual division $S = s/w$ is never evaluated.

Tie-solving order

We often need to determine the maximum of a set of quotients. This seemingly trivial operation is complicated by the possible presence of empty fields. For example, consider a record with only one field non-empty: $x_{\text{first name}} =$ "John". This record x would match *perfectly*—that is $S(x, y_i) = 1$—with all records having the same first name, regardless of the other fields. To allow a more sensible comparison, we introduce a special order. On quotients $S_1 = s_1/w_1$ and $S_2 = s_2/w_2$—written as numerator-denominator pairs (s_1, w_1) and (s_2, w_2)—we define the *tie-solving order* as

$$(s_1, w_1) > (s_2, w_2) :\Leftrightarrow (s_1 w_2 > s_2 w_1) \vee (s_1 w_2 = s_2 w_1 \wedge w_1 > w_2), \tag{5.4}$$

which returns true even if the numerical values of the quotients are the same, but numerator and denominator of the "left" quotient are nominally larger. In this case, more entries contributed to the "left" quotient's score—the "right" contained more empty fields. It also solves the problem of zero denominators, favoring the quotient with non-zero denominator in such a case. If both numerator and denominators are zero, the ordering is irrelevant, as the contribution to the final sum would be zero anyway.

Exchange groups

In real-world record linkage applications, data entry might introduce additional errors. One class of error is the accidental swapping of similar fields, like *first, sur-* and *birth name*. Additionally, fields might change legitimately, e.g., last- and birth name due to marriages. The linkage quality can be improved by grouping some fields into so-called *exchange groups*—like the mentioned name fields.

As described above, in the graph view this is dealt with by fully connected subgraphs—K_3 in the case of the three name fields—topologically excluding the ordering for those vertices. In the database view this can be replicated by the pairwise comparison of all field combinations in the similarity score calculation (5.3).

All the permutations of a set of fields G are included in the *symmetric group* $\mathrm{Sym}(G)$, which has $|\mathrm{Sym}(G)| = |G|!$ entries. Although all fields in G must be of the same type, e.g., numerical or strings, they can have different weights. Hence, we define the similarity score elements for an exchange group $G \subset I$ and permutation $\sigma \in \mathrm{Sym}(G)$ as:

$$\begin{aligned} s_G^\sigma &:= \sum_{i \in G} \delta_{i,\sigma(i)} w_{i,\sigma(i)}\, \mathrm{sim}_i(x_i, y_{\sigma(i)}), \\ w_G^\sigma &:= \sum_{i \in G} \delta_{i,\sigma(i)} w_{i,\sigma(i)}, \\ w_{i,\sigma(i)} &:= \frac{w_i + w_j}{2}. \end{aligned} \tag{5.5}$$

The group's sub-score for permutation σ reads now $S_G^\sigma(x,y) := s_G^\sigma / w_G^\sigma$. Using the identity as a permutation and all fields results, of course, in the similarity score without consideration of exchange groups: $S = S_I^{\text{id}}$. The contribution of the exchange group to the final similarity score is the score of the best-matching permutation—the maximum of all sub-scores. As the described intricacy of comparisons with possible empty fields are now present in even more individual comparisons, the application of the tie-solving order (5.4) is crucial:

$$S_G(x,y) = (s_G, w_G) := \max_{\sigma \in \text{Sym}(G)} (s_G^\sigma, w_G^\sigma). \tag{5.6}$$

To formally include all exchange groups score calculation into the similarity score equation, we define $\mathcal{E}$ as the set of all exchange groups and $\tilde{I} := I \setminus \cup_{G\in\mathcal{E}} G$ as he set of all fields not in any exchange group. The similarity score of two records x and y now becomes the combination of both sets' contributions:

$$S(x,y) = s(x,y)/w(x,y) = \Big(\sum_{G\in\mathcal{E}} s_G + s_{\tilde{I}}^{\text{id}}\Big) \Big/ \Big(\sum_{G\in\mathcal{E}} w_G + w_{\tilde{I}}^{\text{id}}\Big), \tag{5.7}$$

where s_G and w_G are the numerator and denominator of the group scores S_G, as defined in eq. (5.6).

5.2.2 Field Similarity

For some field types a simple equality test is sufficient as a similarity measure—yielding 1 if both compared fields are identical and 0 otherwise. To allow probabilistic, "fuzzy" matching, however, more intricate field comparisons are required. As many PPRL solutions (e.g., Lablans, E. Schmidt, and Ückert [36] and Heidt, Hund, and Fegeler [37]), we use Dice-coefficients on Bloom filters (introduced below) as an appropriate, nevertheless quickly computable similarity measure. Contrary to those existing solutions, however, we compare Bloom filters of complete fields—as opposed to Bloom filters filled with fragments of different fields—and only use them as data structures, *not* relying on them as a security mechanism[38].

[36] Lablans, E. Schmidt, and Ückert (2018)

[37] Heidt, Hund, and Fegeler (2021)

[38] In fact, Christen et al. (2017) show weaknesses in many Bloom filter-based data privacy mechanisms. Our privacy guarantees are provided by the utilized MPC protocols.

Bloom filter

The usual application of Bloom filters is to efficiently index a dataset with large numbers of records, such that is very quickly decidable, whether one element x is part of that set. The reverse—whether the element is *not* included in the set—is only answered probabilistically based on the chosen internal structure of the Bloom filter. This means, that using Bloom filters in that fashion *false positives* may occur. Bloom filters are applied in many applications, such as (computer) virus detection[39], advanced data structures like log-structured merge trees[40], or as a component in genetic optimizations of computationally hard problems such as the search for ground-states of Ising spin glasses[41].

[39] Erdogan and Cao (2007)

[40] O'Neil et al. (1996)

[41] Worring, Mayer, and Hamacher (2021)

Structurally Bloom filters are bit vectors of length m with elements $B_0, B_1, \ldots, B_{m-1}$, initially all set to 0. When inserting an element x into the filter, this element gets hashed by k independent hash functions H_i, modulo the filter length: $x_i = \mathsf{H}_i(x) \mod m$. Instead of using a full set of independent hash function, those can be constructed using only two independent

hash functions: $\mathsf{H}_i(x) = \mathsf{H}_0(x) + i \cdot \mathsf{H}_1(x), \quad i \in \{0, 1, \ldots, k-1\}$ following KIRSCH AND MITZENMACHER [42]. In any case, x gets associated with k values $x_0, \ldots, x_{k-1} \in \{0, 1, \ldots, m-1\}$—*indices* of bits in the Bloom filter. Those bits are set to one.

[42] Kirsch and Mitzenmacher (2006)

The false positive rate for a Bloom filter of length m using k hash functions and including n elements is often given as

$$p^k = (1 - (1 - 1/m)^{kn})^k,$$

which is incorrect, albeit a reasonable approximation for many cases[43].

[43] The work of Bose et al. (2008) examines this question in great detail and gives upper and lower bounds for the false positive rate of Bloom filters.

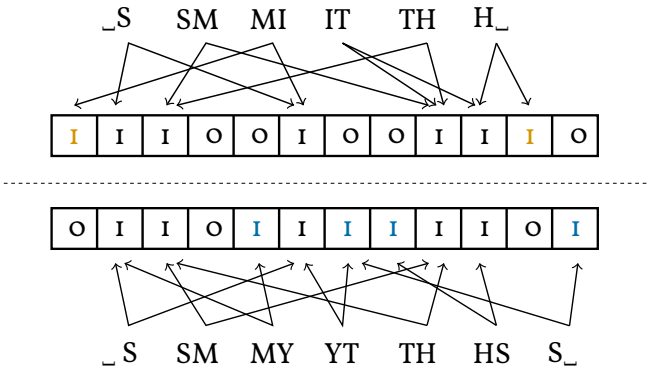


Figure 5.1: Visual example of a Bloom filter-based Dice similarity measurement between the strings "SMITH" and "SMYTHS". Differences in the set bits are colored. This example assumes $k = 2$ independent hash functions and a 12 bit Bloom filter. Note that a change of one letter leads to at most $2k$ changes in the Bloom filter. This means that small changes in the strings lead to small changes in the bit vector.

Sørensen-Dice similarity

THE BLOOM FILTER Dice similarity—introduced to record linkage research by SCHNELL, BACHTELER, AND REIHER [44]—are used to compare string fields like *first-* and *surname* on a more gradual scale than the equality test. The conversion of a string x into a Bloom filter $\mathrm{Bl}(x)$ is not performed by directly hashing the characters of the string, but by first tokenizing the string into n-grams—groupings of n characters, usually $n = 2$—and inserting *those* n-grams into the Bloom filter, thereby setting the corresponding bits in the Bloom filter bitmask.

[44] Schnell, Bachteler, and Reiher (2009)

A useful function to compare bit vectors is the *Hamming weight*—the number of bits set in a bit vector, denoted by Hw. As useful abbreviations, let $X \wedge Y$ denote bitwise AND of the bit vectors X and Y, $H_x := \mathrm{Hw}(\mathrm{Bl}(x))$ the Hamming weight of the Bloom filter of string x, and $H_{x \wedge y} := \mathrm{Hw}(\mathrm{Bl}(x) \wedge \mathrm{Bl}(y))$. The *Sørensen-Dice*-coefficient[45,46], the similarity of two strings is now calculated as

$$\mathrm{sim}_{\mathrm{string}}(x, y) = \frac{2 \cdot \mathrm{Hw}(\mathrm{Bl}(x) \wedge \mathrm{Bl}(y))}{\mathrm{Hw}(\mathrm{Bl}(x)) + \mathrm{Hw}(\mathrm{Bl}(y))} = \frac{2 \cdot H_{x \wedge y}}{H_x + H_y}. \tag{5.8}$$

[45] Dice (1945)

[46] Interestingly, the Dice-coefficient was originally developed to describe set associations in the field of theoretical ecology.

Often only called Dice-coefficient, it has the advantage of being insensitive to the number of zero bits. That means, that the used Bloom filter size can be increased to reduce the false positive rate, while yielding consistent values. As the example in Figure 5.1 demonstrates, it captures the *relative* similarity of strings. Small changes in the string lead to small numbers of bigrams changed, lead to a small difference in bits set. Note that the Dice-coefficient could also have been applied directly to the bigrams of two strings, however Bloom filters constitute a data structure which can be manipulated efficiently in a MPC context.

5.2.3 Mainzelliste Identity Management and Pseudonymization Framework

Mainzelliste[47] is a web-based identity management and pseudonymization framework. It is actively employed for record linkage within a multitude[48] of German and European medical research infrastructures[49], biobanks[50] and patient registries[51].

Mainzelliste is able to manage primary and secondary pseudonyms along IDAT and it can operate as a master patient ID generator. Its local probabilistic record linkage module for patient de-duplication uses a highly optimized[52] version of the EpiLink[53] algorithm. MainSEL extends Mainzelliste's record capabilities by providing adapters between the RESTful APIs of both Mainzelliste and SEL, thus forming a pseudonymization, record linkage and ID management system fit for MPC-based trans-institutional PPRL. MainSEL, including the algorithms and software described in this work, are freely available as open source software under https://github.com/medicalinformatics/MainSEL.

[47] Lablans, Borg, and Ückert (2015)

[48] At the time of writing, around 20 projects. See https://bitbucket.org/medicalinformatics/mainzelliste/src/development/README.md (accessed 12.03.2022) for details.

[49] Joos et al. (2019); Medizin et al. (2019); Prokosch et al. (2018)

[50] Bernemann et al. (2016)

[51] Kronfeld et al. (2021)

[52] Rohde et al. (2021)

[53] Contiero et al. (2005)

5.3 Circuit Design

For the MPC implementation, the main functionalities (5.1) and (5.2) must be translated in Boolean and arithmetic circuits[54]. As MPC circuits may not have dynamic control flow—all branches are evaluated and all loops unrolled—the implementation is carefully optimized. For example, *Single Instruction Multiple Data* (SIMD) vectorization and parallelization is heavily employed, and all sums are constructed as balanced-binary trees to minimize the circuit depths. Furthermore, the computational representation of real values plays an important role: although floating-point calculations are supported in both MPC in general and in the employed MPC framework—ABY[55]—specifically[56], their usage is computationally expensive. This work uses fixed-point representation of decimal values in a performance–precision trade-off.

[54] See Section 2.3

[55] Demmler, Schneider, and Zohner (2015)

[56] Demmler, Dessouky, et al. (2015)

Unfortunately, due to the lack of dynamic control flow the possible usage of *blocking* mechanisms—the pre-filtering of records to reduce the amount of comparisons needed—is an ongoing field of research. X. He et al.[57] show, that common blocking techniques using LSH are incompatible with MPC's security guarantees. As a result, blocking is not considered in this work.

[57] X. He et al. (2017)

In the following sections the circuit designs for similarity score calculation (5.7) and the classification of bestMatch (5.1) are explained in more detail.

Given a record x by Alice and the records $\{y^j\}$ by Bob, the task of the "high-level" circuits (C1) to (C4) are:

C1. calculates all scores' numerators $s(x, y^j)$ and denominators $w(x, y^j)$,

C2. determines the highest score and its index $j^* := \arg\max_j S(x, y^j)$,

C3. tests for a match by calculating the *match bit*

$$m = \begin{cases} 1, & \text{if } S(x, y^{j^*}) = s(x, y^{j^*})/w(x, y^{j^*}) > T \\ & \qquad \iff s(x, y^{j^*}) > Tw(x, y^{j^*}), \\ 0, & \text{otherwise,} \end{cases}$$

C4. calculates the (optional) set intersection cardinality by summing all match bits.

Due to the computational cost of (integer) divisions, (C2) and (C3) are carefully designed, such that the field-weight- and weight sums s and w—calculated in parallel—are only required as components. The actual division $S = s/w$ is never performed. The sequential execution of (C1) to (C3) implement the bestMatch$(x^k, \{y^j\})$. This result can be used for the matchCardinality$(\{x^k\}, \{y^j\})$ functionality in (C4).

5.3.1 Notation

To differentiate between different domains, additional (typographical) notation is used in the following sections. For individual circuit component $\mathsf{x} = \mathcal{C}(x)$ x is used to denote that x is the encoding of value x. Sans-serif font is used for circuit variables, typewriter for circuit functions and algorithms.

To abbreviate bit-length considerations, we define $\mathsf{bitlen}(x) := \mathsf{bitlen}(\mathcal{C}(x)) := \mathsf{bitlen}(\mathsf{x}) := l$, for $\mathsf{x} \in \{0,1\}^l$ or $\mathsf{x} : * \to \{0,1\}^l$.

The three used MPC protocols Arithmetic GMW, Boolean GMW, Yao's Garbled Circuits are abbreviated $\mathcal{A}, \mathcal{B}$ and $\mathcal{Y}$[58], respectively. The spaces of bit-length values l in these protocols are written as $\mathbb{S}^l_{\mathcal{A}}, \mathbb{S}^l_{\mathcal{B}}$ and $\mathbb{S}^l_{\mathcal{Y}}$.

[58] The same abbreviation is used in the ABY publication Demmler, Schneider, and Zohner (2015).

The annotation $\langle x \rangle^l_p$ for a variable's or function's output bit-length l in the protocols $p \in \{\mathcal{A}, \mathcal{B}, \mathcal{Y}\}$ is mostly restricted to the section discussing bit-length and precision considerations. Where unambiguously determinable from the context, the l superscript or p subscript is omitted for brevity.

5.3.2 Fixed-Point Representation

Weights, thresholds, and field similarities are the only occurring real values in the calculation. As discussed in Section 5.3, those real values must be represented in fixed-point representation, that is a specific number of bits in the value is used for the integral part of the value and a specific number of bits is used for the fractional part. The number of bits used for each part determines the achievable precision and range of number representation. The *weight precision* is written as $\mathsf{lw} := \mathsf{bitlen}(w_i)$ and the *similarity* or *Dice precision*, which is the same for all fields $i \in I$ as $\mathsf{ls} := \mathsf{bitlen}(\mathsf{sim}_i)$.

As the output of the field similarity measures sim_i are real numbers between (and including) 0 and 1, their fixed-point representations are calculated as $\mathcal{C}(\text{sim}_i) = \left\lfloor \text{sim}_i \cdot 2^{\mathsf{ls}} \right\rceil$. In case of equality, which outputs either 0 or 1, the rounding can be foregone. Their transformation is just a left-shift by ls. The circuit implementation of the Bloom filter Dice-coefficient $\text{sim}_{\text{string}}$ requires the only evaluation of an (integer-) division. To reduce the performance impact, we use a custom integer-division where the numerator is left-shifted by ls before the integer-division. The operation yields a result between 0 and 2^{ls}.

The scaling of the real-valued thresholds T depends on the field-similarity scaling, as they are compared in inequality (C3). To attain the fixed-point representation for T, it is multiplied with 2^{ls} and subsequently rounded to the nearest integer: $\mathsf{T} = \mathcal{C}(T) = \left\lfloor T \cdot 2^{\mathsf{ls}} \right\rceil$.

Lastly, the real weights $w_i > 0$ are transformed into numbers $\mathsf{w}_i = \mathcal{C}(w_i) \in \{0,1\}^{\mathsf{lw}}$ by rescaling them to use the full available bit range. That means normalizing so that the highest weight has value $2^{\mathsf{lw}} - 1$ and then rounding to the nearest integer:

$$\mathsf{w}_i := \left\lfloor \frac{w_i}{w_{\max}} (2^{\mathsf{lw}} - 1) \right\rceil, \quad w_{\max} := \max_{i \in I} w_i. \tag{5.9}$$

Using this construction, the highest possible precision is achieved because the full range of $\{0,1\}^{\mathsf{lw}}$ is used for the occurring weights.

5.3.3 Circuit Implementation Variants

One of the key contributions of Henecka et al.[59] and Demmler, Schneider, and Zohner [60], apart from the software implementations, was the key insight that performing subsequent operations in different MPC protocols might be more efficient than staying in the same protocol—even when considering the conversion costs. This insight can be translated to this work, as some parts perform more logic operations—that is Boolean operations—and some more arithmetic operations. As the optimal efficiency depends on more factors, such as network bandwidth and latency, we left the choice to the user, enabling them to select protocols for the Boolean parts (protocol β) and arithmetic parts (protocol α). The possible conversion functions are denoted with `a2b` and `b2a`. These functions operate as identity-functions if the same protocol is chosen for α and β.

[59] Henecka et al. (2010)

[60] Demmler, Schneider, and Zohner (2015)

The Boolean sections may be performed in either *Yao's Garbled Circuits* (GC) or Boolean GMW, i.e., $\beta \in \{\mathcal{B}, \mathcal{Y}\}$. The arithmetic components may be performed in the chosen Boolean protocol or in arithmetic GMW, i.e., $\alpha \in \{\mathcal{A}, \beta\}$. This results in four circuit variants:

GMW: $\beta = \alpha = \mathcal{B}$, i.e., the whole circuit implemented in the Boolean GMW protocol.

GMW/A: $\beta = \mathcal{B}$ and $\alpha = \mathcal{A}$, i.e., Boolean/logic components implemented in the Boolean GMW protocol and arithmetic components in Arithmetic Sharing.

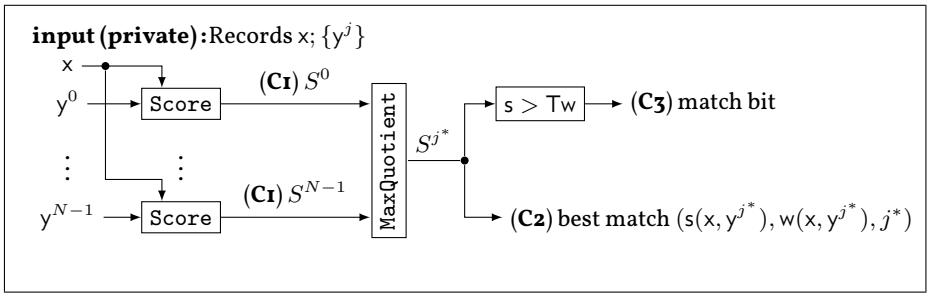
Yao: $\beta = \alpha = \mathcal{Y}$, i.e., the whole circuit implemented in Yao's Garbled Circuit.

Yao/A: $\beta = \mathcal{Y}$ and $\alpha = \mathcal{A}$, i.e., Boolean/logic components implemented in Yao's Garbled Circuit and arithmetic components in Arithmetic Sharing.

Specifically, Circuits 5.2 and 5.3 are of arithmetic nature while Circuits 5.4 and 5.5 are of Boolean nature. Circuit 5.6 is of mixed nature: after two multiplications, several Boolean operations are performed.

5.3.4 Circuit Components

Now each circuit implementation required to attain the main functionalities (C1)–(C3) are described. The inputs for each component are only mentioned for the component using it. They are omitted in "parent" components. A high-level overview of the composition of circuits is shown in Circuit 5.1



Circuit 5.1: High-level circuit calculating (C1)–(C3), consequently implementing functionality bestMatch. The scores $S^j := S(x, y^j)$—the results of (C1)—are calculated by evaluating Circuit 5.2 (Score) for all record input pairs x and y^j from Alice and Bob in parallel. The best match (C2) is then determined by running circuit MaxQuotient on all scores, which, for performance reasons, is a balanced binary-tree fold of Circuit 5.6. Finally, the match bit (C3) is determined by evaluating the threshold comparison(s) $s(x, y^{j^*}) > Tw(x, y^{j^*})$ on the best match.

The circuit implementation of Score is composed of subcomponents, operating on a single pair of records $x = \{x_i\}_{i\in I} = \mathcal{C}(x)$ and $y = \{y_i\}_{i\in I} = \mathcal{C}(y)$ provided (privately) by Alice and Bob. To improve readability, the index j designating the record of Bob's input y^j is omitted. z_i denotes an *individual field* of a single record z.

Circuit 5.2 (Score) calculates the score numerators $s = \mathcal{C}(s)$ and denominators $w = \mathcal{C}(w)$ concurrently, using protocol α (cf. eq. (5.7)). Subcomponents GroupFieldWeight (Circuit 5.3) and MaxQuotient (Circuit 5.6) are used for exchange group score evaluation. The circuits calculate a group's sub-score and find the maximum value of all group sub-scores (cf. eq. (5.6)), respectively.

Circuit 5.2: Score – similarity score (C1) of input records x and y (eq. (5.7)), evaluated in protocol α.

input (public) : field indices I, exchange groups $\mathcal{E}$
output (shared) : sum of field-weights $s(x, y)$, sum of weights $w(x, y)$

1 **foreach** $G \in \mathcal{E}$ **do**
2 **foreach** $\sigma \in \mathrm{Sym}(G)$ **do**
3 $s^\sigma_G, w^\sigma_G \leftarrow$ GroupFieldWeight(G, σ);
4 $s_G, w_G, \_ \leftarrow$ MaxQuotient$((s^\sigma_G, w^\sigma_G)_{\sigma\in\mathrm{Sym}(G)})$;
5 $s_{\tilde{I}}, w_{\tilde{I}} \leftarrow$ GroupFieldWeight$(\tilde{I}, \mathrm{id})$;
6 $s(x, y) \leftarrow s_{\tilde{I}} + \sum_{G\in\mathcal{E}} s_G$; $w(x, y) \leftarrow w_{\tilde{I}} + \sum_{G\in\mathcal{E}} w_G$;

Similarity Circuits

The field similarity sim applies the type dependent comparison function–either the simple equality Circuit 5.4 or the probabilistic Bloom filter Dice-coefficient Circuit 5.5—on field entries $x_i, y_{\sigma(i)}$. If field i has Dice similarity type the (locally) pre-computed field entry's Bloom filter is expected as the circuit's input: $x_i = \mathcal{C}(x_i) = \mathrm{Bl}(x_i)$. The bit-length of field i is denoted by lb_i.

input (public) : group $G \subset I$, permutation $\sigma \in \mathrm{Sym}(G)$, weights $\mathsf{w}_* \in \{0,1\}^{\mathsf{lw}}$
input (private) : empty-field bits $\delta_i^x; \delta_i^y$
output (shared) : sum of field-weights $\mathsf{s}_G^\sigma(\mathsf{x},\mathsf{y})$, sum of weights $\mathsf{w}_G^\sigma(\mathsf{x},\mathsf{y})$

1 **foreach** $i \in G$ **do**
2 $\quad \mathsf{w}'_i \leftarrow \delta_i^x \cdot \delta_{\sigma(i)}^y \cdot \mathsf{w}_{i,\sigma(i)}$;
3 $\quad \mathsf{s}_i \leftarrow \mathsf{w}_i \cdot \mathtt{b2a}(\mathtt{sim}(i, \sigma(i)))$;
4 $\mathsf{s}_G^\sigma \leftarrow \sum_{i \in G} \mathsf{s}_i$; $\mathsf{w}_G^\sigma \leftarrow \sum_{i \in G} \mathsf{w}'_i$;

Circuit 5.3: `GroupFieldWeight` – evaluates eq. (5.5) in protocol α. The empty-field bits δ_i^z are 0 if entry i of record z is empty and 1 otherwise. If one of the entries is empty, then $\mathsf{w}'_i = \mathsf{s}_i = 0$. Note that if $\alpha = B$ or Y, the multiplication between the (single bit) δ's in line 2 is equivalent to and implemented as a logical AND.

input (private) : Field values x; y
return $(\mathsf{x} == \mathsf{y}) \ll \mathsf{ls}$

Circuit 5.4: `Equal` – equality field comparison, evaluated in protocol β.

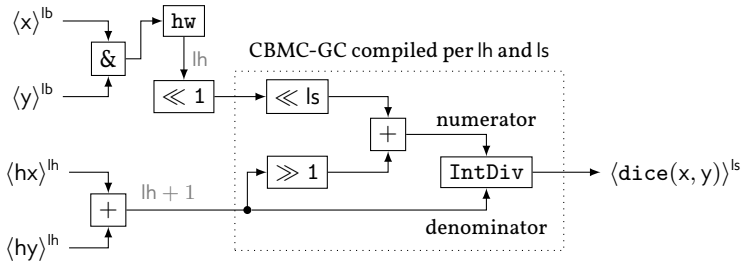
Both circuits output the similarity as values in protocol β with identical bit-lengths (i.e., fixed-point precision) ls, implementing all multiplications and divisions by 2 as locally computable—free—bit-shifts. The components in `dice` marked by a dashed box were implemented using the CBMC-GC-2 compiler[61] which exhaustively optimizes the resulting circuit. The function compiled to a Boolean circuit is $x, y \mapsto ((x \ll \mathsf{ls} + y/2)/y)$, the rounding integer division. Both "/" and `IntDiv` denote the `C` integer division. For all sensible input and output bit-lengths $2 \leq \mathsf{lh} + 1 \leq 12$ and $2 \leq \mathsf{ls} \leq 22 = \lceil 64/3 \rceil$ a circuit optimized for the specific parameter combination was compiled—covering Bloom filters of up to 2,047 bit length. The Hamming weight of bit vectors—Bloom filters in our case—use $\mathsf{lh} = \lceil \log(\mathsf{lb} + 1) \rceil$ bits, as this number of bits is sufficient to represent the maximum value resulting from a sum of lb bits.

[61] Franz et al. (2014); Buescher et al. (2016)

Maximum Quotient Circuit

As previously defined, a group's sub-score is the quotient $\mathsf{s}_G^\sigma / \mathsf{w}_G^\sigma$. The group's *score* then is the maximum of all sub-scores (cf. eq. (5.6)). This maximum fold uses the tie-solving order as defined in eq. (5.4). The implementation of the "building block" for the `MaxQuotient` operation, which outputs not only the larger of two quotients following the tie-solving order but its index as well is shown in Circuit 5.6.

input (private) : Bloom filters x; y and their Hamming-weights $\mathsf{hx} = \mathrm{Hw}(\mathsf{x}); \mathsf{hy} = \mathrm{Hw}(\mathsf{y})$



Circuit 5.5: `Dice` – schematic of the dice similarity field comparison (eq. (5.8)), evaluated in protocol β. The annotations display the bit-length.

Circuit 5.6: `MaxQuotient′` – maximum of two quotients with index, performed in mixed protocols. The `Mux` operation `Mux(c, a, b)` returns a if c is 1 and b otherwise. Note, that this circuit does not implement the full `MaxQuotient` functionality. For that it is chained in a tree structure.

input (shared): Quotients $(s_i, w_i, i), (s_j, w_j, j)$
1 $z_i, z_j \leftarrow \mathsf{a2b}(s_i \cdot w_j), \mathsf{a2b}(s_j \cdot w_i)$;
2 $c \leftarrow (z_i > z_j) \vee (z_i = z_j \wedge \mathsf{a2b}(w_i) > \mathsf{a2b}(w_j))$;
3 **return** $\mathsf{Mux}(c, s_i, s_j), \mathsf{Mux}(c, w_i, w_j), \mathsf{Mux}(c, i, j)$

The index is not necessary for the calculation of a group weight, however, the later bestMatch circuit (C2) uses it to determine the best matching record in Circuit 5.1.

The actual `MaxQuotient` circuit constructs a binary-tree fold of a list of quotients using the building block `MaxQuotient′` as the fold operation.

5.3.5 Precision Parameter Choices

The conscious choice of the bit-length L for arithmetic circuit components, resulting in the fixed-point precisions l_w and l_s, is important to avoid overflows while achieving maximum precision.

The weight sum $w(x, y)$ yielded by Circuit 5.2 is a sum of n weights of bit-length l_w. As such, its maximum value can be represented with $\lceil \log(n) \rceil + l_w$ bits. Likewise, $s(x, y)$ has length $\lceil \log(n) \rceil + l_w + l_s$. However, the largest values occurring in any arithmetic circuit component result from a multiplication—both z_i's in Circuit 5.6, line 1 are the product of a s and a w. In this expression a sum of n weights of length l_w are multiplied with a sum of n field-weights of length $l_w + l_s$. Hence, the length of both z_i's is $\lceil \log(n^2) \rceil + 2l_w + l_s$—the length not allowed to overflow the used data types[62]. When l_w and l_s are chosen to fully use the bit-length L of space $\mathbb{S}_\alpha^L$ while avoiding overflows, $r := L - \lceil \log(n^2) \rceil$ bits remain unused. These are evenly divided between both bit-lengths by setting $l_w = \lceil r/3 \rceil, l_s = \lfloor r/3 \rfloor$ if $r \bmod 3 = 2$ and $l_w = \lfloor r/3 \rfloor, l_s = \lceil r/3 \rceil$ otherwise.

[62] ABY supports $L \in \{8, 16, 32, 64\}$ if arithmetic GMW is used for the arithmetic circuit components, that is $\alpha = \mathcal{A}$.

Our comparison between the similarity score calculation performed using the described fixed-point representation and the same calculation using double precision floating point values[63] yield only small deviations: Using large random numbers as inputs, the observed introduced errors are $< 1\%$ for $L = 16$ bit, $< 0.1\%$ for $L = 32$ bit and negligible for $L = 64$ bit. Most benchmarks in Section 5.5.3 were performed with $L = 32$ and $n = 8$ fields. This results in $l_w = 9$ and $l_s = 8$.

[63] IEEE Standards Board and American National Standards Institute (1985)

5.4 Systems Architecture

The complete MainSEL record linkage system is composed of multiple software packages—Figure 5.2 shows an overview over the architecture. Core elements are Mainzelliste as the data source and management unit and SEL as the MPC compute unit, communicating via JSON REST interfaces. Apart from the relational database used by Mainzelliste, all additional components are required to allow the deployment in restricted network environments, such as hospital data integration centers. The following sections—Sections 5.4.1 and 5.4.2—describe the coupling between Mainzelliste and SEL, as well as the record linkage and ID management process. The networking and deployment considerations are discussed in Section 5.4.4.

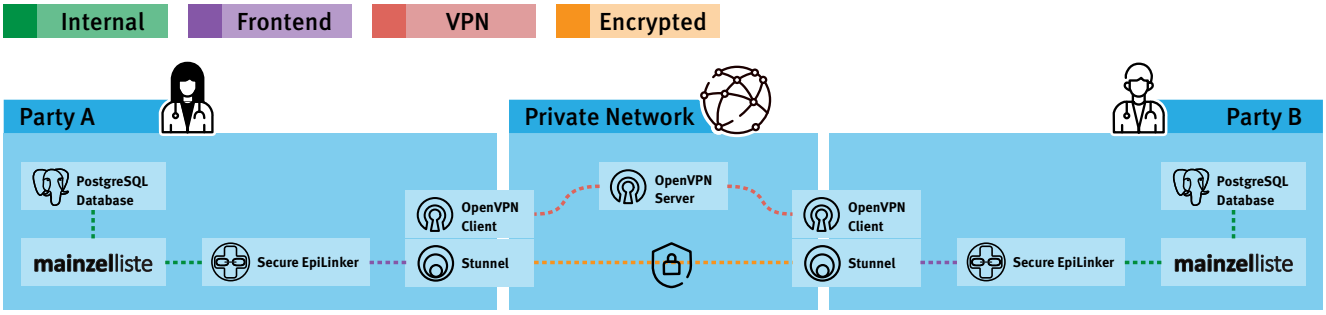


Figure 5.2: MainSEL architectural overview. The diagram shows two MainSEL Docker Compose stacks, both interacting in a virtual, private network established by a OpenVPN server. The MainSEL core components use stack-internal networking. Only the Stunnel and OpenVPN container are exposed to the outside network. Note, that for the ID management process a Linkage Service component may be used, which is not displayed here. This figure was created by the author and used in Kussel et al. (2022), licensed under CC-BY.

5.4.1 Communication

The SEL component is designed for maximum flexibility, hence it starts with only a minimal configuration necessary for establishing the REST endpoints, e.g., listening port(s), network interface to bind to, certificates for HTTPS communication. The remaining "business-logic" configuration—what fields to compare with which comparison function, weights, connection details for the data source, connection details for the remote computation party, and so on—are configured during runtime via the REST interface. Thus, two general phases can be discerned: an initialization phase and the linkage/matching phase. An arbitrary number of remote computation parties can be configured. In the current releases, all parties are authenticated using pre-shared keys. The communication is executed using a secure channel, e.g., TLS-secured[64].

The test of the connections between the local SEL and the (multiple) remote SELs, as well as the linkage service, mark the completion of the initialization phase. In the tests, not only network connectivity is assured, but the compatibility of the algorithm configuration—that is the same fields with the same weights are compared.

The linkage and the matching phase are similar and differ only in the used REST resources and the result—one returning a linkage ID and the other the set intersection cardinality. Both varieties are displayed in Figures 5.3 and 5.4. Note, that steps (1) to (3) are identical.

The linkage/matching phase starts with the local Mainzelliste initiating a linkage or matching task by sending one or more records to the local SEL, as well as a callback address to receive the result (step (1)). Some properties are considered public knowledge, as they are required for efficient circuit creation. Local and remote SEL exchange the numbers of records on each side (step (2)). During this exchange, the remote SEL queries all records from "its" Mainzelliste. As the construction of the circuit and the execution might be performed at different times[65], the number of records transferred could be based on estimates or padded to allow for database growth between circuit generation and execution.

[64] Rescorla (2008)

[65] See Section 2.3

After these prerequisites are satisfied, the actual MPC is performed between the local and the remote SEL (step(3)). At the end of the computation, the execution of linkage and matching start to differ.

In the linkage process (cf. Figure 5.3), both parties hold one share of the index of the best matching (remote) record for each local record, as well as shares of all match bits. These shares are sent to the linkage service, which reconstructs the clear text values (step (4)). Additionally, the remote SEL transmits the encrypted IDs for its records to the linkage service. Note, that the linkage service is *not* a TTP and the IDs are specifically constructed to assure confidentiality. The details are discussed in Section 5.4.2.

The linkage service de- and re-encrypts the best matching IDs with the match bit appended. This *Linkage ID* (LID) is in all outcomes—match, tentative match, and non-match—indistinguishable from a random string. It is transmitted to the local SEL which, in turn, sends it to the pre-configured callback address (step (5)).

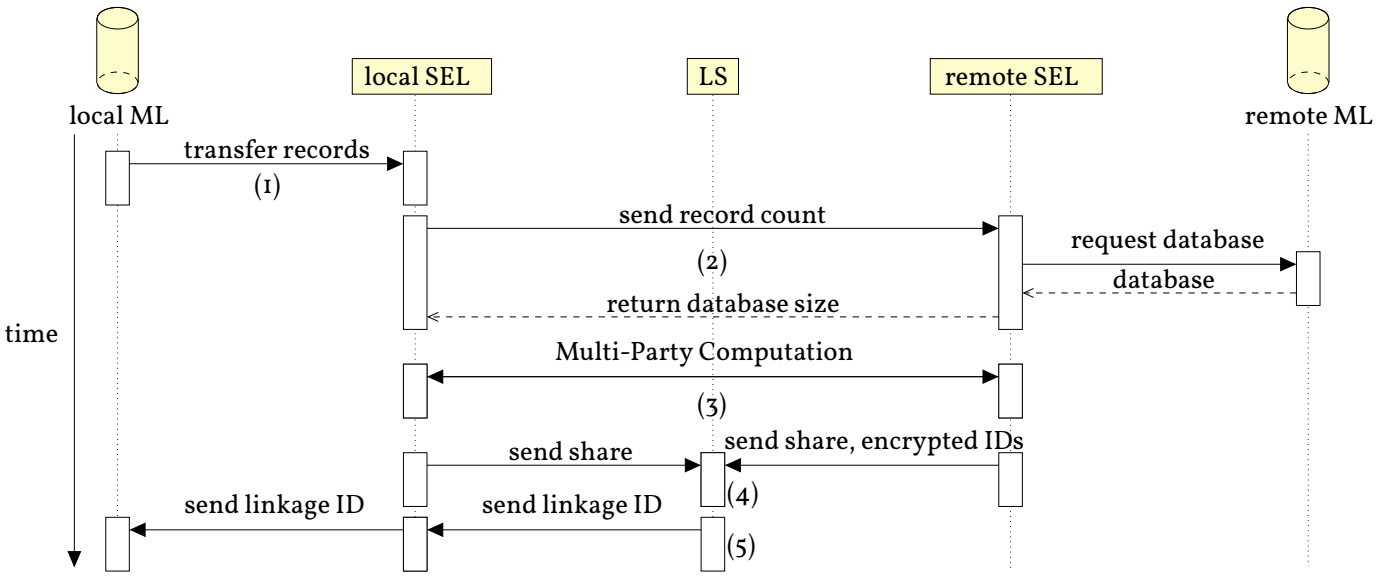


Figure 5.3: Communication sequence diagram of the linkage phase. *ML* stands for Mainzelliste, the patient database and pseudonymization framework, *SEL* stands for the MPC computation unit and *LS* stands for Linkage Service. The numbers in parentheses enumerate the protocol's steps described in Section 5.4.1. This figure was created by the author and used in Stammler, Kussel, et al. (2020), licensed under CC-BY.

The matching phase (cf. Figure 5.4), calculating the set intersection cardinality, does not require a linkage service. As the MPC circuit sums all match bits, it yields directly the number of common dataset records. Both local and remote SEL send this result to the configured callback addresses (step (6)).

5.4.2 ID Generation and Management

MPC provides strong safety guarantees regarding input privacy, however, the privacy of the *outputs*[66] is not guaranteed. The record linkage output—LIDs—pose, in fact, a re-identification risk if two colluding actors on both sides com-

[66] See Section 2.2

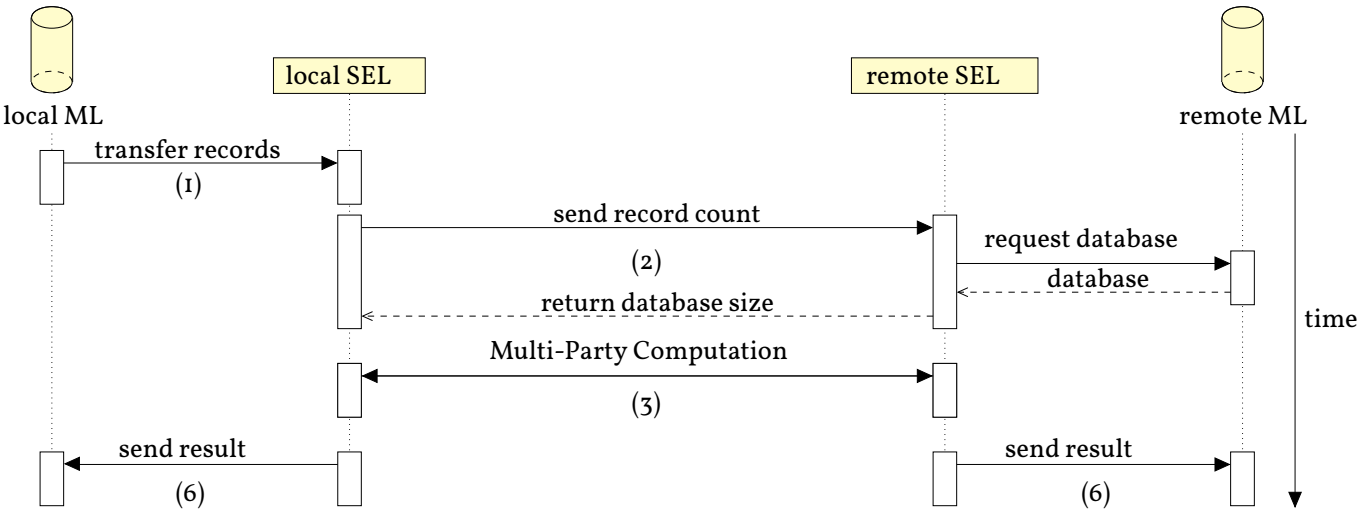


Figure 5.4: Communication sequence diagram of the matching phase. *ML* stands for Mainzelliste, the patient database and pseudonymization framework, *SEL* stands for the MPC compute unit. The numbers in parentheses enumerate the protocol's steps described in Section 5.4.1.

pare those IDs. To mitigate this attack vector, the returned ID must not reveal any information about the matching status, however, this very information is the primary computation objective and required for de-duplicaton and pseudonym assignment.

We achieve confidentiality in the record linkage process, by introducing the *Linkage Service* (LS), a management component concerned with generating, validation and encrypting LIDs. Note, that the LS is *not* a TTP, it does not participate in the record linkage calculation and never receives any private information. It generates random IDs in the setup phase, holds a secret key for each participating party and re-keyes the resulting LIDs and random IDs. As a result of its introduction, the colluding adversaries at both parties can no longer infer the matching status from the LIDs.

An additional benefit of the LS is, that it enables the easy introduction of a LID disclosure policy involving the *Institutional Review Board* (IRB) and *Use and Access Committee* (UAC) by only allowing the decryption of the LIDs after a positive decision of these bodies.

The process steps involving the LS are the following:

1. During setup, the LS generates "raw" LIDs for the data source by drawing random IDs and adding a party specific random but fixed number of zeros. This zero padding enables easy ID validation, later on. These IDs are encrypted with the corresponding party's secret key and transferred to the data source.
2. At the end of a record linkage calculation, the LS receives both secret shares encoding the match bit and the indices of the best matches, as well as the list of all (encrypted) LIDs from the data source.
3. The secret shares are reconstructed and the best matching LIDs are decrypted using the data source's secret key.

4. Comparing the number of appended zeroes with the previously fixed number, the now decrypted IDs are validated.
5. In case of a match, the match bit is appended to the LID. Otherwise, a new random ID is sampled.
6. In both cases, the resulting ID is re-encrypted using the client's secret key and sent to the client.

This procedure ensures, that every ID, regardless of its matching status, is indistinguishable from randomness. Without decrypting the IDs, nothing can be learned from them, even if malicious actors in both parties compare them.

The linkage results can be transformed into a usable result by decrypting them, a procedure that can be restricted—as described before—by an IRB and UAC clearance.

Linkage ID Generation without Linkage Service

While the described record linkage process using a LS for ID management is desirable for easy oversight and policy, the LS is not required from a protocol perspective. Nearly the same functionality could be implemented in the MPC circuit, only the resulting LID would change—both parties would either receive the same LID (match) or a random ID (non-match). This protocol variant could be implemented in the following way:

Both parties input additional per-record randomness into the circuit. If bestMatch results in a match, the randomness of both parties is `XOR`ed to obtain the LID, identical for both parties. If no match is found, each party receives the other party's randomness as an LID. This way, both parties receive seemingly random LIDs as an output. The matches can only be identified after comparing both sets of IDs. In other words, the process is used to generate a trans-institutional master patient identifier.

5.4.3 Record Linkage based Private Set Intersection Cardinality

Using circuit-based MPC protocols provides versatility and easy extensibility to MainSEL. An example is the extension of the bestMatch functionality to allow the calculation of a "fault-tolerant PSI-C"—the number of common patients between the two datasets—by summing up the match bits. This common patient count functionality—matchCardinality—has important real-world applications. In this chapter, we focus on research and study planning applications in the field of rare diseases.

Patients with rare diseases are often registered in many hospitals and other medical facilities, as it is difficult to obtain expert treatment or even diagnosis—leading to not only distributed duplicate entries, but to noisy datasets attesting differing or uncertain diagnosis.

As case numbers at a given institution are most likely too low to conduct statistically significant research, joint cohort studies are a regular study design—a study

design especially biased by duplicate entries when concerned with low sample sizes[67]. The currently required legal process to assess study feasibility—that is cohort sizes—is often comparable to the processes required for full data transfer and often unreasonably complex for this early stage. The privacy guarantees of MPC-based cohort size estimations might reduce those regulatory barriers and lead to faster and more cost-effective rare disease research.

[67] Cheng (1998)

5.4.4 Networking and Deployment

Previous publication[68] describe the complexity and costs involved when carrying out deployments of biomedical applications in clinical networks. This complexity stems mostly from non-standard network topologies, high network compartmentalization and strict regulatory requirements[69].

[68] For example, Lablans, E. Schmidt, and Ückert (2018) and Stammler, Kussel, et al. (2020).

[69] Bundesministerium des Inneren (2007); Bundesministerium des Inneren (2009)

The sensitivity of the data processed inside those networks require high security levels, commonly restricting direct TCP connections using firewall systems and proxies. Ingress network traffic in particular is severely restricted, if not outright filtered out. Computer systems operating in those restricted spaces needing to communicate with outside systems need special design considerations to work as intended. These restrictions unfortunately apply to MainSEL as well.

We solve these challenges by employing an OpenVPN[70] overlay network in the MainSEL system architecture, thus abstracting network connectivity for the core components (cf. Figure 5.2). OpenVPN is a well-known open source VPN solution, highly regarded for its security and simplicity of operation. A full security audit was performed in 2017[71]. Additionally, all outgoing network traffic is routed through "Stunnel"[72]. Stunnel encapsulates all traffic in a valid HTTPS context without needing to modify any of the other components. The introduced (system) complexity is "hidden" from the user by using *Docker Compose*[73] to orchestrate the containerized components..

[70] https://openvpn.net

[71] Raynal et al. (2017)

[72] Wong (2001)

[73] https://github.com/docker/compose

5.5 Benchmarks and Real-World Tests

The experimental assessment of MainSEL includes three parts: first the analysis of record linkage quality. This includes the comparison to Lazrig et al.[74] in terms of match classification error rate. The second set of experiments measure MainSEL's runtime performance in a laboratory environment and lastly, the real-word deployment in eight German university hospital centers is evaluated.

[74] Lazrig et al. (2018)

5.5.1 Record Linkage Quality

The synthetic datasets used in the record linkage quality analysis were generated with the Mockaroo synthetic data generation tool (https://www.mockaroo.com). Mockaroo provides numerous predefined field types, however not all field exists exactly as required. Some post-processing of the raw generated datasets is necessary to adapt them to this analysis, e.g., the Mockaroo field type "Datetime" needs separation into the fields "day", "month" and "year".

This procedure resulted in a dataset with $50{,}000$ records, using the Mockaroo configuration in Table 5.1.

Table 5.1: The Mockaroo configuration used to generate the raw dataset for the record linkage quality analysis. The ZIP code range follows the German ZIP code scheme, with "01001" being the lowest issued code.

Field name	Type	Range
First Name	First Name (European)	
Surname	Last Name	
Birth Name	Last Name	
Birthdate	Datetime	01.01.1930 – 31.12.2019
ZIP code	Number	01001 – 99999
City	City	

The comparison of MainSEL's record linkage quality with Lazrig et al.'s state-of-the-art PPRL implementation[75] needs a different dataset layout, as the one described above is not compatible with their system. We followed the directions given in the publication to generate a second dataset with $50{,}000$ records as well. Table 5.2 shows the second Mockaroo configuration.

[75] Lazrig et al. (2018)

Table 5.2: The Mockaroo configuration used to create the raw dataset for the record linkage quality comparison between MainSEL and the current state-of-the-art (Lazrig et al. (2018)). The configuration follows their dataset structure.

Field name	Type	Range
First Name	First Name (European)	
Surname	Last Name	
Birthdate	Datetime	01.01.1930 – 31.12.2019
SSN	SSN	

Data Selection and Perturbation

To refine the raw dataset for record linkage quality analysis, it must be split in multiple sets with a known overlap. Furthermore, the records must be permuted to benchmark the quality of the probabilistic record linkage algorithms.

For that, first N records are randomly sampled from the complete, raw dataset where N is the total number of unique records. As the field "birth name" is often empty in real-world datasets, 60 % of "birth name" fields are then removed. The described algorithm of (Lazrig et al. (2018)) neither uses a "birth name" field, nor is able to handle empty field. As a result, this step is omitted for the direct comparison between the algorithms. Lastly, the dataset is split into two sets with sizes N_1 and N_2 where the second dataset has an adjustable amount of records also included in the first dataset. To avoid biasing the benchmarks by introducing a fixed structure, the records of the second dataset are randomly shuffled.

To introduce errors, every field in the second dataset is stochastically perturbed. With equal but modifiable probability the following perturbations may be applied:

1. the deletion of a random symbol in the field,
2. the exchange of two random symbols in the field,
3. the field is set to empty.

After this field-wise perturbation, entire fields in a record may be exchanged using the same perturbation probability. This exchange is only applied to compat-

ible fields, e.g., first name and surname, or birth day and birth month. Exchanging, e.g., the city name and the birth year is not permitted, as the fields contain different data types. The application of each modification is independent of the others, i.e., each field or record can be subjected to multiple variations. The exchange between day and month data fields might introduce invalid dates. These fields are sanitized by setting out of bound field values to the nearest valid value.

To be comparable to LAZRIG ET AL.'s testing procedure, the individual variations' probability is chosen such that an adjustable overall field perturbation rate is achieved.

Quality of Record Linkage

THE CHOSEN PARAMETERS used for the generation of the datasets used for the valuation of our EpiLink implementation are the following: Both datasets contain 10,000 records with an overlap of 60 %. The probability of an empty "birth name"-field is set to 60 %. For the varying error rate experiments the probability of each individual perturbation is set to 2.6 % and 5.4 %, corresponding to a total field perturbation probability of 10 % and 20 %, respectively.

The configuration of the EpiLink algorithm is shown in Appendix E.2. A bit precision of $L = 32$ bit was used in all benchmarks. The results are shown in Table 5.3. "TP", "FP" and "FN" denote "True Positives", "False Positives" and "False Negatives", respectively. "Recall" is the True Positive Rate $R = \frac{TP}{TP+FN}$ and "Precision" is $P = \frac{TP}{TP+FP}$. The "F_1-Score" and "Matthews Correlation Coefficient" (MCC) are combined evaluation metrics for binary classification with the following definition:

$$F_1 = \frac{2TP}{2TP+FP+FN}$$
$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Error Rate	TP	FP	FN	Recall	Precision	F_1	MCC
0.1	5967	0	33	0.994	1.0	0.997	0.993
0.2	5820	4	179	0.970	0.982	0.984	0.963

Table 5.3: Record linkage results using MainSEL's EpiLink implementation. The error rate is given per field. Each dataset has 10,000 records with 60 % overlap between sets. "TP", "FP" and "FN" denote "True Positives", "False Positives" and "False Negatives", respectively. The "F_1-Score" and "Matthews Correlation Coefficient" (MCC) are combined evaluation metrics for binary classification.

The direct comparison of MainSEL with the current state-of-the-art— Lazrig et al. (2018)—is complicated by the fact, that no implementation of their algorithm is publicly available. Therefore, we implemented the algorithm following the description in the publication. The algorithm works by generating Bloom filters using the following fields or fragments of fields:

1. First name + Last name + Date of Birth,
2. Date of Birth + SSN,
3. Last name + SSN,
4. Three letters first name + Three letters last name + Soundex first name+ Soundex last name + Date of birth + SSN.

The parameters are chosen according to the publication as well, that is, Bloom filters of length $1{,}000$ bit, 30 hash-functions and a random salt. As record linkage quality and not performance is benchmarked, we chose to not implement their differentially private blocking and their Bloom filter partitioning scheme. These components would improve runtime performance at the cost of decreased accuracy. MainSEL uses the parameters shown in Table E.2 where applicable. For field "SSN" a frequency of 10×10^{-9}—based on assumed uniformly i.i.d. digits in the SSN—and an error rate of 0.088 is chosen—the same as for the first name. MainSEL's Bloom filters use 15 hash-functions at a length of 500bit.

Remember that the datasets used in this comparison do not allow empty fields. That means, that fewer perturbations can be applied, and the individual perturbation probability is adjusted accordingly. Table 5.4 shows, that both algorithms perform very well, so that an exaggerated field perturbation probability is necessary for the differentiation of linkage quality. The results shown are based on a field perturbation probability of 40 %, two datasets with each $10{,}000$ records and 60 % overlap.

Table 5.4: Comparison of record linkage quality between the state-of-the-art algorithm and MainSEL. The error rate is per field is set to 40 %. Each dataset has $10{,}000$ records with 60 % overlap between sets. "TP", "FP" and "FN" denote "True Positives", "False Positives" and "False Negatives" respectively. The "F_1-Score" and "Matthews Correlation Coefficient" (MCC) are combined evaluation metrics for binary classification.

	TP	FP	FN	Recall	Precision	F_1	MCC
Lazrig et al. (2018)	5917	2	63	0.989	0.999	0.993	0.986
MainSEL	5970	1	31	0.995	0.999	0.997	0.993

5.5.2 Performance Benchmarks Experiment Setups

Two different lab setups were used for MainSEL's performance evaluation, as the first setup, used for evaluating the record linkage mode of operation, was not available any longer for the matching mode of operation. However, the resulting measurements are compatible between both setups, thus the reevaluation of the record linkage experiments is not necessary.

The first lab environment—used for evaluating the record linkage mode of operation—consisted of two identical servers with Intel Xeon E5-2690 CPUs (2.90 GHz), 256 GiB RAM each and a local 1 Gbit/s network connection. Both servers ran a recent Arch Linux OS with vanilla Kernel version 4.20.7 and gcc version 8.2.1 for source code compilation.

The second lab environment—used for evaluating the matching mode of operation and the complete containerized system—consisted of two virtual servers with virtual 6 Core processor, 24 GiB of RAM each and a local 1 Gbit/s connection. Both servers ran a Debian 10.12 Linux OS with vanilla Kernel version 4.19.208-1 and gcc version 8.3.0 for source code compilation. We averaged all benchmarks over 10 independent runs.

All ABY security parameters were chosen to achieve a symmetric security level of 128 bit. Furthermore, the bit-length of the arithmetic circuit components were set to $L = 32$ bit, resulting in a score accuracy within 0.1 % (cf. Section 5.3.5). All reported record linkage timings are averaged over at least five iterations, the set intersection cardinality timings are averaged over ten iterations. All benchmarks—except where specifically noted—are using the default EpiLink

configuration that is shipped with the Mainzelliste software (see Appendix E.2), consisting of four Dice-compared and four equality-compared fields. The parameters of this default configuration follow Sariyar, Borg, and Pommerening [76].

[76] Sariyar, Borg, and Pommerening (2011)

Three different network models are of special interest: **A:** *Local Area Network* (LAN)
B: Restricted throughput (100 Mbit/s), but no imposed latency.
C: *Wide Area Network* (WAN).
For details please see Appendix B, the used network parameters are shown in Table 5.5.

Table 5.5: Network parameters for the experimental evaluation of MainSEL

Setting	Bandwidth	Latency
A	1 Gbit/s	<0.4 ms
B	100 Mbit/s	<0.4 ms
C	1 Gbit/s	100 ms

Furthermore, we analyze the behavior of four different protocol variations, namely:

GMW: The system is using the GMW without conversion to arithmetic secret sharing during score evaluation.
Yao: The system is using Yao's Garbled Circuits without conversion to arithmetic secret sharing during score evaluation.
GMW/A: The system is using the GMW protocol and conversion to arithmetic secret sharing during score evaluation.
Yao/A: The system is using Yao's Garbled Circuits and conversion to arithmetic secret sharing during score evaluation.

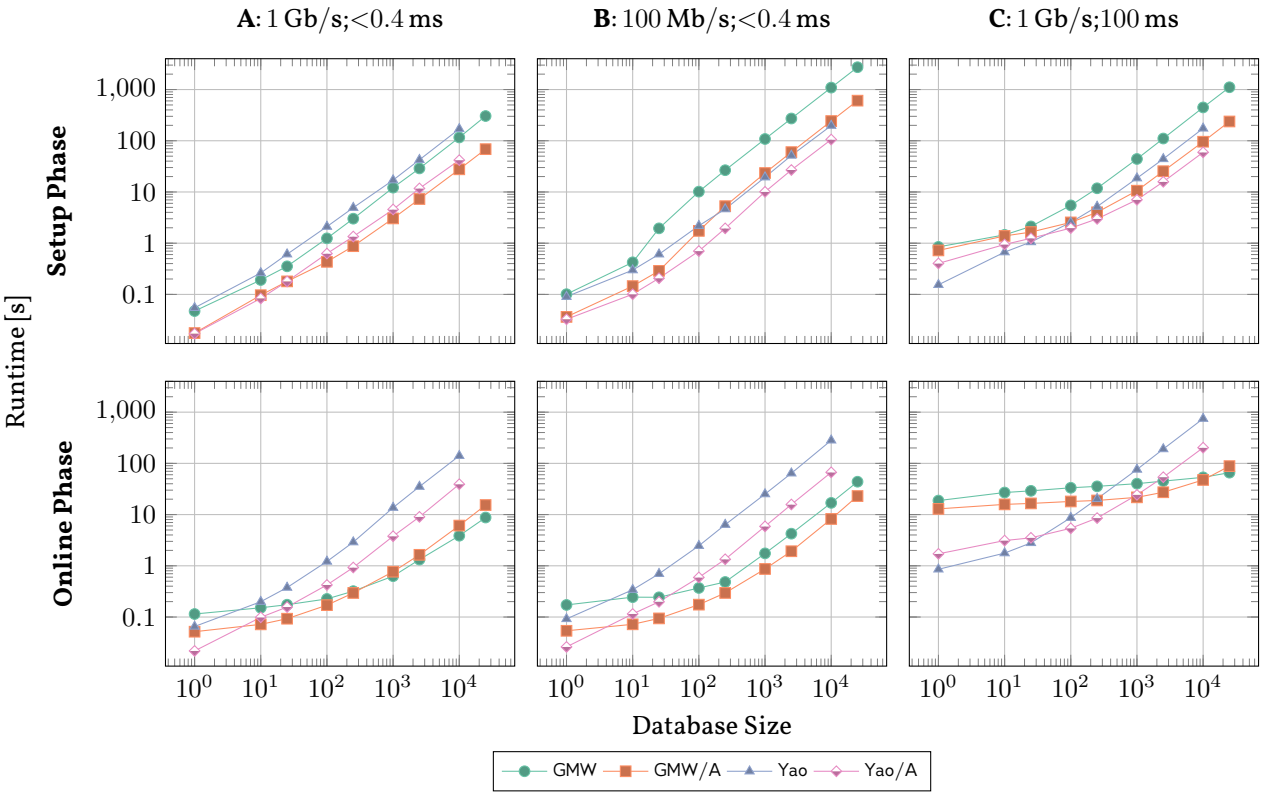


Figure 5.5: Setup and online record linkage runtime in seconds for *varying database sizes* and four circuit variants (cf. page 103), in three network environments: **A**: 1 Gbit/s;<0.4 ms, **B**: 100 Mbit/s;<0.4 ms, **C**: 1 Gbit/s;100 ms. The Epilink configuration of DKFZ's Mainzelliste (Table E.2 in Appendix E.2) was used in all benchmarks.

5.5.3 Performance Benchmarks

Figure 5.5 reports MainSEL's *record linkage* runtimes for varying database sizes and MPC circuit implementations, in three different network environments. The

Database	Comm. [MiB]			Setup Phase [s]			Online Phase [s]		
Size	#Rounds	Setup	Online	A	B	C	A	B	C
1	266	0.6	0.1	0.018	0.036	0.72	0.052	0.054	13
10	330	5.5	0.7	0.097	0.15	1.4	0.072	0.072	16
25	346	13.5	1.7	0.18	0.29	1.6	0.093	0.094	17
100	378	53.7	6.7	0.43	1.7	2.5	0.17	0.17	18
250	394	133.9	16.8	0.87	5.3	4	0.29	0.3	19
1,000	426	555.2	47.1	3	23	11	0.77	0.87	22
2,500	458	1,394.1	119.5	7.3	60	25	1.6	1.9	27
10,000	490	5,577.4	459.4	28	240	96	6.1	8.2	48
25,000	506	13,917.9	1,150.3	69	610	240	15	23	88

Table 5.6: Comparison of the setup and online record linkage runtimes of the MPC linkage procedure of a single record with a remote database in circuit variant GMW/A. Compared are the three networking configurations shown in figure 5.5, for *varying database sizes*. The reported network communication cost is the sum of sent and received data. See appendix E.1 for the complete set of tables.

database sizes ranged from 1 to 10,000 records. The four assessed circuit variants are described on page 103. The tables containing all benchmark results are shown in Appendix E.1.

As one of the network settings exhibit an exaggerated network latency of 100 ms—network setting **C**—the number of communication rounds plays an important part in the overall protocol performance evaluation. Remember, that GC has a constant number of communication rounds while both Boolean and arithmetic GMW requires one interaction round per layer of AND gates—that depends on the circuit's *multiplicative depths*[77]. Table 5.6 shows the circuit's multiplicative depth for varying database sizes and shows a logarithmic relationship between the two parameters. Starting with 266 rounds for one record and GMW/A, the number of rounds grows to 506 for 25,000 records. This logarithmic relationship can be explained by the algorithmic design of the record linkage circuits: While the first parts of any circuit runs in parallel for all database record—thus with a fixed circuit depths independent of the number of database entries—the second part, the maximum score determination, is constructed as a balanced binary-tree, explaining the logarithmic growth.

[77] See Section 2.3

After a "transient phase"—a ramp-up for small database sizes—all circuit variants exhibit a linear asymptotic runtime complexity. The transient phase is most

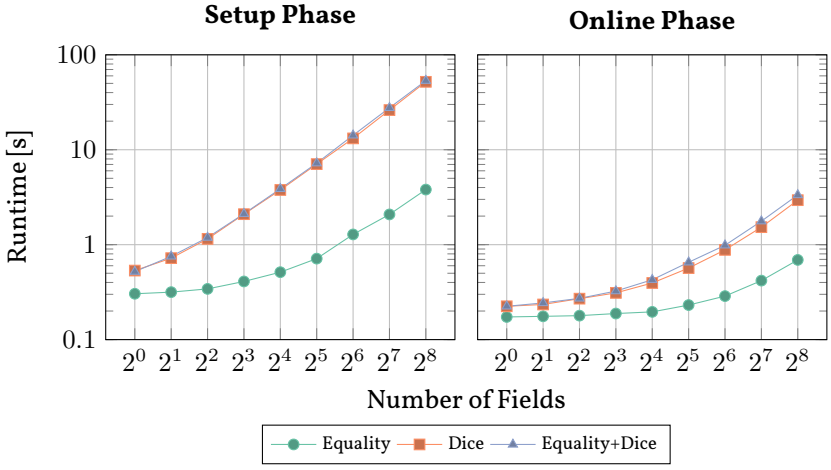


Figure 5.6: Setup and online record linkage runtime in seconds for *varying number of fields* and *varying field types*: 1) only 12 bit integer fields with equality comparison 2) only 500 bit Bloom filters with Dice comparison or 3) both, counted as pairs. Network environment **A**: LAN was used with a database size of 1,000 records and the GMW/A circuit variant.

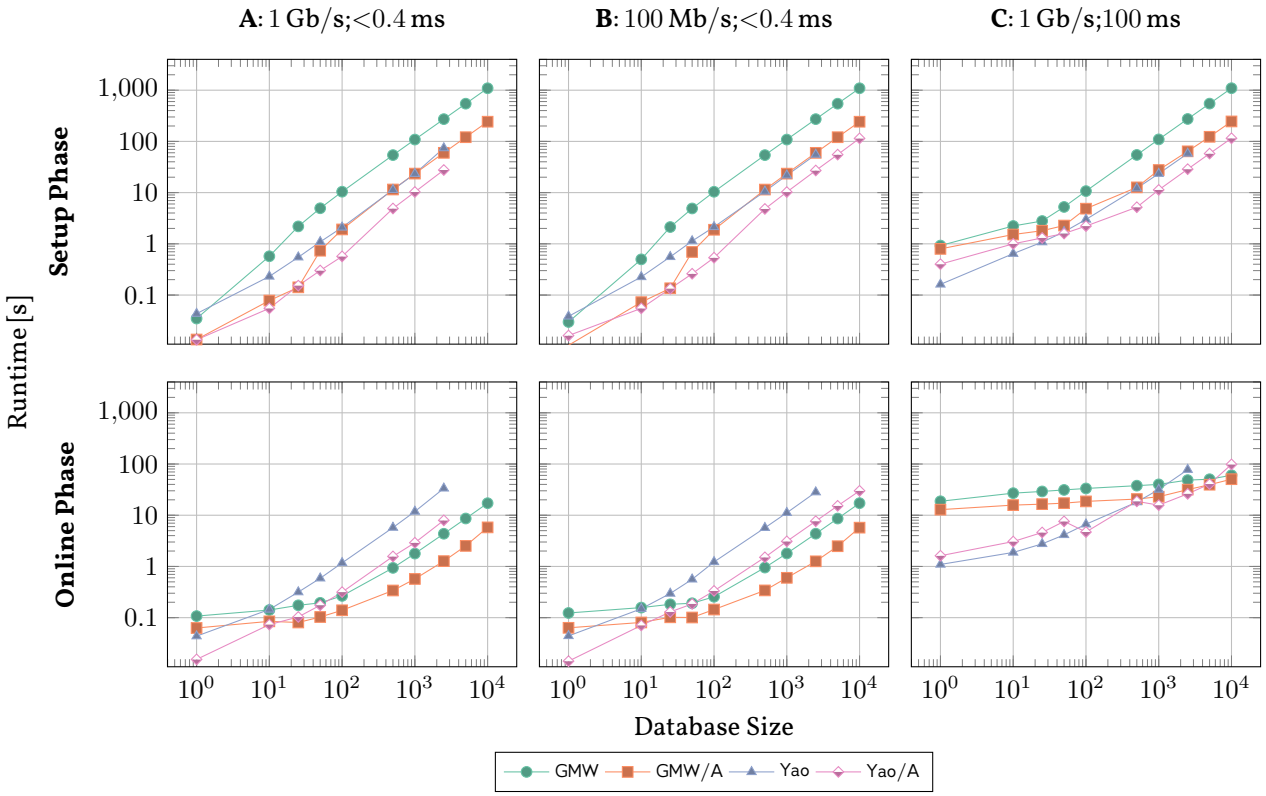


Figure 5.7: Setup and online matching runtime in seconds for *varying database sizes* and four circuit variants in three network environments: **A**: 1 Gbit/s;<0.4 ms, **B**: 100 Mbit/s;<0.4 ms, **C**: 1 Gbit/s;100 ms. The Epilink configuration of DKFZ's Mainzelliste (Table E.2 in Appendix E.2) was used in all benchmarks.

pronounced for the high-latency network setting **C** and all GMW protocol variants. Each communication round induce an additional runtime penalty due to the network delay. For larger database sizes, this effect is no longer dominant, as the larger communication size per round amortize the multiple-rounds influence. Bandwidth becomes the limiting resource—compare network settings **B** and **C**. In conclusion, the GMW/A protocol mixture performs best in nearly all networks settings, both in the setup and in the online phase.

A similar pattern can be observed for a growing number of *fields*—shown in Figure 5.6—keeping the database size fixed at 1,000 records. A similar transient phase occurs for smaller communication sizes per round. By comparing the runtimes for Dice-compared Bloom filter fields to the runtimes of the combined "Equality + Dice" experiment it is visible, that the equality-compared integer fields do not contribute considerable to the overall runtime—not surprising considering the difference in complexity between both field- and comparison types[78].

[78] cf. Circuit 5.4 and Circuit 5.5.

Set Intersection Cardinality Benchmarks

THE SAME SET of network settings and protocol variants are compared in *matching mode*—that is calculating the set intersection cardinality between two datasets. As before, one record is matched against a database of varying size. The results are shown in Figure 5.7. The results are—irrespective of more noise [79]—compatible with the record linkage mode. This is not surprising, as the sole circuit difference is the sum over all matching bits. The experiments show, that a matching computation with 10,000 comparisons is concluded in roughly five minutes considering the worst case network model and around 20 % faster in the best

[79] Probably caused by varying workloads in other virtual server instances on the same physical server.

Database	Comm. [MiB]			Setup Phase [s]			Online Phase [s]		
Size	#Rounds	Setup	Online	A	B	C	A	B	C
1	266	0.6	0.1	0.014	0.01	0.8	0.063	0.063	13
10	330	5.7	0.7	0.078	0.073	1.5	0.085	0.081	16
25	346	14.1	1.7	0.14	0.14	1.8	0.081	0.1	16
50	362	28.1	3.4	0.73	0.69	2.3	0.1	0.1	17
100	378	53.7	6.7	1.9	1.9	4.9	0.14	0.14	19
500	410	279	25.6	12	11	13	0.34	0.34	21
1,000	426	557.8	47.1	24	23	28	0.57	0.6	23
2,500	458	1,394.4	115.5	60	60	64	1.3	1.3	32
5,000	474	2,788.6	222.5	120	120	120	2.5	2.5	39
10,000	490	5,577.4	444.9	240	240	250	5.8	5.7	51

Table 5.7: Comparison of the setup and online runtimes of the MPC RL based intersection cardinality procedure of varying numbers of records in circuit variant GMW/A. Compared are the three networking configurations from figure 5.7, for *varying database sizes*. The reported network communication cost is the sum of sent and received data.

setting. Table 5.7 displays the results of the GMW/A protocol variant.

5.5.4 Real-World Deployment and Tests

In the context of the *(German) Medical Informatics Initiative* (MI-I) use case *Collaboration On Rare Diseases* (CORD_MI) we had the opportunity to conduct several real-world evaluations in eight German medical centers and university locations (shown in Table 5.8). The assessments which legal conditions govern the processing of real patients' data using MPC protocols are still pending. Hence, synthetic datasets were used for these evaluations.

Two goals were pursued with these tests: first the collection of feedback from (bio)medical researchers regarding the user experience of MainSEL and how well it solves their requirements. Second, we wanted to enable researchers and technical personnel to gather experience with MPC applications. Unfortunately, MPC is often seen as an experimental technology only suited to academic research and not as a well understood, mature set of techniques based on decades of research. We are convinced, that MPC techniques are able to solve pressing problems in medical research, especially in the field of rare diseases.

As we were interested in the feedback from a broad range of researchers, we tried to lower the barriers to participate in the experiments by supplying helper scripts, pre-setup configuration files and a free-to-use OpenVPN server. The *Extract, Transform, Load* (ETL) processes employed in these "ad hoc" setups differ from a future production deployment. For the evaluations *Comma Saparated Value* (CSV) files were used as data sources.

Mainzelliste, and therefore MainSEL, can directly couple with *Health Level Seven International* (HL7) *Fast Healthcare Interoperability Resources* (FHIR) R4[80] based

[80] https://www.hl7.org/fhir/R4

Table 5.8: Institutions triplet teams participating in the synthetic data, real world evaluations.

Party 1	Party 2	Party 3
University Medical Centre Mannheim	RTWH Aachen University	Berlin Institute of Health
University Hospital Carl Gustav Carus, Dresden	University Hospital Frankfurt	University Medical Centre Mannheim
University Hospital Tübingen	University Hospital Würzburg	University Hospital Regensburg

pipelines, which is a preferred and highly encouraged way of data loading for permanent installations. HL7 FHIR R4 is an established standard in medical informatics, and actively supported as an interface language in many hospital or laboratory information systems.

Synthetic Data Generation

For the promotion of specialized rare disease documentation—like Orpha codes—and interoperability, the CORD_MI project created a synthetic dataset for usage in German medical facilities. The synthetic datasets for the real-world evaluation of MainSEL are loosely based on those "gold-standard" data.

The CORD_MI data were modeled to cover the "MI-I Kerndatensatz"[81] (common-core dataset of the German medical informatics initiative) following real world hospital statistics distributions—including vastly more information than required in MainSEL's evaluation. We extracted the attributes of the module "Person"—the demographic and identifiable data of a patient. Around 54,000 records were prepared as CSV tables. As the CORD_MI dataset records is not suitable for a record linkage evaluation[82], we increased the variance in the fields "first name", "last name", and "city" by sampling those fields from the list of 50,000 random records, used in the record linkage quality evaluation (cf. Section 5.5.1). Including a pairwise overlap and a small common overlap between all fragments, we split this database into three parts with around 18,000 records each—shown in Figure 5.8.

[81] Ganslandt et al. (2018)

[82] The fields "first name" exhibits only two different values followed by a random number, e.g., "Hans_143" and "Grete_322". The fields "last name" and "city" show similar restrictions.

System Runtime Performance

In addition to the "bare" MPC performance benchmarked in Section 5.5.3, the runtime performance of the composed MainSEL system is of interest for real-world deployment as well. Calculating the set intersection cardinality between two parties with 100 patients each—requiring 10,000 comparisons in total—we measured the runtime of the additional cryptographic elements—namely Stunnel and the OpenVPN components. The composition of the overall runtime is displayed in Figure 5.9. Both parties used the system specifications outlined in Section 5.5.3. The central OpenVPN server ran on a dedicated server using an AMD EPYC™ 7702 processor with 4 dedicated cores running on 3.34 GHz, 16 GiB RAM and a 2.5 Gbit/s network interface. Note, that the network bandwidth is sufficient to fully saturate both clients' network interfaces.

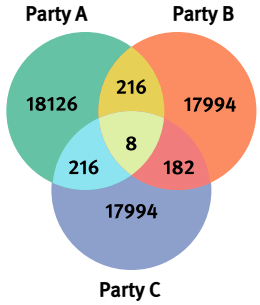


Figure 5.8: All three generated datasets consist of roughly 18,000 records including a pairwise overlap of around 200 records. In addition, 8 records are included in all three datasets. This figure was created by the author and used in Kussel et al. (2022), licensed under CC-BY.

5.6 Discussion

5.6.1 Setup and Online Phases Division

The separation of a MPC calculation into two distinct phases—the input data independent *setup phase* and the subsequent *online phase* when all inputs are known[83]—enables a useful *online* mode of operation. The designed record linkage and matching mode circuits only depend on the number of input records and the field structure—that is the EpiLink configuration—but not on the data itself. Assuming that this information remains—more or less—immutable, two MainSEL instances can run the setup phase ahead of time and only need to perform the quicker online phase[84] once the computation is actually triggered. One pos-

[83] cf. Section 2.3

[84] The online phase usually requires an order of magnitude less communication, thus, running much faster.

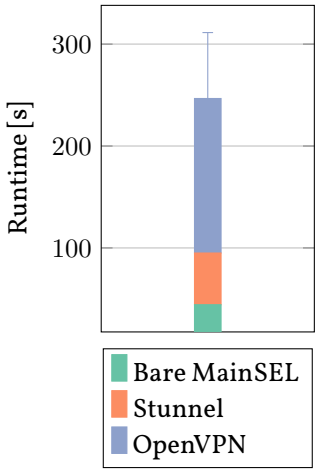


Figure 5.9: Composition of the full MainSEL system runtime for the set intersection cardinality calculation between two databases with 100 patients each. The "Bare MainSEL" setup consists of only the PostgreSQL, Mainzelliste, and Secure EpiLinker containers.

sible application of using MainSEL as an online system would be the continuous record linkage between two databases with updating the LID's every time a new record is inserted—i.e., a new patient is admitted. In this mode of operation the division of setup and online phase would turn the online phase runtimes to be the significant, observable ones. However, one initial full database cross-linkage would be required nonetheless.

5.6.2 Performance and Complexity Discussion

Using circuit variant GMW/A and the fastest network environment—network setup **A**—a full cross-linkage of two medium-to-large-sized databases with 10,000 patients each would take 78 h for the setup and 17 h for the online phase—approximately 4 days in total. In the high latency networking setup **C**, it would take almost 17 days. These runtime extrapolation remain applicable to the calculation of the record linkage-based private set intersection cardinality. However, for many matching mode use cases—e.g., cohort size estimations for rare disease research—the expected number of records in both databases is considerably smaller.

For privacy reasons[85] we do not incorporate blocking techniques in our record linkage procedure, which would drastically reduce record linkage runtimes. Currently, all M records from one database are compared to N records of the second one. This quadratic number of comparisons scale very quickly, hence, even a "factorial" decrease in the number of comparisons, i.e., by binning, would notably benefit the performance.

For many applications this full linkage would only be needed to run once initially, when two parties enter the secure record linkage system. Once the systems are linked, updating by including a newly admitted patient to an existing database of size 10,000 would take 6.1 s online time for circuit variant GMW/A or 4 s in the pure GMW protocol, assuming network setting **A**. In high-latency network environment **C**, it would take 48 s for protocol variant GMW/A.

The runtime complexity differs between full cross-linkage and online usage mode: due to the exhaustive pair comparisons, the computation- and communication complexity is $\mathcal{O}(M \cdot N)$ for full *initial* cross-linking, while during normal operation the complexity becomes $\mathcal{O}(N)$, i.e., linear in the size of the data source. This linear complexity results in practical runtimes for MainSEL in a broad range of practical applications.

From a runtime perspective, the inclusion of, for example, equality-compared *Medical Data* (MDAT) in addition to the demographic data would not heavily impact runtime performance. As Figure 5.6 shows, the impact of simple equality comparisons is nearly negligible compared to Bloom-Dice comparisons.

Protocol Variants and Network Settings

Based on the usage scenario requirements the choice of an optimal configuration varies. As discussed before, for many environments the optimization for fastest online phase runtimes is sensible, as the setup phases can be performed be-

[85] X. He et al. (2017)

tween timing critical online phases. Excluding the edge-cases of small databases and very high latencies, the GMW/A protocol mixture constitutes a balanced default configuration. This observation enables non-technical personnel to deploy and use MaiSEL without benchmarking extensively beforehand.

These results are in agreement with Demmler, Schneider, and Zohner 's[86] insight, that the hybrid usage of mixed MPC protocols proves more efficient in most applications. The runtime improvements gained by choosing the appropriate protocol for each algorithmic section outweighs the additional computation and communication incurred by the protocol conversions.

[86] Demmler, Schneider, and Zohner (2015)

As it is widely known, network communication presents itself as MPC's principal bottleneck. By restricting the network bandwidth or incurring additional latency between both parties—or both—runtimes substantially increase. For 10,000 record comparisons the runtime of matching differs more than 20 % between the best and the worst network model. This is unsurprising, as, first, large amounts of data must be transmitted[87], and second, the latency incurs a runtime penalty for each of the multiple hundred interaction rounds in the GMW protocol.

[87] Table 5.7 shows, that for 10,000 comparisons the required communication size exceeds 6 GiB.

Impact of Firewall and Proxy Traversal

The laboratory[88] and real-world performance benchmarks[89] attest MainSEL practical and feasible performance for real-world workloads, showing it to be useful tool for medical researchers. However, the composition of the runtime shown in Figure 5.9 reveal the high overhead of the components included for firewall and proxy traversal. Compared to the "bare" system using only Mainzelliste and SEL, the complete MainSEL system performs approximately 5.6 times worse. Removing OpenVPN would result in a 259 % performance boost, further removing Stunnel would result in an additional 214 % runtime improvement. A future direction of research and engineering strongly suggested by this fact is the exploration of other proxy and firewall traversal mechanisms[90], as well as the decoupling of ABY's network layer to directly implement authenticated communication channels. These steps would further increase MainSEL's capability to adapt to larger workloads.

[88] See Section 5.5.3

[89] See Section 5.5.4

[90] While the commonly used protocols and methods, e.g., STUN (RFC 8489), TURN (RFC 5766), and ICE (RFC 8445), work with TCP traffic, most implementations only handle UDP traffic, as media streaming is the most common use case.

Results from Real-World Testing

The deployment and evaluation of MainSEL in eight German university hospitals and universities[91] was successful as all test sites managed to perform the complete test suite with correct results. Thanks to the feedback of the researchers gathered during the tests we were able to improve MainSEL's robustness and adapt it better to the researchers' needs.

[91] see Table 5.8

While technically interoperable with all institutions' firewalls, the used OpenVPN network can be used to circumvent the firewall rules and network policies. For the trial experiments using synthetic data this was acceptable, however, for operational deployments specialized gateways defined and configured collaboratively with the institutions' IT security teams are mandatory.

5.7 Beyond Demographic Data

Although this work describes only record linkage operations based on IDAT, the algorithm is easily generalizable to include other types of data.

Inspired by the success of the strategy to transform problems originating firmly in the graph theory realm to similarity measure spaces, we identified three related data types which are not thoroughly explored yet but show promising first results. All three applications are based on a transformation "pipeline" instead of finding approximate subgraph isomorphisms: The graph is decomposed into fragments or paths of a certain length, those fragments, including vertex- and edge properties, are inserted into a binary data structure by means of *locality sensitive hashing*, and lastly probabilistic similarity measures are computed. It turns out, that this approach is promising for finding "broad" similarities—that is, where a certain generality is required and no specific criteria of similarity can be given. This section introduces and sketches three possible expansions: Comparison of small molecules with regard to chemical similarity, the discovery of similar patients based on electronic health records, and the identification of disturbed biological pathways using transcriptome similarity comparisons.

5.7.1 Chemical Similarity of Small Molecules

Small molecules play an important role in pharmaceutical research. Finding promising substances to bind to specific protein sites is an expensive and time-consuming procedure. Pharmaceutical companies want to collaborate in this endeavor, but are unwilling to disclose their database of candidates. By calculating the overlap "in the chemical space" in a privacy-preserving fashion, the development of new targeted drugs in personal health could be performed quicker and more expedient.

One real-world use case is the classification and comparison of cystic fibrosis drugs in the CandActCFTR database[92]. This database collects information regarding the chemical structure of published drug trials, and additional annotations, such as the tested cell lines, mutation variant of the CFTR gene, etc. Unfortunately, pharmaceutical companies are seldom willing to share their *negative* experiment results and have neither commercial nor publication incentives to disclose those substances. However, providing privacy-preserving means to demarcate the sampled *chemical space* could lead to a more open flow of knowledge and help with allocating research funds to more promising drug candidates.

[92] Nietert et al. (2021)

This application requires a broad concept of chemical similarity. However, focusing on specific properties, such as charge distribution, limits the measure's expressiveness, hence, its utility for the given application.

Fragment-based molecule fingerprints[93]—that is, the molecule graph is separated into paths of a certain length and these fragments are hashed to generate a bit vector representing the molecule—are fast structures for molecular similarity calculation and *Tanimoto* coefficient[94] comparisons $T(X,Y) = \frac{|X \cap Y|}{|X|+|Y|-|X \cap Y|}$[95] achieve a good correspondence to chemical similarity in many applications[96].

[93] Cereto-Massagué et al. (2015)

[94] Tanimoto (1958)

[95] Compare to the Dice-Sørensen coefficient in Section 5.2.2

[96] Sheridan and Kearsley (2002); Bajusz, Rácz, and Héberger (2015)

There are suitable fragment-based fingerprinting algorithms available[97], however for this application they need modification, e.g., to allow configurable length.

[97] For example, the "FP2" fingerprinting algorithm, implemented in the open source package "Open Babel": https://openbabel.org/docs/dev/Fingerprints/fingerprints.html.

5.7.2 Similar Patient Discovery

For certain groups of patients, e.g., undiagnosed patients or cancer cases outside established medical guidelines, the discovery of similar patients open the way to diagnoses or promising treatments. Unfortunately, most of the time those cases are individually rare, that means a similar patient is most likely to be found in the records of a different hospital, and most of the time it is not clearly defined what the relevant criteria of "similar" are.

Based on the assumption, that similar diseases cause similar trajectories or "artifacts" in the medical history, one could undertake the identification by modelling the patients' medical history as a graph. In an "arborescent" modelling all lab values, medical images, hospital encounters, demographic data, etc. are structured hierarchically into an acyclic graph. By "similar trajectories or artifacts" we assume, for example, that patients that have multiple chest NMR images, laboratory values for inflammation markers, and gene sequencing of the BRCA1/2 genes, are *highly likely* suffering from breast cancer, in some kind, form, or fashion.

As patient similarity is an important research topic, for example in clinical molecular tumor boards or for the detection of disease subtypes[98], a large body of work exists[99]. However, to the best of our knowledge, no privacy-preserving similar patient analysis algorithm exists.

[98] L. Li et al. (2015)

[99] Parimbelli et al. (2018)

5.7.3 Transcriptome Pathway Defect Identification

Between genomic information and expressed phenotype lay many distinct steps and "realms", all with their own regulatory dependencies and malfunction opportunities. One of those "realms" is the domain of *transcriptomes*. Transcriptome measurements consist of snapshots of all transcription processes within a cell, that is concentrations of RNA strands. From these concentrations a regulatory graph can be constructed, linking genomics to proteomics. Edges, that is dependencies in the system, can be inhibitory or activating. Through privacy-preserving similarity matching of transcriptome graphs, using the graph similarity techniques described above, against a curated, labeled database, previously undetected defects in transcription pathways could be identified[100].

[100] Xu et al. (2016)

5.8 Outcome and Prospects

MainSEL, the MPC-based *Privacy-Preserving Record Linkage* (PPRL) framework described in this chapter, is the first practical probabilistic PPRL solution allowing the linkage and calculation of record linkage-based *Private Set Intersection Cardinality* (PSI-C) without information leakage[101] while handling noisy, incomplete and heterogeneous data. By using MPC techniques the functionality is easily extendable and provably secure under rigorously defined threat models. With that

[101] In the semi-honest setting and the absence of a sufficiently powerful quantum computer.

MainSEL fulfills the requirements of the real-world medical research environment.

By exploiting structural similarities between the probabilistic record linkage graph problem and the corresponding table-based problem—namely the insight, that the fully normalized relational database structure matches the graph structure of the problem—we were able to transform the $\mathcal{NP}$-hard task of finding approximate subgraph isomorphisms to a less general but application appropriate problem space exhibiting polynomial complexity.

Using highly optimized hybrid MPC circuits for the functionality—including a novel high-level approach to generate optimized integer division circuits—MainSEL achieves practical runtimes for medium to large sized databases[102] completing $10{,}000$ comparisons in around 5 min—including all overheads originating in the usage of OpenVPN and Stunnel.

[102] In an online mode of operation even for very large databases.

To allow the production deployment, we provide an open source containerized software package ready to be used with minimal configuration in real hospital networks. To assure easy integration and up-to-date deployments we follow the *Continuous Integration/Continuous Deployment* (CI/CD) paradigm utilizing automated build and containerization pipelines.

The laboratory performance measurements as well as the real-world experiments conducted between eight German university medical centers in cooperation with researchers in the rare disease's community attest MainSEL practicality when it comes to applications in medical research. The feedback collected during these tests helped us customize MainSEL and its documentation to better satisfy the research needs. Evaluations using real patient data are in preparation and scheduled for the second half of 2022.

The focus of MainSEL on data privacy leads to vastly higher security guarantees than existing PPRL solutions—often based on centralized bloom filters. In particular:

- The processing of sensitive patient data uses modern, tried-and-tested cryptographic techniques for protection.
- No central component and no TTP is required in the computation. No sensitive information leaves the institution's network perimeter, no IDAT of any kind is centrally collected.
- Even in the case of a data breach in a participating party, only the compromised parties' data is at risk—the input data of all other parties remains private.

These improved security guarantees elevate the state-of-the-art in record linkage and opens up innovative research opportunities by potentially being able to perform record linkage without data transfer consent and associated high regulatory burden.

During the work on MainSEL, we identified possibilities for further improvement and research that fall into two broad categories: First, improvements of the secure record linkage algorithms and second, extensions of the interfaces and application.

One often requested feature is the inclusion of more than two parties. While the usage of MainSEL with more than two data owners and two computation parties—an outsourcing scenario[103]—is easily achievable, "real" multi-party record linkage is not trivial, as most probabilistic record linkage similarity measures are not transitive—Alice matching Bob and Bob matching Carol does not imply a match between Alice and Carol. The naïve application of the EpiLink similarity to that scenario could lead to reduced matching quality and conflicting matching status between multiple parties. On a more technical side, most (practical) MPC protocols scale quadratic in the number of participating parties, leading to challenges when it comes to runtime performance of those possible n-party protocols.

[103] See Section 2.3.6

A major performance detriment of MainSEL is the exhaustive pairwise comparison of records leading to $N \cdot M$ record linkage operations when linking two databases of size N and M, respectively. Computation reduction strategies—blocking—are not easily applicable, as often used techniques, such as *Locality-Sensitivity Hashing* (LSH), are known to impair MPC's strong security guarantees[104]. The development of *Oblivious Random Access Memory* (ORAM)-based blocking constructions might provide a way to reduce computational workload while providing full security.

[104] X. He et al. (2017)

While MainSEL is tested in real clinical networks, the utilization of the firewall and proxy traversal using (TCP-based) variants of, e.g., STUN[105], ICE[106], and TURN[107] would increase performance up to 557 % and we consider this the next step to improve hospital deployment.

[105] RFC 8489, Accessed 2022/05/08.

[106] RFC 8445, Accessed 2022/05/08.

[107] RFC 5766, Accessed 2022/05/08.

Lastly, to further enhance the flexibility of MainSEL and allow the usage with non-IDAT data types, the development and implementation of additional similarity measures and matching classifiers is necessary. Section 5.7 sketches three promising extension possibilities.

Part III

Final Remarks

Chapter 6
Conclusion

In this dissertation, privacy-preserving biomedical distributed analyses were explored through the lens of graph theory and statistical physics. While some research questions did not deal with "graph problems" in the strictest sense, the methodology and associated epistemology of the fields mentioned were employed to find efficient, secure solutions and—if required—transform the problem to a more effectively solvable problem space.

Privacy-Preserving Epistasis Analysis

Circuit-based *Secure Multi-Party Computation* (MPC) techniques, as well as novel cryptographic building blocks, were used to design, implement, and experimentally evaluate a protocol suit for privacy-preserving epistasis analysis (Chapter 3).

Previous (unencrypted) analyses of epistasis (gene–gene and gene–environment interactions) led to valuable insights regarding the complex regulatory genetic networks associated with certain diseases, paving the way for novel treatments[1]. As large datasets of sensitive genomic data are required for this kind of analysis, privacy-preserving distributed computations are a natural fit. However, due to the computational complexity of the analysis of large, coupled systems, the only recent advances considering privacy are based on *Statistical Disclosure Control* (SDC) techniques—more specifically based on *Differential Privacy* (DP)[2]—hence reducing the data's utility[3].

[1] H. Zhang et al. (2017); Yang et al. (2017); Meng et al. (2017); Y. M. Cho et al. (2004); Liu et al. (2009)

[2] T. T. Le et al. (2017); Chen, X. Zhang, and R. Zhang (2019)

[3] Naveed et al. (2015)

By designing and implementing the novel, efficient cryptographic building blocks *Arithmetic Greater Than* (AGT) (Section 3.3.1) and ASWAP (Section 3.3.2), we were able to design, implement, and evaluate a suit of privacy-preserving MPC protocols for the performance of feature selection on large genomic datasets (Section 3.2) and utility-lossless *Multifactor Dimensionality Reduction* (MDR) (Section 3.3). Both tasks work in conjunction. While the final analysis of the non-linear models is performed using *Private Multifactor Dimensionality Reduction* (PMDR), the feature selection algorithms remove noisy features, hence, increasing the quality of the resulting correlations and optimizing runtime performance (Section 3.6.4).

While providing researchers with algorithms and tools to perform research not possible before, due to data protection regulations, we contributed to a powerful general MPC framework by extending its functionality and provided the first implementation of a novel garbling scheme[4] (Section 3.5.1).

[4] Rosulek and Roy (2021)

Privacy-Preserving Solution to the Kidney Exchange Problem

By designing privacy-preserving, MPC-based protocols for the construction and evaluation of a medical compatibility graph, a heuristic (local) optimizer for the graph problem of kidney exchange was developed (Chapter 4). The developed software—*Secure and Private Investigation of the Kidney Exchange problem* (SPIKE)—meets all functional requirements with regard to privacy, efficiency, decentralization and (medical) flexibility, providing medical professionals with a tool to speed up and reduce the cost of kidney exchanges, thus allowing more patients to access a potentially live-saving procedure.

SPIKE achieves a $400\times$ to $30{,}000\times$ speedup compared to the state-of-the-art (Section 4.3.2) and enhances the medical quality of the solutions by implementing a more thorough compatibility assessment of donors and recipients during compatibility graph construction, following recent transplantation guidelines and medical evidence (Section 4.1).

Secure Record Linkage and Private Graph Similarity

The problem of federated record linkage was solved with a fully decentralized MPC solution, which combines the highest security requirements with feasible runtimes. A *Trusted Third Party* (TTP) has thereby been entirely omitted (Chapter 5).

The problem of record linkage can be viewed as the search for approximate subgraph isomorphisms (Section 5.2). Preliminary work is devoted to this topic[5], but this approach is not a suitable candidate for translation into a privacy-protecting MPC protocol due to its inherent superpolynomial runtime complexity. The insight that the graph structure considered in this approach resembles that of a fully normalized database leads to the implementation of a solution that, while not as general, can be executed efficiently in polynomial time.

[5] M. Schmidt et al. (2020)

The developed framework—*Mainzelliste Secure EpiLinker* (MainSEL)—allows for fault-resistant similarity analysis (Section 5.2.2). In addition to the *Institutional Review Board* (IRB)-compliant issuance of linkage IDs via a cross-site linkage service (which is not a TTP, since it does not participate in the computation), a protocol for the direct generation of linkage IDs was also presented (Section 5.4.2). The use of circuit-based MPC leads to the easy extension of the protocol. As an example of this, a use case from rare disease research is addressed and a record linkage-based *Private Set Intersection Cardinality* (PSI-C) computation is described (Section 5.4.3). Extensive experimental evaluations of the record linkage quality and the runtime behavior of MainSEL across multiple network and protocol variations attest to MainSEL's practicality (Section 5.5). The deployment of MainSEL has been simplified and automated to the point that it has been successfully field-tested in eight university hospitals' real IT context. The use of real patient data is targeted for the second half of 2022.

Based on the matching methods developed for MainSEL, ultimately three additional extension scenarios not yet conclusively explored were presented in which

graph systems are compared for similarity (Section 5.7). A reliable comparison of pharmacologically relevant small molecules is a requested use case from cystic fibrosis research. The molecular structure of potential drugs is to be evaluated for similarity in a chemically meaningful way without giving out any information. One promising approach is to use (customized) graph fragment fingerprints in the MainSEL EpiLink algorithm. As a second extension, similar patients should be identified in distributed databases, where the similarity criterion is not sharply defined. By modeling the patient record and their medical history as a graph system, a path-based embedding in binary data structures and thus an efficient comparison is possible as well. As a final possible research avenue, the detection of defects in gene regulatory networks via the use of transcriptome analyses is outlined.

Where does the research presented in this work lead us? While the detailed accomplishments have been described above, three "intangible" aspects have been achieved: First, a way of embedding diverse graph structures with associated data into structures suitable for established, powerful similarity comparison algorithms has been shown. More applications for this technique, in addition to the three given research opportunities, are sure to come. Second, the presented research has real-world impact. Medicinal researchers are introduced to the existence of privacy-preserving MPC techniques and have been using the developed tools in test settings. The application to real medical research, protecting real patients' data is scheduled. Furthermore, data protection officers, legal scholars, and legislators have been involved and familiarized with these techniques and are starting to see the high protection levels, as well as the numerous applications of MPC. Third and ultimately, a connection has been established between methods and models from graph theory and statistical physics to cryptographic protocols and data protection in the field of medicine. The successfully finished research projects give evidence of the potential of this interdisciplinary approach and promise further algorithms and protocols following this path.

Chapter 7
References

Abraham, D. J., Blum, A., and Sandholm, T. (2007)
"Clearing Algorithms for Barter Exchange Markets: Enabling Nationwide Kidney Exchange". *Proceedings of the 9th ACM Conference on Electronic Commerce (EC)*. ACM. Association for Computing Machinery, pp. 295–304. DOI: 10.1145/1250910.1250954 (cited on p. 60).

El-Agroudy, A. E., Hassan, N. A., Bakir, M. A., Foda, M. A., and Shokeir, A. A. (2003)
Effect of Donor/Recipient Body Weight Mismatch on Recipient and Graft Outcome in Living-Donor Kidney Transplantation. *American Journal of Nephrology* 23, pp. 294–299. DOI: 10.1159/000072819 (cited on p. 64).

Akemann, G., Baik, J., and Di Francesco, P., eds. (2011)
The Oxford Handbook of Random Matrix Theory. Oxford University Press. DOI: 10.1093/oxfordhb/9780198744191.001.0001 (cited on p. 5).

Albert, R. and Barabási, A.-L. (2002)
Statistical Mechanics of Complex Networks. *Reviews of Modern Physics* 74.1, pp. 47–97. DOI: 10.1103/RevModPhys.74.47 (cited on p. 5).

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002)
"Lymphocytes and the Cellular Basis of Adaptive Immunity". *Molecular Biology of the Cell*. 4th edition. Garland Science (cited on p. 62).

Alexanderson, G. (2006)
About the cover: Euler and Königsberg's Bridges: A historical view. *Bulletin of the American Mathematical Society* 43.4, pp. 567–573. DOI: 10.1090/S0273-0979-06-01130-X (cited on p. 1).

Aly, G. and Roth, K. H. (2018)
Die restlose Erfassung: Volkszählen, Identifizieren, Aussondern im Nationalsozialismus. S. Fischer Verlag (cited on p. 6).

Ambinder, E. P. (2005)
A History of the Shift Toward Full Computerization of Medicine. *Journal of Oncology Practice* 1.2, pp. 54–56. DOI: 10.1200/jop.2005.1.2.54 (cited on p. 6).

Araki, T., Furukawa, J., Ohara, K., Pinkas, B., Rosemarin, H., and Tsuchida, H. (2021)
"Secure Graph Analysis at Scale". *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, pp. 610–629. DOI: 10.1145/3460120.3484560 (cited on pp. 59, 77).

Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., et al. (2018)
Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation: Combining Probabilistic Graphical Models with Deep Learning for Structured Prediction. *IEEE Signal Processing Magazine* 35.1, pp. 37–52. DOI: 10.1109/MSP.2017.2762355 (cited on p. 36).

Asharov, G., Halevi, S., Lindell, Y., and Rabin, T. (2018)
"Privacy-Preserving Search of Similar Patients in Genomic Data". Vol. 2018. 4, pp. 104–124. DOI: doi:10.1515/popets-2018-0034 (cited on p. 33).

Asharov, G., Lindell, Y., Schneider, T., and Zohner, M. (2013)
"More Efficient Oblivious Transfer and Extensions for Faster Secure Computation". *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, pp. 535–548. DOI: 10.1145/2508859.2516738 (cited on p. 21).

Asharov, G., Lindell, Y., Schneider, T., and Zohner, M. (2017)
More Efficient Oblivious Transfer Extensions. *Journal of Cryptology* 30.3, pp. 805–858. DOI: 10.1007/s00145-016-9236-6 (cited on pp. 17, 149).

Ashby, V. B., Leichtman, A. B., Rees, M. A., Song, P. X.-K., Bray, M., Wang, W., and Kalbfleisch, J. D. (2017)
A Kidney Graft Survival Calculator that Accounts for Mismatches in Age, Sex, HLA, and Body Size. *Clinical Journal of the American Society of Nephrology* 12.7, pp. 1148–1160. DOI: 10.2215/CJN.09330916 (cited on pp. 60, 63).

Aumasson, J.-P. (2017)
Serious Cryptography: A Practical Introduction to Modern Encryption. No Starch Press (cited on p. 12).

Bajusz, D., Rácz, A., and Héberger, K. (2015)
Why is Tanimoto Index an Appropriate Choice for Fingerprint-based Similarity Calculations? *Journal of Cheminformatics* 7.1, p. 20. DOI: 10.1186/s13321-015-0069-3 (cited on p. 110).

Baker, D., Knoppers, B. M., Phillips, M., Enckevort, D. v., Kaufmann, P., Lochmuller, H., and Taruscio, D. (2018)
Privacy-Preserving Linkage of Genomic and Clinical Data Sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1. DOI: 10.1109/TCBB.2018.2855125 (cited on p. 81).

Barbujani, G. and Colonna, V. (2010)
Human Genome Diversity: Frequently Asked Questions. *Trends in Genetics* 26.7. DOI: 10.1016/j.tig.2010.04.002 (cited on p. 35).

Barth-Jones, D. (2012)
The "Re-Identification" of Governor William Weld's Medical Information: A Critical Re-Examination of Health Data Identification Risks and Privacy Protections, Then and Now. ID 2076397. DOI: 10.2139/ssrn.2076397 (cited on p. 9).

Bashshur, R. and Shannon, G. W. (2009)
History of Telemedicine: Evolution, Context, and Transformation. Mary Ann Liebert New Rochelle, NY (cited on p. 6).

Baxter, R. J. (2016)
Exactly Solved Models in Statistical Mechanics. Elsevier (cited on p. 5).

Beaver, D., Micali, S., and Rogaway, P. (1990)
"The Round Complexity of Secure Protocols". *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*. ACM Press, pp. 503–513. DOI: 10.1145/100216.100287 (cited on p. 19).

Beaver, D. (1991)
"Efficient Multiparty Protocols using Circuit Randomization". *Advances in Cryptology – CRYPTO*. Springer, pp. 420–432. DOI: 10.1007/3-540-46766-1_34 (cited on p. 21).

Beaver, D. (1995)
"Precomputing Oblivious Transfer". *Advances in Cryptology — CRYPTO*. Springer, pp. 97–109. DOI: 10.1007/3-540-44750-4_8 (cited on p. 17).

Beaver, D. (1996)
"Correlated Pseudorandomness and the Complexity of Private Computations". *Proceedings of the 28th annual*

ACM symposium on Theory of computing - STOC. ACM Press, pp. 479–488. DOI: 10.1145/237814.237996 (cited on p. 17).

Bellman, R. (1958)
On a Routing Problem. *Quarterly of Applied Mathematics* 16.1, pp. 87–90. DOI: 10.1090/qam/102435 (cited on p. 77).

Bernemann, I., Kersting, M., Prokein, J., Hummel, M., Klopp, N., and Illig, T. (2016)
Zentralisierte Biobanken als Grundlage für die medizinische Forschung. *Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz* 59.3, pp. 336–343. DOI: 10.1007/s00103-015-2295-2 (cited on p. 89).

Bezanson, R. P. (1992)
The Right to Privacy Revisited: Privacy, News, and Social Change, 1890-1990. *California Law Review* 80, p. 1133. DOI: 10.2307/3480738 (cited on p. 6).

Biesecker, L. G., Green, E. D., Manolio, T., Solomon, B. D., and Curtis, D. (2021)
Should All Babies Have Their Genome Sequenced at Birth? *The BMJ* 375. DOI: 10.1136/bmj.n2679 (cited on p. 31).

Birka, T., Hamacher, K., Kussel, T., Möllering, H., and Schneider, T. (2022)
SPIKE: Secure and Private Investigation of the Kidney Exchange problem. *BMC Medical Informatics and Decision Making*. In Review, Pre-Print: https://arxiv.org/abs/2204.09937 (cited on pp. 58, 65).

Biró, P. and Cechlárová, K. (2007)
Inapproximability of the Kidney Exchange Problem. *Information Processing Letters* 101.5, pp. 199–202. DOI: https://doi.org/10.1016/j.ipl.2006.09.012 (cited on p. 70).

Biró, P., Klundert, J. van de, Manlove, D., Pettersson, W., Andersson, T., Bunapp, L., Chromy, P., Delgado, P., et al. (2021)
Modelling and Optimisation in European Kidney Exchange Programmes. *European Journal of Operational Research* 291.2, pp. 447–456. DOI: https://doi.org/10.1016/j.ejor.2019.09.006 (cited on p. 57).

Black, E. (2001)
IBM and the Holocaust: The Strategic Alliance Between Nazi Germany and America's Most Powerful Corporation. Crown Books (cited on p. 6).

Bonte, C., Makri, E., Ardeshirdavani, A., Simm, J., Moreau, Y., and Vercauteren, F. (2018)
Towards Practical Privacy-Preserving Genome-Wide Association Study. *BMC Bioinformatics* 19 (537). DOI: 10.1186/s12859-018-2541-3 (cited on p. 32).

Bose, P., Guo, H., Kranakis, E., Maheshwari, A., Morin, P., Morrison, J., Smid, M., and Tang, Y. (2008)
On the False-Positive Rate of Bloom Filters. *Information Processing Letters* 108.4, pp. 210–213. DOI: 10.1016/j.ipl.2008.05.018 (cited on p. 88).

Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., and Scholl, P. (2019)
"Efficient Pseudorandom Correlation Generators: Silent OT Extension and More". *Advances in Cryptology – CRYPTO*. 448. Springer, pp. 489–518. DOI: 10.1007/978-3-030-26954-8_16 (cited on p. 17).

Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (2014)
(Leveled) Fully Homomorphic Encryption Without Bootstrapping. *ACM Transactions on Computation Theory (TOCT)* 6.3, pp. 1–36 (cited on p. 25).

Brassard, G. (2005)
Brief History of Quantum Cryptography: A Personal Perspective. *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, 2005.*, pp. 19–23. DOI: 10.1109/ITWTPI.2005.1543949 (cited on p. 16).

Braun, L., Demmler, D., Schneider, T., and Tkachenko, O. (2021)
MOTION – A Framework for Mixed-Protocol Multi-Party Computation. *ACM Transactions on Privacy and Security (TOPS)* 25.2, pp. 1–35. DOI: 10.1145/3490390 (cited on pp. 20, 23, 51).

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017)
Classification and Regression Trees. Routledge. DOI: 10.1201/9781315139470 (cited on p. 38).

Breuer, M., Meyer, U., and Wetzel, S. (2022)
"Privacy-Preserving Maximum Matching on General Graphs and its Application to Enable Privacy-Preserving Kidney Exchange". *ACM Conference on Data and Application Security and Privacy (CODASPY)* (cited on pp. 58–61, 66, 72, 74, 76).

Breuer, M., Meyer, U., Wetzel, S., and Mühlfeld, A. (2020)
"A Privacy-Preserving Protocol for the Kidney Exchange Problem". *Proceedings of the 19th Workshop on Privacy in the Electronic Society (WPES)*, pp. 151–162. DOI: 10.1145/3411497.3420213 (cited on pp. 58–61, 72, 74–76).

Bryan, K. and Leise, T. (2006)
The $25,000,000,000 Eigenvector: The Linear Algebra behind Google. *SIAM Review* 48.3, pp. 569–581. DOI: 10.1137/050623280 (cited on p. 4).

Buescher, N., Holzer, A., Weber, A., and Katzenbeisser, S. (2016)
"Compiling Low Depth Circuits for Practical Secure Computation". *Computer Security – ESORICS*. Springer, pp. 80–98. DOI: 10.1007/978-3-319-45741-3_5 (cited on p. 93).

Bundesgesundheitsministerium (2021)
Seltene Erkrankungen. URL: https://www.bundesgesundheitsministerium.de/themen/praevention/gesundheitsgefahren/seltene-erkrankungen.html (visited on 08/11/2021) (cited on p. 79).

Bundesministerium des Inneren (2007)
Nationaler Plan zum Schutz der Informationsinfrastrukturen (NPSI). Technical Report. Bunderministerium des Inneren und für Heimat (cited on p. 99).

Bundesministerium des Inneren (2009)
Nationale Strategie zum Schutz Kritischer Infrastrukturen (KRITIS-Strategie). Technical Report. Bunderministerium des Inneren und für Heimat (cited on p. 99).

Carvalho, M., Klimentova, X., Glorie, K., Viana, A., and Constantino, M. (2021)
Robust Models for the Kidney Exchange Problem. *INFORMS Journal on Computing* 33.3, pp. 861–881. DOI: 10.1287/ijoc.2020.0986 (cited on p. 60).

Castro, L. de, Agrawal, R., Yazicigil, R., Chandrakasan, A., Vaikuntanathan, V., Juvekar, C., and Joshi, A. (2021)
Does Fully Homomorphic Encryption Need Compute Acceleration? Pre-Print (cited on p. 14).

Catrina, O. and Hoogh, S. de (2010)
"Secure Multiparty Linear Programming Using Fixed-Point Arithmetic". *European Symposium on Research in Computer Security (ESORICS)*. Springer, pp. 134–150. DOI: 10.1007/978-3-642-15497-3_9 (cited on p. 59).

Cereto-Massagué, A., Ojeda, M. J., Valls, C., Mulero, M., Garcia-Vallvé, S., and Pujadas, G. (2015)
Molecular Fingerprint Similarity Search in Virtual Screening. *Methods*. Virtual Screening 71, pp. 58–63. DOI: 10.1016/j.ymeth.2014.08.005 (cited on p. 110).

Chen, Q., Zhang, X., and Zhang, R. (2019)
Privacy-Preserving Decision Tree for Epistasis Detection. *Cybersecurity* 2 (7). DOI: 10.1186/s42400-019-0025-z (cited on pp. 32, 33, 54, 117).

Cheng, A. M. (1998)
"The Causes, Impact and Detection of Duplicate Observations". *SAS Conference Proceedings*, p. 11 (cited on p. 99).

Cho, H., Wu, D. J., and Berger, B. (2018)
Secure Genome-Wide Association Analysis using Multiparty Computation. *Nature Biotechnology* 36, pp. 547–551. DOI: 10.1038/nbt.4108 (cited on p. 32).

Cho, Y. M., Ritchie, M. D., Moore, J. H., Park, J. Y., Lee, K.-U., Shin, H. D., Lee, H. K., and Park, K. S. (2004)
Multifactor-Dimensionality Reduction Shows a Two-Locus Interaction Associated with Type 2 Diabetes Mellitus. *Diabetologia* 47.3. DOI: 10.1007/s00125-003-1321-3 (cited on pp. 32, 40, 117).

Choi, S. G., Katz, J., Kumaresan, R., and Zhou, H.-S. (2012)
"On the Security of the "Free-XOR" Technique". *Theory of Cryptography*. Springer, pp. 39–53. DOI: 10.1007/978-3-642-28914-9_3 (cited on p. 19).

Chou, T. and Orlandi, C. (2015)
"The Simplest Protocol for Oblivious Transfer". *Progress in Cryptology – LATINCRYPT 2015*. Vol. 9230. Springer, pp. 40–58. DOI: 10.1007/978-3-319-22174-8_3 (cited on pp. 17, 18).

Christen, P., Schnell, R., Vatsalan, D., and Ranbaduge, T. (2017)
"Efficient Cryptanalysis of Bloom Filters for Privacy-Preserving Record Linkage". *Advances in Knowledge Discovery and Data Mining*. Springer, pp. 628–640 (cited on pp. 81, 87).

Cohen, W. W. (2000)
Data Integration Using Similarity Joins and a Word-based Information Representation Language. *ACM Transactions on Information Syststems* 18.3, pp. 288–321. DOI: 10.1145/352595.352598 (cited on p. 85).

Commission of the European Communities (2008)
Rare Diseases: Europe's challenges. Communication. Commission of the European Communities (cited on p. 79).

Contiero, P., Tittarelli, A., Tagliabue, G., Maghini, A., Fabiano, S., Crosignani, P., and Tessandori, R. (2005)
The EpiLink Record Linkage Software. *Methods of Information in Medicine* 44.01, pp. 66–71. DOI: 10.1055/s-0038-1633924 (cited on pp. 82, 84, 85, 89).

Couteau, G. (2018)
"New Protocols for Secure Equality Test and Comparison". *17. International Conference on Cryptology And Network Security (CANS)*. Springer, pp. 303–320. DOI: 10.1007/978-3-319-93387-0_16z (cited on p. 47).

Dai, W. and Sunar, B. (2016)
"cuHE: A Homomorphic Encryption Accelerator Library". *Cryptography and Information Security in the Balkans*. Springer, pp. 169–186. DOI: 10.1007/978-3-319-29172-7_11 (cited on p. 25).

Damgard, I., Pastro, V., Smart, N. P., and Zakarias, S. (2012)
"Multiparty Computation from Somewhat Homomorphic Encryption". *Advances in Cryptology – CRYPTO*. Vol. 7417, pp. 643–662. DOI: 10.1007/978-3-642-32009-5_38 (cited on p. 52).

De Cristofaro, E., Gasti, P., and Tsudik, G. (2012)
"Fast and Private Computation of Cardinality of Set Intersection and Union". *Cryptology and Network Security (CANS)*. Vol. 7712. Springer, pp. 218–231. DOI: 10.1007/978-3-642-35404-5_17 (cited on pp. 80, 81).

Del Vecchio, D. and Sontag, E. D. (2007)
"Dynamics and Control of Synthetic Bio-molecular Networks". *2007 American Control Conference*, pp. 1577–1588. DOI: 10.1109/ACC.2007.4282302 (cited on p. 35).

Demmler, D., Dessouky, G., Koushanfar, F., Sadeghi, A.-R., Schneider, T., and Zeitouni, S. (2015)
"Automated Synthesis of Optimized Circuits for Secure Computation". *Proceedings of the 2015 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, pp. 1504–1517. DOI: 10.1145/2810103.2813678 (cited on pp. 23, 89).

Demmler, D., Hamacher, K., Schneider, T., and Stammler, S. (2017)
"Privacy-Preserving Whole-Genome Variant Queries". *16. International Conference on Cryptology And Network Security (CANS)*. Vol. 11261. Springer, pp. 1–22. DOI: 10.1007/978-3-030-02641-7_4 (cited on p. 33).

Demmler, D., Schneider, T., and Zohner, M. (2015)
"ABY – A Framework for Efficient Mixed-Protocol Secure Two-Party Computation". *Proceedings 2015 Network and Distributed System Security Symposium (NDSS)*. Internet Society. DOI: 10.14722/ndss.2015.23113 (cited on pp. 22, 23, 40, 48, 58, 75, 82, 89–91, 109).

Denning, D. E., Denning, P. J., and Schwartz, M. D. (1979)
The Tracker: A Threat to Statistical Database Security. *ACM Transactions on Database Systems* 4.1, pp. 76–96. DOI: 10.1145/320064.320069 (cited on p. 10).

Desai, T., Ritchie, F., and Welpton, R. (2006)
Five Safes: Designing Data Access for Research. Working Papers 20161601. University of the West of England, Bristol, p. 27. DOI: 10.13140/RG.2.1.3661.1604 (cited on p. 6).

Dessouky, G., Koushanfar, F., Sadeghi, A.-R., Schneider, T., Zeitouni, S., and Zohner, M. (2017)
"Pushing the Communication Barrier in Secure Computation using Lookup Tables". *24. Network and Distributed System Security Symposium (NDSS)*. Internet Society. DOI: 10.14722/ndss.2017.23097 (cited on p. 43).

Dice, L. R. (1945)
Measures of the Amount of Ecologic Association Between Species. *Ecology* 26.3, pp. 297–302. DOI: 10.2307/1932409 (cited on p. 88).

Dijkstra, E. W. (1982)
"On the Role of Scientific Thought". *Selected writings on computing: a personal perspective*. Springer, pp. 60–66 (cited on p. 65).

Dreier, J. and Kerschbaum, F. (2011)
Practical Secure and Efficient Multiparty Linear Programming Based on Problem Transformation. Pre-Print 108 (cited on p. 59).

Duncan, G. (2011)
Statistical Confidentiality: Principles and Practice. Springer. DOI: 10.1007/978-1-4419-7802-8 (cited on p. 32).

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006)
"Calibrating Noise to Sensitivity in Private Data Analysis". *Theory of Cryptography*. Springer, pp. 265–284. DOI: 10.1007/11681878_14 (cited on pp. 10, 11).

Dwork, C. and Roth, A. (2013)
The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9.3-4. DOI: 10.1561/0400000042 (cited on p. 11).

Edmonds, J. (1965)
Paths, Trees, and Flowers. *Canadian Journal of Mathematics* 17, pp. 449–467. DOI: 10.4153/CJM-1965-045-4 (cited on p. 77).

ElGamal, T. (1985)
"A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". *Advances in Cryptology - CRYPTO*. Springer, pp. 10–18. DOI: 10.1007/3-540-39568-7_2 (cited on p. 25).

Ellison, B. (2014)
A Systematic Review of Kidney Paired Donation: Applying Lessons from Historic and Contemporary Case Studies to Improve the US Model. Working Paper. Wharton Research Scholars (cited on pp. 58, 60).

Erdogan, O. and Cao, P. (2007)
Hash-AV: Fast Virus Signature Scanning by Cache-Resident Filters. *International Journal of Security and Networks* 2.1/2, p. 50. DOI: 10.1504/IJSN.2007.012824 (cited on p. 87).

Erdös, P. and Rényi, A. (1959)
On Random Graphs I. *Publicationes Mathematicae Debrecen* 6, pp. 290–297 (cited on p. 5).

European Data Protection Board (2021)
Recommendations 01/2020 on Measures that Supplement Transfer Tools to Ensure Compliance with the EU Level of Protection of Personal Data. Recommendation. European Data Protection Board (cited on pp. 16, 76).

European Medicines Agency (2017)
External Guidance on the Implementation of the European Medicines Agency Policy on the Publication of Clinical Data for Medicinal Products for Human Use. Technical Report. European Medicines Agency (cited on p. 8).

Eurotransplant (2018a)
"Histocompatibility". *Eurotransplant Manual ver. 4.5*. Chap. 10 (cited on p. 62).

Eurotransplant (2018b)
"Kidney". *Eurotransplant Manual ver. 4.5*. Chap. 4 (cited on p. 62).

Eurotransplant (2021)
Annual Report 2020. URL: https://www.eurotransplant.org/wp-content/uploads/2021/08/ETP_AR2020_opm_LR.pdf (visited on 04/03/2022) (cited on p. 57).

Federal Court of Justice of Germany (2017)
VI ZR 135/13. URL: http://juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?Gericht=bgh%5C&Art=en%5C&nr=78741%5C&pos=0%5C&anz=1 (visited on 05/20/2022) (cited on p. 24).

Fellegi, I. P. and Sunter, A. B. (1969)
A Theory for Record Linkage. *Journal of the American Statistical Association* 64.328, pp. 1183–1210. DOI: 10.1080/01621459.1969.10501049 (cited on pp. 81, 83).

Floyd, R. W. (1962)
Algorithm 97: Shortest Path. *Communications of the ACM* 5.6, p. 345. DOI: 10.1145/367766.368168 (cited on p. 77).

Ford Jr, L. R. (1956)
Network Flow Theory. Technical Report. Rand Corp Santa Monica Ca (cited on p. 77).

Fouque, P.-A., Poupard, G., and Stern, J. (2000)
"Sharing Decryption in the Context of Voting or Lotteries". *International Conference on Financial Cryptography*. Springer, pp. 90–104. DOI: 10.1007/3-540-45472-1_7 (cited on p. 60).

Franz, M., Holzer, A., Katzenbeisser, S., Schallhart, C., and Veith, H. (2014)
"CBMC-GC: An ANSI C Compiler for Secure Two-Party Computations". *Compiler Construction*. Springer, pp. 244–249 (cited on pp. 23, 93).

Ganslandt, T., Boeker, M., Löbe, M., Prasser, F., Schepers, J., Semler, S. C., Thun, S., and Sax, U. (2018)
"Der Kerndatensatz der Medizininformatik-Initiative: Ein Schritt zur Sekundärnutzung von Versorgungsdaten auf nationaler Ebene". *Forum der Medizin-Dokumentation und Medizin-Informatik (MDI)*. Vol. 20. 1, pp. 17–21 (cited on p. 107).

Gentry, C. (2009)
"A Fully Homomorphic Encryption Scheme". PhD Thesis. Stanford, CA: Stanford University (cited on p. 25).

Gkoulalas-Divanis, A., Vatsalan, D., Karapiperis, D., and Kantarcioglu, M. (2021)
Modern Privacy-Preserving Record Linkage Techniques: An Overview. *IEEE Transactions on Information Forensics and Security*. DOI: 10.1109/TIFS.2021.3114026 (cited on p. 81).

GKV-Spitzenverband (2021)
Zahlen und Grafiken - GKV-Spitzenverband. URL: https://www.gkv-spitzenverband.de/service/zahlen_und_grafiken/zahlen_und_grafiken.jsp (visited on 12/20/2021) (cited on p. 80).

Goldreich, O., Micali, S., and Wigderson, A. (1987)
"How to Play ANY Mental Game". *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*. ACM. DOI: 10.1145/28395.28420 (cited on p. 20).

Goldsmith, J. (2000)
How Will The Internet Change Our Health System? *Health Affairs* 19.1, pp. 148–156. DOI: 10.1377/hlthaff.19.1.148 (cited on p. 6).

Gottesman, D. (2000)
Theory of Quantum Secret Sharing. *Physical Review A* 61.4, p. 042311. DOI: 10.1103/PhysRevA.61.042311 (cited on p. 26).

Halevi, S. (2017)
"Homomorphic Encryption". *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*. Springer, pp. 219–276. DOI: 10.1007/978-3-319-57048-8_5 (cited on p. 24).

Halevi, S. and Shoup, V. (2014)
"Algorithms in HElib". *Advances in Cryptology – CRYPTO*. Springer. DOI: 10.1007/978-3-662-44371-2_31 (cited on p. 25).

Hamacher, K., Katzenbeisser, S., **Kussel, T.**, and Stammler, S. (2020)
Genomische Daten und der Datenschutz. *Datenschutz und Datensicherheit (DuD)* 44.2, pp. 87–93. DOI: 10.1007/s11623-020-1229-9 (cited on p. 32).

Haslop, T. (2020)
"Minimal Depth Sorting Networks". Bachelor Thesis. Bremen: University of Bremen (cited on p. 48).

Hastings, M., Hemenway, B., Noble, D., and Zdancewic, S. (2019)
"SoK: General-Purpose Compilers for Secure Multi-Party Computation". *IEEE Symposium on Security and Privacy (S&P)* (cited on p. 23).

Hatzinger, M., Stastny, M., Grützmacher, P., and Sohn, M. (2016)
Die Geschichte der Nierentransplantation. *Der Urologe* 55.10, pp. 1353–1359. DOI: 10.1007/s00120-016-0205-3 (cited on p. 61).

Hazay, C. and Lindell, Y. (2010)
Efficient Secure Two-Party Protocols: Techniques and Constructions. Springer (cited on p. 13).

He, J., Wang, K., Edmondson, A. C., Rader, D. J., Li, C., and Li, M. (2011)
Gene-Based Interaction Analysis by Incorporating External Linkage Disequilibrium Information. *European Journal of Human Genetics* 19.2, pp. 164–172. DOI: 10/b5pcs5 (cited on p. 32).

He, X., Machanavajjhala, A., Flynn, C., and Srivastava, D. (2017)
"Composing Differential Privacy and Secure Computation: A Case Study on Scaling Private Record Linkage". *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, pp. 1389–1406. DOI: 10.1145/3133956.3134030 (cited on pp. 81, 83, 89, 108, 113).

Heidt, C. M., Hund, H., and Fegeler, C. (2021)
A Federated Record Linkage Algorithm for Secure Medical Data Sharing. *German Medical Data Sciences: Bringing Data to Life*, pp. 142–149. DOI: 10.3233/SHTI210062 (cited on pp. 81, 87).

Helminger, L. and Rechberger, C. (2022)
"Multi-Party Computation in the GDPR". *Privacy Symposium 2022 – Data Protection Law International Convergence and Compliance with Innovative Technologies (DPLICIT)*, p. 18 (cited on p. 24).

Henecka, W., Kögl, S., Sadeghi, A.-R., Schneider, T., and Wehrenberg, I. (2010)
"TASTY: Tool for Automating Secure Two-partY computations". *17. ACM Conference on Computer and Communications Security (CCS'10)*. ACM, pp. 451–462. DOI: 10.1145/1866307.1866358 (cited on p. 91).

Hillery, M., Bužek, V., and Berthiaume, A. (1999)
Quantum Secret Sharing. *Physical Review A* 59.3, pp. 1829–1834. DOI: 10.1103/PhysRevA.59.1829 (cited on pp. 26, 27).

Hopcroft, J. E. and Karp, R. M. (1973)
An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing* 2.4, pp. 225–231. DOI: 10.1137/0202019 (cited on p. 77).

Hunt, E. B., Marin, J., and Stone, P. J. (1966)
Experiments in Induction. Academic Press (cited on p. 38).

IEEE Standards Board and American National Standards Institute (1985)
IEEE Standard for Binary Floating-Point Arithmetic. *ANSI/IEEE Std 754-1985*, pp. 1–20. DOI: 10.1109/IEEESTD.1985.82928 (cited on p. 94).

Impagliazzo, R. and Rudich, S. (1989)
"Limits on the Provable Consequences of One-Way Permutations". *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. Association for Computing Machinery, pp. 44–61. DOI: 10.1145/73007.73012 (cited on p. 17).

Inan, A., Kantarcioglu, M., Ghinita, G., and Bertino, E. (2010)
"Private Record Matching using Differential Privacy". *Proceedings of the 13th International Conference on Extending Database Technology (EDBT)*. Association for Computing Machinery, pp. 123–134. DOI: 10.1145/1739041.1739059 (cited on p. 83).

ISBT Working Party for Red Cell Immunogenetics and Blood Group Terminology (2021)
Table of Blood Group Systems. Technical Report v10.0. International Society of Blood Transfusion (cited on p. 64).

Ishai, Y., Kilian, J., Nissim, K., and Petrank, E. (2003)
"Extending Oblivious Transfers Efficiently". *Advances in Cryptology - CRYPTO*. Vol. 2729. Springer, pp. 145–161. DOI: 10.1007/978-3-540-45146-4_9 (cited on p. 17).

Järvinen, K., Leppäkoski, H., Lohan, E.-S., Richter, P., Schneider, T., Tkachenko, O., and Yang, Z. (2019)
"PILOT: Practical Privacy-Preserving Indoor Localization Using OuTsourcing". *IEEE European Symposium on*

Security and Privacy (EuroS&P). IEEE European Symposium on Security and Privacy (EuroS&P), pp. 448–463. DOI: 10/ghz8g7 (cited on pp. 41, 68, 147, 154, 155).

Jones, K. S. (2003)
Privacy: What's Different Now? *Interdisciplinary Science Reviews* 28.4, pp. 287–292. DOI: 10.1179/030801803225008677 (cited on p. 6).

Joos, S., Nettelbeck, D. M., Reil-Held, A., Engelmann, K., Moosmann, A., Eggert, A., Hiddemann, W., Krause, M., et al. (2019)
German Cancer Consortium (DKTK) - A National Consortium for Translational Cancer Research. Universität Freiburg. DOI: 10.1002/1878-0261.12430 (cited on p. 89).

Joung, J.-G. and Fei, Z. (2009)
Identification of microRNA Regulatory Modules in Arabidopsis via a Probabilistic Graphical Model. *Bioinformatics* 25.3, pp. 387–393. DOI: 10.1093/bioinformatics/btn626 (cited on p. 36).

Kales, D., Rechberger, C., Schneider, T., Senker, M., and Weinert, C. (2019)
"Mobile Private Contact Discovery at Scale". *28. USENIX Security Symposium (USENIX)*. USENIX, pp. 1447–1464 (cited on p. 81).

Kamara, S., Mohassel, P., and Raykova, M. (2011)
Outsourcing Multi-Party Computation. Pre-Print 272 (cited on p. 23).

Kauffman, S. A. (1969)
Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets. *Journal of Theoretical Biology* 22.3, pp. 437–467. DOI: 10.1016/0022-5193(69)90015-0 (cited on p. 35).

Keller, M. (2020)
"MP-SPDZ: A Versatile Framework for Multi-Party Computation". *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, pp. 1575–1590. DOI: 10.1145/3372297.3417872 (cited on pp. 24, 61, 74).

Kerckhoffs, A. (1883)
La cryptographie militaire, ou, Des chiffres usités en temps de guerre: avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef. Librairie militaire de L. Baudoin (cited on p. 12).

Kerschbaum, F., Schroepfer, A., Zilli, A., Pibernik, R., Catrina, O., Hoogh, S. de, Schoenmakers, B., Cimato, S., et al. (2011)
Secure Collaborative Supply-Chain Management. *Computer* 44.9, pp. 38–43. DOI: 10.1109/MC.2011.224 (cited on p. 59).

Kim, Y. and Park, T. (2015)
Robust Gene-Gene Interaction Analysis in Genome Wide Association Studies. *PLOS ONE* 10.8, e0135016. DOI: 10/f727qw (cited on p. 32).

Kira, K. and Rendell, L. A. (1992)
"A Practical Approach to Feature Selection". *Machine Learning*. Elsevier (cited on p. 38).

Kirsch, A. and Mitzenmacher, M. (2006)
"Less Hashing, Same Performance: Building a Better Bloom Filter". *Algorithms – ESA 2006*. Springer, pp. 456–467. DOI: 10.1007/11841036_42 (cited on p. 88).

Klar, M. and Kühling, J. (2018)
Datenschutz-Grundverordnung Art. 4 Abs. 1 RN 25. C.H. Beck (cited on p. 8).

Kolesnikov, V. and Kumaresan, R. (2013)
"Improved OT extension for transferring short secrets". *CRYPTO* (cited on pp. 17, 33, 43, 45, 51, 56).

Kolesnikov, V., Kumaresan, R., Rosulek, M., and Trieu, N. (2016)
"Efficient Batched Oblivious PRF with Applications to Private Set Intersection". *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM. DOI: 10.1145/2976749.2978381 (cited on p. 81).

Kolesnikov, V., Sadeghi, A.-R., and Schneider, T. (2009)
"Improved Garbled Circuit Building Blocks and Applications to Auctions and Computing Minima". *8. International Conference on Cryptology And Network Security (CANS)*. Vol. 5888. Springer, pp. 1–20. DOI: 10.1007/978-3-642-10433-6_1 (cited on p. 43).

Kolesnikov, V. and Schneider, T. (2008)
"Improved Garbled Circuit: Free XOR Gates and Applications". *35. International Colloquium on Automata, Languages and Programming (ICALP)*. Vol. 5126. Springer, pp. 486–498. DOI: 10.1007/978-3-540-70583-3_40 (cited on pp. 19, 33, 48, 49, 56).

Koller, D. and Friedman, N. (2009)
Probabilistic Graphical Models: Principles and Techniques. MIT Press (cited on pp. 36, 37).

Kononenko, I. (1994)
"Estimating Attributes: Analysis and Extensions of RELIEF". *Machine Learning: ECML*. Springer, pp. 171–182 (cited on pp. 32, 33, 36, 38, 39).

Kreplak, J., Madoui, M.-A., Cápal, P., Novák, P., Labadie, K., Aubert, G., Bayer, P. E., Gali, K. K., et al. (2019)
A Reference Genome for Pea Provides Insight into Legume Genome Evolution. *Nature Genetics* 51.9, pp. 1411–1422. DOI: 10.1038/s41588-019-0480-1 (cited on p. 35).

Kronfeld, K., Schlangen, M., Schlüchter, D., and Volk, M. (2021)
Datenschutzkonzept Deutsches Mukoviszidose-Register. URL: https://www.muko.info/fileadmin/user_upload/angebote/qualitaetsmanagement/register/datenschutz/datenschutzkonzept.pdf (visited on 03/10/2022) (cited on p. 89).

Kuhn, H. W. (1955)
The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly* 2.1-2, pp. 83–97. DOI: 10.1002/nav.3800020109 (cited on p. 77).

Kussel, T., Brenner, T., Tremper, G., Schepers, J., Lablans, M., and Hamacher, K. (2022)
Record Linkage-based Patient Intersection Cardinality for Rare Disease Studies using Mainzelliste and Secure Multi-Party Computation. *BMC Journal of Translational Medicine*. In Review, Pre-Print: https://www.researchsquare.com/article/rs-1486673/v1. DOI: 10.21203/rs.3.rs-1486673/v1 (cited on pp. 95, 107).

Kuzu, M., Kantarcioglu, M., Durham, E., and Malin, B. (2011)
"A Constraint Satisfaction Cryptanalysis of Bloom Filters in Private Record Linkage". *Privacy Enhancing Technologies (PETS)*. Springer, pp. 226–245 (cited on p. 81).

Lablans, M., Borg, A., and Ückert, F. (2015)
A RESTful Interface to Pseudonymization Services in Modern Web Applications. *BMC Medical Informatics and Decision Making* 15, p. 2. DOI: 10.1186/s12911-014-0123-5 (cited on pp. 80, 89).

Lablans, M., Schmidt, E., and Ückert, F. (2018)
An Architecture for Translational Cancer Research As Exemplified by the German Cancer Consortium. *JCO Clinical Cancer Informatics* 2, pp. 1–8. DOI: 10.1200/CCI.17.00062 (cited on pp. 81, 87, 99).

Larrañaga, P., Inza, I., and Flores, J. L. (2005)
A Guide to the Literature of Inferring Genetic Networks by Probabilistic Graphical Models. John Wiley & Sons. Chap. 13 (cited on p. 35).

Laud, P. and Pankova, A. (2018)
Privacy-Preserving Record Linkage in Large Databases Using Secure Multiparty Computation. *BMC Medical Genomics* 11.4, p. 84. DOI: 10.1186/s12920-018-0400-8 (cited on p. 81).

Lazrig, I., Ong, T. C., Ray, I., Ray, I., Jiang, X., and Vaidya, J. (2018)
"Privacy Preserving Probabilistic Record Linkage Without Trusted Third Party". *16th Annual Conference on Privacy, Security and Trust (PST)*, pp. 1–10. DOI: 10.1109/pst.2018.8514192 (cited on pp. 81–83, 99–102).

Le, T. T., Simmons, W. K., Misaki, M., Bodurka, J., White, B. C., Savitz, J., and McKinney, B. A. (2017)
Differential Privacy-based Evaporative Cooling Feature Selection and Classification with Relief-F and Random Forests. *Bioinformatics* 33.18, pp. 2906–2913. DOI: 10.1093/bioinformatics/btx298 (cited on pp. 32, 117).

Lee, S., Son, D., Kim, Y., Yu, W., and Park, T. (2018)
Unified Cox Model Based Multifactor Dimensionality Reduction Method for Gene-Gene Interaction Analysis of the Survival Phenotype. *BioData Mining* 11 (27). DOI: 10.1186/s13040-018-0189-1 (cited on p. 32).

Leeaphorn, N., Pena, J. R. A., Thamcharoen, N., Khankin, E. V., Pavlakis, M., and Cardarelli, F. (2018)
HLA-DQ Mismatching and Kidney Transplant Outcomes. *Clinical Journal of the American Society of Nephrology* 13.5, pp. 763–771. DOI: 10.2215/CJN.10860917 (cited on pp. 62, 63).

Leeson, S. and Desai, S. P. (2015)
Medical and Ethical Challenges During the First Successful Human Kidney Transplantation in 1954 at Peter Bent Brigham Hospital, Boston. *Anesthesia & Analgesia* 120.1, pp. 239–245. DOI: 10.1213/ANE.0000000000000521 (cited on p. 61).

Lefaucheur, C., Loupy, A., Hill, G. S., Andrade, J., Nochy, D., Antoine, C., Gautreau, C., Charron, D., et al. (2010)
Preexisting Donor-Specific HLA Antibodies Predict Outcome in Kidney Transplantation. *Journal of the American Society of Nephrology* 21.8, pp. 1398–1406. DOI: 10.1681/ASN.2009101065 (cited on p. 62).

Li, J. and Atallah, M. J. (2006)
"Secure and Private Collaborative Linear Programming". *2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 1–8. DOI: 10.1109/COLCOM.2006.361848 (cited on p. 59).

Li, L., Cheng, W.-Y., Glicksberg, B. S., Gottesman, O., Tamler, R., Chen, R., Bottinger, E. P., and Dudley, J. T. (2015)
Identification of Type 2 Diabetes Subgroups Through Topological Analysis of Patient Similarity. *Science Translational Medicine* 7.311, 311ra174–311ra174. DOI: 10.1126/scitranslmed.aaa9364 (cited on p. 111).

Li, N., Li, T., and Venkatasubramanian, S. (2007)
"t-Closeness: Privacy Beyond k-Anonymity and l-Diversity". *IEEE 23rd International Conference on Data Engineering (ICDE)*, pp. 106–115. DOI: 10.1109/ICDE.2007.367856 (cited on p. 9).

Lim, W. H., Chadban, S. J., Clayton, P., Budgeon, C. A., Murray, K., Campbell, S. B., Cohney, S., Russ, G. R., et al. (2012)
Human Leukocyte Antigen Mismatches Associated with Increased Risk of Rejection, Graft Failure, and

Death Independent of Initial Immunosuppression in Renal Transplant Recipients. *Clinical Transplantation* 26.4, E428–E437. DOI: https://doi.org/10.1111/j.1399-0012.2012.01654.x (cited on p. 63).

Lindell, Y. (2020)
Secure Multiparty Computation. *Communications of the ACM* 64.1, pp. 86–96. DOI: 10.1145/3387108 (cited on p. 15).

Liu, J., Sun, K., Bai, Y., Zhang, W., Wang, X., Wang, Y., Wang, H., Chen, J., et al. (2009)
Association of Three-Gene Interaction among MTHFR, ALOX5AP and NOTCH3 with Thrombotic Stroke: A Multicenter Case–Control Study. *Human Genetics* 125.5. DOI: 10.1007/s00439-009-0659-0 (cited on pp. 32, 40, 117).

Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007)
L-Diversity: Privacy Beyond k-Anonymity. *ACM Transactions on Knowledge Discovery from Data* 1.1, 3–es. DOI: 10.1145/1217299.1217302 (cited on p. 9).

Malkhi, D., Nisan, N., Pinkas, B., and Sella, Y. (2004)
"Fairplay — A Secure Two-Party Computation System", p. 17 (cited on p. 23).

Mazloom, S. and Gordon, S. D. (2018)
"Secure Computation with Differentially Private Access Patterns". *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, pp. 490–507. DOI: 10.1145/3243734.3243851 (cited on p. 59).

Mazloom, S., Le, P. H., Ranellucci, S., and Gordon, S. D. (2020)
"Secure Parallel Computation on National Scale Volumes of Data". *29th USENIX Security Symposium (USENIX)*. USENIX Association, pp. 2487–2504 (cited on p. 59).

Medizin, N. N. G., Büttner, R., Wolf, J., and Kron, A. (2019)
Das nationale Netzwerk Genomische Medizin (nNGM): Modell für eine innovative Diagnostik und Therapie von Lungenkrebs im Spannungsfeld eines öffentlichen Versorgungsauftrages. *Der Pathologe* 40.3, pp. 276–280. DOI: 10.1007/s00292-019-0605-4 (cited on p. 89).

Meng, Y., Groth, S., Quinn, J. R., Bisognano, J., and Wu, T. T. (2017)
An Exploration of Gene-Gene Interactions and Their Effects on Hypertension. *International Journal of Genomics* 2017, p. 7208318. DOI: 10/gbjxns (cited on pp. 32, 40, 117).

Micciancio, D. and Goldwasser, S. (2002)
Complexity of Lattice Problems: A Cryptographic Perspective. Springer (cited on p. 5).

Miller, A. J., Kiberd, B. A., Alwayn, I. P., Odutayo, A., and Tennankore, K. K. (2017)
Donor-Recipient Weight and Sex Mismatch and the Risk of Graft Loss in Renal Transplantation. *Clinical Journal of the American Society of Nephrology* 12.4, pp. 669–676. DOI: 10.2215/CJN.07660716 (cited on pp. 64, 154).

Monod, J., Changeux, J.-P., and Jacob, F. (1963)
Allosteric Proteins and Cellular Control Systems. *Journal of Molecular Biology* 6.4, pp. 306–329. DOI: 10.1016/S0022-2836(63)80091-1 (cited on p. 35).

Moore, J. H. and White, B. C. (2007)
"Tuning ReliefF for Genome-Wide Genetic Analysis". *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO)*. Springer, pp. 166–175. DOI: 10.1007/978-3-540-71783-6_16 (cited on pp. 33, 36, 39, 52, 55).

Murphy, D., McCulloch, C. E., Lin, F., Banerjee, T., Bragg-Gresham, J. L., Eberhardt, M. S., Morgenstern, H., Pavkov, M. E., et al. (2016)

Trends in Prevalence of Chronic Kidney Disease in the United States. *Annals of Internal Medicine* 165.7, pp. 473–481. DOI: 10.7326/M16-0273 (cited on p. 57).

Naccache, D. (2011)
"Standard Model". *Encyclopedia of Cryptography and Security*. Springer, pp. 1253–1253. DOI: 10.1007/978-1-4419-5906-5_518 (cited on p. 17).

Naor, M., Pinkas, B., and Sumner, R. (1999)
"Privacy Preserving Auctions and Mechanism Design". *Proceedings of the 1st ACM Conference on Electronic Commerce (EC)*. ACM Press, pp. 129–139. DOI: 10.1145/336992.337028 (cited on p. 19).

Narayanan, A. and Shmatikov, V. (2007)
How To Break Anonymity of the Netflix Prize Dataset. *arXiv:cs/0610105* (cited on p. 10).

Naveed, M., Ayday, E., Clayton, E. W., Fellay, J., Gunter, C. A., Hubaux, J.-P., Malin, B. A., and Wang, X. (2015)
Privacy in the Genomic Era. *ACM Computing Surveys* 48.1. DOI: 10.1145/2767007 (cited on pp. 32, 117).

Nayak, K., Wang, X. S., Ioannidis, S., Weinsberg, U., Taft, N., and Shi, E. (2015)
"GraphSC: Parallel Secure Computation Made Easy". *2015 IEEE Symposium on Security and Privacy (S&P)*, pp. 377–394. DOI: 10.1109/SP.2015.30 (cited on p. 59).

Newman, M. (2018)
Networks. Oxford University Press (cited on p. 4).

Newton-Cheh, C., Johnson, T., Gateva, V., Tobin, M. D., Bochud, M., Coin, L., Najjar, S. S., Zhao, J. H., et al. (2009)
Genome-Wide Association Study Identifies Eight Loci Associated with Blood Pressure. *Nature Genetics* 41.6 (6). DOI: 10.1038/ng.361 (cited on p. 35).

Nguyen, M. C. (2021)
"Evaluation of HLA Typing Data and Transplant Outcome in Pediatric Renal Transplantation". PhD thesis. Medizinische Universität Wien (cited on pp. 62, 63).

Nietert, M. M., Vinhoven, L., Auer, F., Hafkemeyer, S., and Stanke, F. (2021)
Comprehensive Analysis of Chemical Structures That Have Been Tested as CFTR Activating Substances in a Publicly Available Database CandActCFTR. *Frontiers in Pharmacology* 12. DOI: 10.3389/fphar.2021.689205 (cited on p. 110).

Ntokou, I.-S. A., Iniotaki, A. G., Kontou, E. N., Darema, M. N., Apostolaki, M. D., Kostakis, A. G., and Boletis, J. N. (2011)
Long-Term Follow Up for Anti-HLA Donor Specific Antibodies Postrenal Transplantation: High Immunogenicity of HLA Class II Graft Molecules. *Transplant International* 24.11, pp. 1084–1093. DOI: 10.1111/j.1432-2277.2011.01312.x (cited on p. 62).

O'Neil, P., Cheng, E., Gawlick, D., and O'Neil, E. (1996)
The Log-Structured Merge-Tree (LSM-Tree). *Acta Informatica* 33.4, pp. 351–385. DOI: 10.1007/s002360050048 (cited on p. 87).

Opelz, G. and Döhler, B. (2012)
Association of HLA Mismatch with Death with a Functioning Graft after Kidney Transplantation: A Collaborative Transplant Study Report. *American Journal of Transplantation* 12.11, pp. 3031–3038. DOI: https://doi.org/10.1111/j.1600-6143.2012.04226.x (cited on p. 63).

Opelz, G. (1997)
Impact of HLA Compatibility on Survival of Kidney Transplants from Unrelated Live Donors. *Transplantation* 64 (10), pp. 1473–1475. DOI: 10.1097/00007890-199711270-00017 (cited on p. 63).

Oswald, M. (2013)
ISB1523: Anonymisation Standard for Publishing Health and Social Care Data. Technical Report. National Health Serdice (NHS) (cited on p. 8).

Ozaki, K., Ohnishi, Y., Iida, A., Sekine, A., Yamada, R., Tsunoda, T., Sato, H., Sato, H., et al. (2002)
Functional SNPs in the Lymphotoxin-αGene That Are Associated with Susceptibility to Myocardial Infarction. *Nature Genetics* 32.4 (4). DOI: 10.1038/ng1047 (cited on p. 35).

Paillier, P. (1999)
"Public-Key Cryptosystems Based on Composite Degree Residuosity Classes". *Advances in Cryptology — EUROCRYPT*. Springer, pp. 223–238. DOI: 10.1007/3-540-48910-X_16 (cited on p. 25).

Pansart, L., Cambazard, H., Catusse, N., and Stauffer, G. (2014)
"Kidney Exchange Problem: Models and Algorithms". *19ème congrès annuel de la société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF)* (cited on p. 60).

Pape, U. and Conradt, D. (1980)
"Maximales Matching in Graphen". *Ausgewählte Operations Research Software in FORTRAN* (cited on p. 61).

Parimbelli, E., Marini, S., Sacchi, L., and Bellazzi, R. (2018)
Patient Similarity for Precision Medicine: A Systematic Review. *Journal of Biomedical Informatics* 83, pp. 87–96. DOI: 10.1016/j.jbi.2018.06.001 (cited on p. 111).

Pearl, J. (1988)
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (cited on p. 36).

Peixoto, T. P. and Gauvin, L. (2018)
Change Points, Memory and Epidemic Spreading in Temporal Networks. *Nature Scientific Reports* 8.1, p. 15511. DOI: 10.1038/s41598-018-33313-1 (cited on p. 2).

Pelizzola, A. (2005)
Cluster Variation Method in Statistical Physics and Probabilistic Graphical Models. *Journal of Physics A: Mathematical and General* 38.33, R309 (cited on p. 36).

Pinkas, B., Schneider, T., Weinert, C., and Wieder, U. (2018)
"Efficient Circuit-Based PSI via Cuckoo Hashing". *37. Advances in Cryptology – EUROCRYPT*. Vol. 10822. Springer, pp. 125–157. DOI: 10.1007/978-3-319-78372-7_5 (cited on p. 80).

Prokosch, H.-U., Acker, T., Bernarding, J., Binder, H., Boeker, M., Börries, M., Daumke, P., Ganslandt, T., et al. (2018)
MIRACUM: Medical Informatics in Research and Care in University Medicine: A large data sharing network to enhance translational research and medical care. *Methods of Information in Medicine*. DOI: 10.3414/me17-02-0025 (cited on p. 89).

Rabin, M. O. (1981)
How to Exchange Secrets with Oblivious Transfer, p. 26 (cited on p. 16).

Rathee, D., Rathee, M., Kumar, N., Chandran, N., Gupta, D., Rastogi, A., and Sharma, R. (2020)
"CrypTFlow2: Practical 2-Party Secure Inference". *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, pp. 325–342. DOI: 10.1145/3372297.3417274 (cited on pp. 33, 43, 47, 56, 149).

Raynal, F., Bedrune, J.-B., Bouyat, J., Campana, G., and Zimmer, D. (2017)
OpenVPN 2.4.0 Security Assessment. Technical Report 17-03-284-REP. OSTIF (cited on p. 99).

Rescorla, E. (2008)
The Transport Layer Security (TLS) Protocol Version 1.2. URL: https://tools.ietf.org/html/rfc5246 (visited on 06/24/2019) (cited on p. 95).

Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., and Moore, J. H. (2001)
Multifactor-Dimensionality Reduction Reveals High-Order Interactions among Estrogen-Metabolism Genes in Sporadic Breast Cancer. *The American Journal of Human Genetics* 69.1. DOI: 10.1086/321276 (cited on pp. 32, 33, 36, 40, 41).

Rivest, R. L., Adleman, L., Dertouzos, M. L., et al. (1978)
On Data Banks and Privacy Homomorphisms. *Foundations of secure computation* 4.11, pp. 169–180 (cited on p. 25).

Rohde, F., Franke, M., Sehili, Z., Lablans, M., and Rahm, E. (2021)
Optimization of the Mainzelliste Software for Fast Privacy-Preserving Record Linkage. *Journal of Translational Medicine* 19.1, p. 33. DOI: 10.1186/s12967-020-02678-1 (cited on pp. 81, 89).

Rosulek, M. and Roy, L. (2021)
"Three Halves Make a Whole? Beating the Half-Gates Lower Bound for Garbled Circuits". *Advances in Cryptology – CRYPTO*. Springer, pp. 94–124. DOI: 10.1007/978-3-030-84242-0_5 (cited on pp. 19, 33, 40, 48, 51, 56, 117).

Santos, C., Costa, R., Malheiro, J., Pedroso, S., Almeida, M., Martins, L. S., Dias, L., Tafulo, S., et al. (2014)
Kidney Transplantation Across a Positive Crossmatch: A Single-Center Experience. 46.6, pp. 1705–1709. DOI: 10.1016/j.transproceed.2014.05.012 (cited on p. 62).

Sariyar, M., Borg, A., and Pommerening, K. (2011)
Controlling False Match Rates in Record Linkage Using Extreme Value Theory. *Journal of Biomedical Informatics* 44.4, pp. 648–654. DOI: 10.1016/j.jbi.2011.02.008 (cited on p. 103).

Schmidt, M., Hamacher, K., Reinhardt, F., Lotz, T. S., Groher, F., Suess, B., and Jager, S. (2020)
SICOR: Subgraph Isomorphism Comparison of RNA Secondary Structures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 17.6, pp. 2189–2195. DOI: 10.1109/TCBB.2019.2926711 (cited on pp. 83, 118).

Schneider, T. and Tkachenko, O. (2019)
"EPISODE: Efficient Privacy-PreservIng Similar Sequence Queries on Outsourced Genomic DatabasEs". *14. ACM ASIA Conference on Computer and Communications Security (ASIACCS)*. ACM, pp. 315–327. DOI: 10.1145/3321705.3329800 (cited on pp. 33, 48, 51, 149).

Schneider, T. and Zohner, M. (2013)
"GMW vs. Yao? Efficient Secure Two-Party Computation with Low Depth Circuits". *Financial Cryptography and Data Security (FC)*. Springer, pp. 275–292. DOI: 10.1007/978-3-642-39884-1_23 (cited on pp. 20, 48).

Schnell, R., Bachteler, T., and Reiher, J. (2009)
Privacy-Preserving Record Linkage using Bloom filters. *BMC Medical Informatics and Decision Making* 9, pp. 1–11. DOI: 10.1186/1472-6947-9-41 (cited on pp. 81, 82, 88).

Schnell, R. and Borgs, C. (2018)
Protecting Record Linkage Identifiers Using a Language Model for Patient Names. *German Medical Data Sciences: A Learning Healthcare System*, pp. 91–95. DOI: 10.3233/978-1-61499-896-9-91 (cited on p. 81).

Schroepfer, A., Kerschbaum, F., and Mueller, G. (2010)
"Li - An Intermediate Language for Mixed-Protocol Secure Computation". *IEEE 35th Annual Computer Software and Applications Conference (COMPSAC)*, pp. 298–307. DOI: 10.1109/COMPSAC.2011.46 (cited on p. 59).

Schuster, S. C., Miller, W., Ratan, A., Tomsho, L. P., Giardine, B., Kasson, L. R., Harris, R. S., Petersen, D. C., et al. (2010)
Complete Khoisan and Bantu Genomes from Southern Africa. *Nature* 463.7283 (7283). DOI: 10.1038/nature08795 (cited on p. 35).

Microsoft Research (2022)
Microsoft SEAL (release 4.0). https://github.com/Microsoft/SEAL (cited on p. 25).

Sheridan, R. P. and Kearsley, S. K. (2002)
Why Do We need So Many Chemical Similarity Search Methods? *Drug Discovery Today* 7.17, pp. 903–911. DOI: 10.1016/S1359-6446(02)02411-X (cited on p. 110).

Simmons, S. and Berger, B. (2016)
Realizing Privacy Preserving Genome-Wide Association Studies. *Bioinformatics* 32.9, pp. 1293–1300. DOI: 10.1093/bioinformatics/btw009 (cited on p. 12).

Smyth, P. and Goodman, R. M. (1990)
"Rule Induction using Information Theory". *Knowledge Discovery in Databases*. MIT Press. Chap. 9 (cited on p. 38).

Songhori, E. M., Hussain, S. U., Sadeghi, A.-R., Schneider, T., and Koushanfar, F. (2015)
"TinyGarble: Highly Compressed and Scalable Sequential Garbled Circuits". *IEEE Symposium on Security and Privacy (S&P)*, pp. 411–428. DOI: 10.1109/SP.2015.32 (cited on pp. 13, 23).

Stammler, S., Katzenbeisser, S., and Hamacher, K. (2016)
"Correcting Finite Sampling Issues in Entropy l-diversity". *Privacy in Statistical Databases*. Springer, pp. 135–146. DOI: 10.1007/978-3-319-45381-1_11 (cited on p. 9).

Stammler, S., Kussel, T., Schoppmann, P., Stampe, F., Tremper, G., Katzenbeisser, S., Hamacher, K., and Lablans, M. (2020)
Mainzelliste SecureEpiLinker (MainSEL): Privacy-Preserving Record Linkage Using Secure Multi-Party Computation. *Bioinformatics* 38 (6), pp. 1657–1668. DOI: 10.1093/bioinformatics/btaa764 (cited on pp. 96, 99).

Stinson, D. R. (2005)
Cryptography: Theory and Practice. Chapman and Hall/CRC (cited on p. 12).

Sung, Y. C. (2007)
The HLA System: Genetics, Immunology, Clinical Testing, and Clinical Implications. *Yonsei Medical Journal* 48 (1), pp. 11–23. DOI: 10.3349/ymj.2007.48.1.11 (cited on p. 62).

Sweeney, L. (2000)
Simple Demographics Often Identify People Uniquely. *Health (San Francisco)* 671.2000, pp. 1–34 (cited on p. 8).

Sweeney, L. (2002)
k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05, pp. 557–570. DOI: 10.1142/s0218488502001648 (cited on p. 8).

Tanimoto, T. T. (1958)
Elementary Mathematical Theory of Classification and Prediction (cited on p. 110).

The Court of Justice of the European Union (2016)
C-582/14. URL: http://curia.europa.eu/juris/documents.jsf?num=C-582/14 (visited on 05/20/2022) (cited on p. 24).

Thurlow, J. S., Joshi, M., Yan, G., Norris, K. C., Agodoa, L. Y., Yuan, C. M., and Nee, R. (2021)
Global Epidemiology of End-Stage Kidney Disease and Disparities in Kidney Replacement Therapy. *American Journal of Nephrology* 52.2, pp. 98–107. DOI: 10.1159/000514550 (cited on p. 57).

Tkachenko, O., Weinert, C., Schneider, T., and Hamacher, K. (2018)
"Large-Scale Privacy-Preserving Statistical Computations for Distributed Genome-Wide Association Studies". *13. ACM ASIA Conference on Computer and Communications Security (ASIACCS)*. ACM, pp. 221–235. DOI: 10.1145/3196494.3196541 (cited on p. 32).

Toft, T. (2009)
"Solving Linear Programs Using Multiparty Computation". *Financial Cryptography and Data Security*. Vol. 5628. Springer, pp. 90–107. DOI: 10.1007/978-3-642-03549-4_6 (cited on p. 59).

Vadhan, S. (2011)
"Computational Complexity". *Encyclopedia of Cryptography and Security*. Springer, pp. 235–240. DOI: 10.1007/978-1-4419-5906-5_442 (cited on p. 15).

Vatsalan, D., Christen, P., and Verykios, V. S. (2013)
A Taxonomy of Privacy-preserving Record Linkage Techniques. *Inf. Syst.* 38.6, pp. 946–969. DOI: 10.1016/j.is.2012.11.005 (cited on p. 81).

Vatsalan, D., Sehili, Z., Christen, P., and Rahm, E. (2017)
"Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges". *Handbook of Big Data Technologies*. Springer, pp. 851–895. DOI: 10.1007/978-3-319-49340-4_25 (cited on p. 81).

Waiser, J., Schreiber, M., Budde, K., Fritsch, L., Böhler, T., Hause, I., and Neumayer, H.-H. (2000)
Age-matching in Renal Transplantation. *Nephrology Dialysis Transplantation* 15.5, pp. 696–700. DOI: 10.1093/ndt/15.5.696 (cited on pp. 64, 153, 154).

Wang, M. H., Cordell, H. J., and Van Steen, K. (2019)
Statistical Methods for Genome-Wide Association Studies. *Seminars in Cancer Biology* 55. DOI: 10.1016/j.semcancer.2018.04.008 (cited on p. 35).

Wang, M., Ji, Z., Wang, S., Kim, J., Yang, H., Jiang, X., and Ohno-Machado, L. (2017)
Mechanisms to Protect the Privacy of Families when using the Transmission Disequilibrium Test in Genome-Wide Association Studies. *Bioinformatics* 33.23, pp. 3716–3725. DOI: 10.1093/bioinformatics/btx470 (cited on p. 12).

Warshall, S. (1962)
A Theorem on Boolean Matrices. *Journal of the ACM* 9.1, pp. 11–12. DOI: 10.1145/321105.321107 (cited on p. 77).

Watts, D. J. and Strogatz, S. H. (1998)
Collective Dynamics of "Small-World" Networks. *Nature* 393.6684, pp. 440–442. DOI: 10.1038/30918 (cited on p. 5).

Weerd, A. E. de and Betjes, M. G. H. (2018)
ABO-Incompatible Kidney Transplant Outcomes: A Meta-Analysis. *Clinical Journal of the American Society of Nephrology* 13.8, pp. 1234–1243. DOI: 10.2215/CJN.00540118 (cited on p. 64).

Wiesner, S. (1983)
Conjugate Coding. *ACM SIGACT News* 15.1, pp. 78–88. DOI: 10.1145/1008908.1008920 (cited on pp. 16, 25).

Wirth, F., Kussel, T., Hamacher, K., and Prasser, F. (2021)
A Simple but Powerful No-Code Approach to Practical Secure Multiparty Computing in Medical Research: Development Study. *BMC Bioinformatics*. In Review (cited on p. 22).

Wong, W. (2001)
Stunnel: SSLing Internet Services Easily. Technical Report. SANS Institute (cited on p. 99).

Worring, A., Mayer, B. E., and Hamacher, K. (2021)
"Genetic Algorithm Niching by (Quasi-)Infinite Memory". *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Association for Computing Machinery, pp. 296–304. DOI: 10.1145/3449639.3459365 (cited on p. 87).

Xu, T., Le, T. D., Liu, L., Wang, R., Sun, B., and Li, J. (2016)
Identifying Cancer Subtypes from miRNA-TF-mRNA Regulatory Networks and Expression Data. *PLOS ONE* 11.4, e0152792. DOI: 10.1371/journal.pone.0152792 (cited on p. 111).

Yang, L., Qu, B., Xia, X., Kuang, Y., Li, J., Fan, K., Guo, H., Zheng, H., et al. (2017)
Impact of Interaction between the G870A and EFEMP1 Gene Polymorphism on Glioma Risk in Chinese Han Population. *Oncotarget* 8.23, pp. 37561–37567. DOI: 10/f9xkd5 (cited on pp. 32, 117).

Yao, A. C.-C. (1986)
"How to Generate and Exchange Secrets". *27th Annual Symposium on Foundations of Computer Science (SFCS)*, pp. 162–167. DOI: 10.1109/sfcs.1986.25 (cited on p. 17).

Zabicki, R. and Ellis, S. R. (2017)
"Penetration Testing". *Computer and Information Security Handbook (Third Edition)*. Morgan Kaufmann. Chap. 75, pp. 1031–1038. DOI: 10.1016/B978-0-12-803843-7.00075-2 (cited on p. 81).

Zahur, S., Rosulek, M., and Evans, D. (2015)
"Two Halves Make a Whole". *Advances in Cryptology - EUROCRYPT*. Springer, pp. 220–250. DOI: 10.1007/978-3-662-46803-6_8 (cited on pp. 19, 51).

Zhang, H., Zheng, W., Hua, L., Wang, Y., Li, J., Bai, H., Wang, S., Du, M., et al. (2017)
Interaction between PPAR γ and SORL1 Gene with Late-Onset Alzheimer's Disease in Chinese Han Population. *Oncotarget* 8.29, pp. 48313–48320. DOI: 10/f94x4z (cited on pp. 32, 117).

Zhang, Z.-j., Li, Y., and Man, Z.-x. (2005)
Multiparty Quantum Secret Sharing. *Physical Review A* 71.4, p. 044301. DOI: 10.1103/PhysRevA.71.044301 (cited on p. 26).

Zhoua, J.-Y., Chenga, J., Huanga, H.-F., Shen, Y., Jiang, Y., and Chen, J.-H. (2013)
The Effect of Donor-Recipient Sex Mismatch on Short- and Long-term Graft Survival in Kidney Transplantation: A Systematic Review and Meta-Analysis. *Clinical Transplantation* 27.5, pp. 764–771. DOI: https://doi.org/10.1111/ctr.12191 (cited on pp. 64, 154).

Part IV

Appendix

Chapter A
Abbreviations

3HG	Three-Halves Garbling
AGT	Arithmetic Greater Than
BFS	Breadth-First Search
CI/CD	Continuous Integration/Continuous Deployment
CORD_MI	Collaboration On Rare Diseases
CSV	Comma Saparated Value
DAG	Directed Acyclic Graph
DNA	desoxyribonucleid acid
DP	Differential Privacy
EA	Epistasis Analysis
ETL	Extract, Transform, Load
FHE	Fully Homomorphic Encryption
FHIR	Fast Healthcare Interoperability Resources
GC	Yao's Garbled Circuits
GDPR	General Data Protection Regulation
GHZ	Greenberger-Horne-Zeilinger
GT	Greater Than
GWAS	Genome-Wide Association Study
HE	Homomorphic Encryption
HIPAA	Health Insurance Portability and Accountability Act
HL7	Health Level Seven International
HLA	Human Leucocyte Antigens
HMAC	Hash-based Message Authentication Codes
i.i.d.	independent and identically distributed
IDAT	Identifying Personal Data
ILP	Integer Linear Programming
IRB	Institutional Review Board
KEP	Kidney Exchange Problem
kNN	k Nearest Neighbors
LAN	Local Area Network
LID	Linkage ID
LS	Linkage Service
LSH	Locality-Sensitivity Hashing
LWE	Learning with Errors
MainSEL	Mainzelliste Secure EpiLinker
MDAT	Medical Data
MDR	Multifactor Dimensionality Reduction
MI-I	(German) Medical Informatics Initiative
MPC	Secure Multi-Party Computation
mRNA	messenger ribonucleid acid

MSB	Most Significant Bit
MT	Multiplication Triples
ODEs	ordinary differential equations
ORAM	Oblivious Random Access Memory
OT	Oblivious Transfer
OTP	One-Time Pad
PDF	Probability Density Function
PEA	Practical Private Epistasis Analysis using MPC
PEA	Private Epistasis Analysis
PFS	Private Feature Selection
PGM	Probabilistic Graphical Model
PGMs	Probabilistic Graphical Models
PMDR	Private Multifactor Dimensionality Reduction
PPKEP	Privacy-Preserving Kidney Exchange Protocol
PPRL	Privacy-Preserving Record Linkage
PRelief-F	Private Relief-F
PRNG	Pseudorandom Number Generator
PSI	Private Set Intersection
PSI-C	Private Set Intersection Cardinality
PTuRF	Private Tuned Relief-F
RNA	ribonucleid acid
SDC	Statistical Disclosure Control
SIMD	Single Instruction Multiple Data
SNP	Single Nucleotide Polymorphism
SNPs	Single Nucleotide Polymorphisms
SPIKE	Secure and Private Investigation of the Kidney Exchange problem
SSS	Shamir's Secret Sharing
SWHE	Somewhat Homomorphic Encryption
TTP	Trusted Third Party
TuRF	Tuned Relief-F
UAC	Use and Access Committee
WAN	Wide Area Network

Chapter B
Experimental Network Settings

To gain significance for a wide variety of application settings, the experimental evaluations of the protocols in this dissertation were designed to have high informative value for deployment in a *Wide Area Network* (WAN) and a *Local Area Network* (LAN) setting. The rationale behind the parameter choice is—albeit slightly differing between the protocols—always the same, hence generally described in this chapter. The chosen network performance parameters, as well as additional evaluated network settings are detailed in the respective chapters.

The two common network settings are:

WAN The WAN scenario models a range of applications. The similarity is, that all computational parties are geographically spread. Most larger medical institutions are connected via high-bandwidth network connections—for example the German research network ("Deutsches Forschungsnetz", DFN), a research exclusive high-performance carrier network. As we hope *Secure Multi-Party Computation* (MPC) protocols to reduce the legal effort required to participate in a computation, smaller, local hospitals and medical practices might participate directly in those computations. For these institutions residential internet access with reduced bandwidth is realistic. Typical cross-European network latencies are around 13 ms and across North America 30 ms[1]. Higher latencies might be assumed, as present proxy systems and firewalls, as well as packet loss and connection unreliabilities might increase the network latency. Hence, the assumed *WAN* network setting consists of a connection between 100 Mbit/s and 10 Gbit/s bandwidth and a conservatively restricted latency between 50 ms and 100 ms simulated by using the `tc` commandline tool[2].

LAN This setting is relevant for large, institutions interconnected in close distance and high-performance network connections or multiple medical institutions perform the protocol in an outsourced model using two computation parties. As the computation parties can not infer any sensitive information—non-collusion assumed—servers from, for example, competing cloud computing providers co-located at the same internet exchange point can be chosen, thus having a high-bandwidth connection with no additional network delay. The benchmarks in this *LAN* setting use between 1 Gbit/s and 10 Gbit/s bandwidth network with an average latency of under 1 ms.

[1] https://www.verizon.com/business/terms/latency Accessed: 2022-04-20

[2] https://man7.org/linux/man-pages/man8/tc.8.html

CHAPTER C
Epistasis Analysis Supplementary Material

C.1 PRIVATE TUNED RELIEF-F FEATURE SELECTION

PEA's *Private Tuned Relief-F* (PTuRF) protocol (Protocol C.1) works similar to *Private Relief-F* (PRelief-F) (Protocol 3.3). The key difference is, that in each iteration the "weakest" features—the features with the lowest weight—are pruned. This speeds up subsequent iterations, as the number of features is reduced and decreases the influence of noisy attributes. However, this filtering requires a partial sort—implemented following the `kNN` design of JÄRVINEN ET AL.[1] like in PRelief-F—in *every* iteration. Additionally, the removal of features potentially introduces an order and with that a sampling bias. This is mitigated by randomly shuffling the records before feature selection.

[1] Järvinen et al. (2019)

C.2 BATCH OPERATION OF AGT

One optimization of our novel ATG protocol is the possibility to efficiently compare one value with β others, formally $\{x > y_i\}_{i=1}^{\beta}$. Instead of using 1-bit messages in the 1-out-of-N OT construction, it is more efficient to "pool" and concatenate all message buffers for all batched comparisons and evaluate one, combined 1-out-of-N OT. For large values of β this improves the communication size by

$$\frac{(\gamma+1)(2\kappa+2^{\ell_s})+\lceil\epsilon/(\ell_s-1)\rceil(2\kappa+2^{\epsilon})}{(\gamma+1)2^{\ell_s}+\lceil\epsilon/(\ell_s-1)\rceil 2^{\epsilon}}.$$

The improvements in communication size for varying bit lengths are shown in Table C.1. At the cost of increasing the communication rounds to 3, and for ℓ=63 a factor of 3.1. At the cost of increasing $\ell-1$ the communication size can be reduced to $2\ell-2$ bits.

ℓ (bit)	7	15	31	63
Improvement	3.0×	3.9×	3.0×	3.1×

Table C.1: Communication improvement due to batching for AGT

Possible application of Batch-AGT includes the counting of arithmetic values greater than a certain threshold by invoking a Hamming weight calculation on the Batch-AGT results.

C.3 SECURITY OF THE NOVEL AGT

Informally, our AGT protocol's security is obvious, as it only a black-box composition of multiple $\binom{N}{1}$-OT operations producing uniformly distributed output values in each step.

More specifically, the first call to the $\binom{N}{1}$-OT functionality takes in the first ℓ_s bits of $\langle\delta\rangle^{\mathcal{A}}$ and results in a secret share $(c,r) \in \{0,1\}^2$, where

Protocol C.1: PEA's Private TuRF protocol

Function PTuRF(R, α)**:**

The dataset R is the concatenation of each data owners P_i raw dataset R_i.

The dataset consists of all records $R := (r^1, \ldots, r^k)$, where the record $r^j := ((g^{j,1}, \ldots, g^{j,m}), \alpha^j) : r^j \in R$ with each genotype $g^{j,\lambda} \in \{1, 2, 3\}$ of person j at locus λ and each group $\alpha \in \{+, -\}$, denoting the case and control group, respectively. The function returns the index positions of the most weighted genotypes. $\theta = n - \left\lfloor \frac{a\alpha}{n} \right\rfloor$ denotes the number of features to retain in each iteration, where $\lfloor \cdot \rfloor$ means rounding down to the nearest integer.

for $j = \sigma(1) \ldots \sigma(k)$ **do** // For all permuted records in the Dataset

// Initialize distance and difference matrices to the
// numerical maximum value and zero, respectively

$\langle m^{\text{hit}}_{\text{dist}} \rangle^{\mathcal{Y}} \leftarrow [\langle \text{MAX\_VALUE} \rangle^{\mathcal{Y}}, \ldots, \langle \text{MAX\_VALUE} \rangle^{\mathcal{Y}}]$

$\langle m^{\text{hit}}_{\text{ineq}} \rangle^{\mathcal{Y}} \leftarrow [[\langle 0 \rangle^{\mathcal{Y}}, \ldots, \langle 0 \rangle^{\mathcal{Y}}], \ldots, [\langle 0 \rangle^{\mathcal{Y}}, \ldots, \langle 0 \rangle^{\mathcal{Y}}]]$

$\langle m^{\text{miss}}_{\text{dist}} \rangle^{\mathcal{Y}} \leftarrow [\langle \text{MAX\_VALUE} \rangle^{\mathcal{Y}}, \ldots, \langle \text{MAX\_VALUE} \rangle^{\mathcal{Y}}]$

$\langle m^{\text{miss}}_{\text{ineq}} \rangle^{\mathcal{Y}} \leftarrow [[\langle 0 \rangle^{\mathcal{Y}}, \ldots, \langle 0 \rangle^{\mathcal{Y}}], \ldots, [\langle 0 \rangle^{\mathcal{Y}}, \ldots, \langle 0 \rangle^{\mathcal{Y}}]]$

for $\ell > j$ **do** // For all pairs of records

$\langle D_{j\ell} \rangle^{\mathcal{Y}} \leftarrow \emptyset$

for $\lambda = 1 \ldots m$ **do** // For all genotypes

$\langle D_{j\ell} \rangle^{\mathcal{Y}}$.**append** $\left(\Delta\left(\langle g^{j,\lambda} \rangle^{\mathcal{Y}}, \langle g^{\ell,\lambda} \rangle^{\mathcal{Y}}\right)\right)$

for $\forall \ell \neq j$ **do** // For all (unordered) pairs

if $j < \ell$ **then**

$\langle d \rangle^{\mathcal{Y}} \leftarrow \text{Hw}(\langle D_{j\ell} \rangle^{\mathcal{Y}})$

else

$\langle d \rangle^{\mathcal{Y}} \leftarrow \text{Hw}(\langle D_{\ell j} \rangle^{\mathcal{Y}})$

if $\langle \alpha^j \rangle^{\mathcal{Y}} == \langle \alpha^\ell \rangle^{\mathcal{Y}}$ **then** // If records have same label

$\langle m^{\text{hit}}_{\text{dist}} \rangle^{\mathcal{Y}}, \langle m^{\text{hit}}_{\text{ineq}} \rangle^{\mathcal{Y}} \leftarrow \texttt{kNN}(\langle m^{\text{hit}}_{\text{dist}} \rangle^{\mathcal{Y}}, \langle m^{\text{hit}}_{\text{ineq}} \rangle^{\mathcal{Y}}, \langle d \rangle^{\mathcal{Y}}, k)$

else

$\langle m^{\text{miss}}_{\text{dist}} \rangle^{\mathcal{Y}}, \langle m^{\text{miss}}_{\text{ineq}} \rangle^{\mathcal{Y}} \leftarrow \texttt{kNN}(\langle m^{\text{miss}}_{\text{dist}} \rangle^{\mathcal{Y}}, \langle m^{\text{miss}}_{\text{ineq}} \rangle^{\mathcal{Y}}, \langle d \rangle^{\mathcal{Y}}, k)$

$\langle W \rangle^{\mathcal{Y}} \leftarrow \langle W \rangle^{\mathcal{Y}} + \langle m^{\text{miss}}_{\text{ineq}} \rangle^{\mathcal{Y}} - \langle m^{\text{hit}}_{\text{ineq}} \rangle^{\mathcal{Y}}$

for $\forall \ell$ **do**

// The features are sorted by weight and only the first
// (best) $\varphi \cdot a$ are retained

$\langle g'^{\ell} \rangle^{\mathcal{Y}} \leftarrow \texttt{kNN}(\langle g^{\ell} \rangle^{\mathcal{Y}}, \langle W \rangle^{\mathcal{Y}}, \varphi \cdot a)$

$\langle g^{\ell} \rangle^{\mathcal{Y}} \leftarrow \langle g'^{\ell} \rangle^{\mathcal{Y}}[1 : \varphi \cdot a]$

return $\langle R \rangle^{\mathcal{Y}}$

$c := (\langle\delta\rangle_0^{\mathcal{A}}[1:\ell_s] + \langle\delta\rangle_1^{\mathcal{A}}[1:\ell_s] \geq 2^{\ell_s}) \oplus r$. The uniformly random bit r is only known to the OT sender and "blinds" c, hence, forming a uniformly distributed secret share (c, r).

The following $\binom{N}{1}$-OT calls process the remaining substrings of $\langle\delta\rangle^{\mathcal{A}}$: Parties P_0 and P_1 call $\binom{N}{1}$-OT, resulting in a new (c, r) share pair, where $c := (\langle\delta\rangle_0^{\mathcal{A}}[\ell_{\text{prev}} : \ell_{\text{prev}} + \ell'_s - 1] + \langle\delta\rangle_1^{\mathcal{A}}[\ell_{\text{prev}} : \ell_{\text{prev}} + \ell'_s - 1] + (c_{\text{prev}} \oplus r_{\text{prev}}) \geq 2^{\ell'_s}) \oplus r$. The results of the previous OT calls are written as $(c_{\text{prev}}, r_{\text{prev}})$ and r again is a random bit generated and known only by the OT sender.

The resulting final secret share is constructed locally using the intermediate shares described above.

A formal proof can be derived trivially from the security proof of Algorithm 2 by Rathee et al.[2].

[2] Rathee et al. (2020)

C.4 Batch Operation of ASWAP

ASWAP is easily generalizable to a batch construction taking one (fixed) $\langle b\rangle^{\mathcal{B}}$ and for batches of size β a vector $\{\langle x_{i,0}\rangle^{\mathcal{A}}, \langle x_{i,1}\rangle^{\mathcal{A}}\}_{i=1}^{\beta}$ as inputs. The replacement of the multiplication in the computation of $\langle\delta\rangle^{\mathcal{A}}$ by a batched multiplication inspired by Schneider and Tkachenko [3] reduces the communication size from $2\beta(\kappa + \ell)$ to $2(\kappa + \beta\ell)$. This results in an improvement factor of $\beta(\kappa + \ell)/(\kappa + \beta\ell)$, approximately $\sim 1 + \kappa/\ell$ for large values of β. Note, that for PEA the security factor is set to κ=128.

[3] Schneider and Tkachenko (2019)

Batch-ASWAP is useful in combination with AGT to achieve some kind of k-anonymity[4] by suppressing arithmetic values smaller than some thresholds. If the number of communication rounds is not critical, e.g., for very large numbers of input values or negligible network latency the parallelization of the computation can result in amortized communication sizes of $\sim 4\ell$ bit. The blinding of one million 32-bit values against a common threshold would result in only 16 MB of communication.

[4] See Section 2.2

C.5 Security

The security of both ASWAP and batch-ASWAP follow from the security of the correlated OT construction[5], as they only compose C-OT with unmodified secret shares in a black-box fashion.

[5] Asharov, Lindell, et al. (2017)

C.6 Correctness of ASWAP

Both Arithmetic Swap (ASWAP) and batch-ASWAP, compute the same expression using different primitives for oblivious computation. The prove of correctness, however, is equivalent for both protocol variants:

$$\begin{aligned}
\langle x_0' \rangle^{\mathcal{A}} &= \langle x_0 \rangle^{\mathcal{A}} + \langle \delta \rangle^{\mathcal{A}} \\
&= (\langle b \rangle^{\mathcal{B}} + \neg \langle b \rangle^{\mathcal{B}}) \cdot \langle x_0 \rangle^{\mathcal{A}} \quad + \langle b \rangle^{\mathcal{B}} \cdot (\langle x_1 \rangle^{\mathcal{A}} - \langle x_0 \rangle^{\mathcal{A}}) \\
&= \langle b \rangle^{\mathcal{B}} \cdot \langle x_0 \rangle^{\mathcal{A}} + \neg \langle b \rangle^{\mathcal{B}} \cdot \langle x_0 \rangle^{\mathcal{A}} \quad + \langle b \rangle^{\mathcal{B}} \cdot \langle x_1 \rangle^{\mathcal{A}} - \langle b \rangle^{\mathcal{B}} \cdot \langle x_0 \rangle^{\mathcal{A}}) \\
&= \langle b \rangle^{\mathcal{B}} \cdot \langle x_1 \rangle^{\mathcal{A}} + \neg \langle b \rangle^{\mathcal{B}} \cdot \langle x_0 \rangle^{\mathcal{A}}
\end{aligned}$$

$$\begin{aligned}
\langle x_1' \rangle^{\mathcal{A}} &= \langle x_1 \rangle^{\mathcal{A}} - \langle \delta' \rangle^{\mathcal{A}} \\
&= (\langle b \rangle^{\mathcal{B}} + \neg \langle b \rangle^{\mathcal{B}}) \cdot \langle x_1 \rangle^{\mathcal{A}} \quad - \langle b \rangle^{\mathcal{B}} \cdot (\langle x_1 \rangle^{\mathcal{A}} - \langle x_0 \rangle^{\mathcal{A}}) \\
&= \langle b \rangle^{\mathcal{B}} \cdot \langle x_1 \rangle^{\mathcal{A}} + \neg \langle b \rangle^{\mathcal{B}} \cdot \langle x_1 \rangle^{\mathcal{A}} \quad - \langle b \rangle^{\mathcal{B}} \cdot \langle x_1 \rangle^{\mathcal{A}} + \langle b \rangle^{\mathcal{B}} \cdot \langle x_0 \rangle^{\mathcal{A}}) \\
&= \langle b \rangle^{\mathcal{B}} \cdot \langle x_0 \rangle^{\mathcal{A}} + \neg \langle b \rangle^{\mathcal{B}} \cdot \langle x_1 \rangle^{\mathcal{A}}
\end{aligned}$$

C.7 Fit Parameter

For all extrapolations of PEA's runtimes and all network settings the following power-function model was used:

$$f(x) = a \cdot x^b + c$$

The parameters shown in Tables C.2 and C.3 were computed in Matlab 2021a (9.10.0.1602286) with the Trust-Region algorithm with a maximum of 400 iterations. All coefficients are given with 95 % confidence bounds. The fitting was perfomed based on timings measured in milliseconds.

Parameter	Setup Phase	Confidence Bounds	Online Phase	Confidence Bounds
		LAN setting		
a	-304.1	$(-3.4 \times 10^4, 3.409 \times 10^4)$	3.842	$(1.261, 6.424)$
b	-0.051	$(-6.641, 6.539)$	2.703	$(2.523, 2.884)$
c	496.5	$(-3.417 \times 10^4, 3.516 \times 10^4)$	775.2	$(-201.3, 1{,}752)$
r^2	0.9735		0.9999	
RMSE	3.93		96.02	
		WAN setting		
a	225.1	$(-6{,}366, 6{,}816)$	2.787	$(0.3154, 5.258)$
b	0.6419	$(-6.003, 7.287)$	2.791	$(2.552, 3.029)$
c	-74.56	$(-1.286 \times 10^4, 1.271 \times 10^4)$	877.6	$(-329.6, 2{,}085)$
r^2	0.9773		0.9999	
RMSE	216.89		120.08	

Table C.2: Fit parameters for varying numbers of records using PTuRF

Parameter	Setup Phase	Confidence Bounds	Online Phase	Confidence Bounds
		LAN setting		
a	0.2905	(−1.82,2.401)	28.88	(19.47, 38.28)
b	0.8467	(−0.6702,2.364)	1.6	(1.53, 1.671)
c	221	(212,230)	579.8	(40.56, 1,119)
r^2	0.8289		0.9998	
RMSE	2.90		266.16	
		WAN setting		
a	1,208	(−4,008,6,424)	25.92	(6.041, 45.8)
b	0.2527	(−0.3819,0.8873)	1.618	(1.452, 1.784)
c	−966.4	(−6,833,4,900)	538.6	(−679.2, 1,756)
r^2	0.9166		0.9991	
RMSE	346.69		604.59	

Table C.3: Fit parameters for varying numbers of features using PTuRF

CHAPTER D
Kidney Exchange Supplementary Material

D.1 SPIKE SUBPROTOCOLS

As many subprotocols of *Secure and Private Investigation of the Kidney Exchange problem* (SPIKE) are either similar to the ones shown in the main text or structurally repetitive, they are presented in the following sections.

D.1.1 Subprotocols for Compatibility Matching

The subprotocols in the compatibility matching phase of SPIKE calculate the individual match quality factors' contributions to the edge weight. A recurring theme is the categorization according to the medical factor in question and a category associated return value.

```
1 Function evaluateHLA(⟨hla_d⟩^B, ⟨hla_r⟩^B):
2     ⟨mm⟩^B ← {⟨0⟩^B}^|HLA|
3     for i = 1 ... |HLA| do
4         ⟨mm⟩^B ← ⟨hla_d⟩^B[i] ⊕ ⟨hla_r⟩^B ; // SIMD
5     ⟨sum⟩^B ← HammingW({⟨0⟩^B}^|HLA|, ⟨mm⟩^B)
6     ⟨c⟩^B ← ⟨sum⟩^B < ⟨5⟩^B
7     ⟨b⟩^B ← ⟨sum⟩^B < ⟨3⟩^B
8     ⟨a⟩^B ← ⟨sum⟩^B == ⟨0⟩^B
9     return ⟨a⟩^B ? ⟨A⟩^B (⟨b⟩^B ? ⟨B⟩^B : (⟨c⟩^B ? ⟨C⟩^B : ⟨0⟩^B))
```

Subprotocol D.1: Subprotocol `evalHLA` compares the number of *Human Leucocyte Antigens* (HLA) mismatches between donor and recipient.

HLA Antigen Comparison. Subprotocol D.1 calculates the number of HLA mismatches between the donor and recipient. Depending on the number of mismatches, the weight associated with the respective classes is returned. The number of comparisons and `MUX` gates suggest the usage of a Boolean circuit-based protocol. To avoid conversions, the circuit is evaluated in Boolean GMW ($\mathcal{B}$).

```
1 Function evaluateABO(⟨bg_d⟩^B, ⟨bg_r⟩^B):
2     ⟨a⟩^B ← ¬((⟨bg_r⟩^B[1] ⊕ ⟨bg_d⟩^B[1]) ∨ (⟨bg_r⟩^B[2] ⊕ ⟨bg_d⟩^B[2]))
3     ⟨b⟩^B ← (⟨bg_r⟩^B[2] ∧ ¬⟨bg_d⟩^B[1]) ∨ (⟨bg_r⟩^B[1] ∧ ¬⟨bg_d⟩^B[2])
4     ⟨v⟩^B ← ⟨a⟩^B ∨ ⟨b⟩^B
5     return ⟨v⟩^B ? ⟨best_age⟩^B : ⟨0⟩^B
```

Subprotocol D.2: Subprotocol `evalABO` checks the donor's and the recipient's blood group on compatibility.

ABO blood group comparison. In Subprotocol D.2, the blood group of the donor bg_d and the blood group of the recipient bg_r—both encoded according to Table D.1—are checked on compatibility[1]. Using said encoding, compatibility manifests in the truthful evaluation of at least one of the conditions: (1) $bg_d = bg_r$, (2) $bg_r[1] > bg_d[0]$, or (3) $bg_r[0] > bg_d[1]$. This subprotocol is evaluated in $\mathcal{B}$.

[1] See Table 4.2

Table D.1: Encoding of the different blood groups

Encoding	Blood Group
00	O
01	A
10	B
11	AB

Age Comparison. Following the age groups of WAISER ET AL.[2], Subprotocol D.3

[2] Waiser et al. (2000)

Subprotocol D.3: Subprotocol `evalAge` categorizes the match quality according to the three categories by Waiser et al. (2000).

```
1 Function evaluateAge(⟨a_d⟩^B, ⟨a_r⟩^B):
2     ⟨eq⟩^B ← ⟨a_d⟩^B == ⟨a_r⟩^B
3     ⟨yg⟩^B ← ¬⟨a_d⟩^B ∧ ⟨a_r⟩^B
4     return ⟨yg⟩^B ? (⟨eq⟩^B ? ⟨A⟩^B : ⟨B⟩^B) : (⟨eq⟩^B ? ⟨A⟩^B : ⟨0⟩^B)
```

evaluates the age-group memberships of the donors and recipients to return compatibility scores according to the combination optimality. This subprotocol uses $\mathcal{B}$, due to its low multiplicative depth.

Subprotocol D.4: Subprotocol `evalSex` categorizes the match quality according to the sex disparities following Zhoua et al. (2013).

```
1 Function evaluateSex(⟨s_d⟩^B, ⟨s_r⟩^B):
2     ⟨eq⟩^B ← ⟨s_d⟩^B == ⟨s_r⟩^B
3     ⟨fdmr⟩^B ← ⟨s_d⟩^B ∧ ¬⟨s_r⟩^B
4     return ⟨fdmr⟩^B ? (⟨eq⟩^B ? ⟨A⟩^B : ⟨0⟩^B) : (⟨eq⟩^B ? ⟨A⟩^B : ⟨B⟩^B)
```

Sex Comparison. Subprotocol D.4 returns a match quality score based on the combination of biological sexes of the donors and recipients following the recommendations of Zhoua et al.[3]. This subprotocol is logically identical to Subprotocol D.3, hence, it is also evaluated in $\mathcal{B}$.

[3] Zhoua et al. (2013)

Subprotocol D.5: Subprotocol `evalWeight` compares the weight of the donors and recipients to determine a match quality following Miller et al. (2017).

```
1 Function evaluateWeight(⟨w_d⟩^B, ⟨w_r⟩^B):
2     return ⟨w_d⟩^B < ⟨w_r⟩^B ? ⟨0⟩^B : ⟨A⟩^B
```

Weight Comparison. Miller et al.[4] describe the transplantation success probabilities with regard to the weight differences of donors and recipients, hence, Subprotocol D.5 evaluates the compatibility of a donor and recipient based on their weight. While $\mathcal{Y}$ allows for a more efficient comparison, the cost of converting the result to $\mathcal{B}$ for processing in the calling procedure makes the complete evaluation in $\mathcal{B}$ more efficient.

[4] Miller et al. (2017)

D.1.2 Subprotocols for Cycle Computation

(Edge) Weight Removal. For the calculation of the numbers of cycles in the graph a unweighted adjacency matrix is required. Subprotocol D.6 creates an (arithmetically shared) unweighted adjacency matrix by setting all matrix elements with a non-zero weight to one. While this operation could be performed in constant time—all matrix elements are independent—the runtime of this subprotocol is nearly negligible for realistic matrix sizes, such that no premature *Single Instruction Multiple Data* (SIMD) optimization was attempted.

kNN Sort Protocol. For partially sorting the list of cycles based on the cycle weight, Subprotocol D.7 performs a kNN-Sort, based on the kNN protocol of Järvinen et al.[5]. The k cycler with the most weight are sorted to the first array positions. Note, that the length of cycles `cLen` is a public parameter. Because the multiplicative operations are mostly dependent on each other, `kNNSort` results in a rather deep circuit. This leads to a most efficient evaluation in $\mathcal{Y}$, even if conversion costs are included.

[5] Järvinen et al. (2019)

```
1  Function removeWeights(⟨compG⟩^B):
2      ⟨uG⟩^A ← matrix ∈ ⟨0⟩^{A^|pairs|}
3      for i = 1 ... |pairs| do
4          for j = 1 ... |pairs| do
5              ⟨uG⟩^A[i][j] ← b2a(⟨compG⟩^B[i][j] > ⟨0⟩^B ? ⟨1⟩^B : ⟨0⟩^B)
6      return ⟨uG⟩^A
```

Subprotocol D.6: Subprotocol `removeWeights` converts a weighted adjacency matrix to a unweighted one

```
1  Function kNNSort(⟨cyclesSet⟩^Y, k):
2      ⟨sortedW⟩^Y ← ∅
3      ⟨sortedC⟩^Y ← ∅
4      for i = 1 ... k + 1 do
5          ⟨sortedW⟩^Y.append(⟨0⟩^Y)
6          ⟨vertices⟩^Y ← ∅
7          for j = 1 ... cLen do
8              ⟨vertices⟩^Y.append(|⟨pairs⟩^Y|)
9          ⟨sortedC⟩^Y.append(⟨vertices⟩^Y)
10     for i = 1 ... |cyclesSet| do
11         ⟨sortedW⟩^Y[k] ← ⟨cyclesSet⟩^Y[i][1]
12         ⟨sortedC⟩^Y[k] ← ⟨cyclesSet⟩^Y[i][2]
13         for j = 1 ... k do
14             ⟨sel⟩^Y ← ⟨sortedW⟩^Y[j] > ⟨sortedW⟩^Y[j − 1]
15             ⟨tmp1⟩^Y ← ⟨sortedW⟩^Y[j]
16             ⟨tmp2⟩^Y ← ⟨sortedW⟩^Y[j − 1]
17             ⟨sortedW⟩^Y[j] ← ⟨sel⟩^Y ? ⟨tmp2⟩^Y : ⟨tmp1⟩^Y
18             ⟨sortedW⟩^Y[j − 1] ← ⟨sel⟩^Y ? ⟨tmp1⟩^Y : ⟨tmp2⟩^Y
19             for l = 1 ... cLen do
20                 ⟨tmp1⟩^Y ← ⟨sortedC⟩^Y[j][l]
21                 ⟨tmp2⟩^Y ← ⟨sortedC⟩^Y[j − 1][l]
22                 ⟨sortedC⟩^Y[j][l] ← ⟨sel⟩^Y ? ⟨tmp2⟩^Y : ⟨tmp1⟩^Y
23                 ⟨sortedC⟩^Y[j − 1][l] ← ⟨sel⟩^Y ? ⟨tmp1⟩^Y : ⟨tmp2⟩^Y
24     ⟨result⟩^Y ← ∅
25     for i = 1 ... |cycles| do
26         ⟨result⟩^Y.append(tuple(⟨sortedW⟩^Y[i], ⟨sortedC⟩^Y[i]))
27     return ⟨result⟩^Y
```

Subprotocol D.7: Subprotocol `kNNSort` partially sorts the array of cycles based on the cycle weights. It is based on Järvinen et al. (2019).

```
1  Function removeDuplicates(⟨sortedCycles⟩^Y):
2      for i = 1 ... |cycles| do
3          ⟨c1⟩^Y ← ⟨sortedCycles⟩^Y[i][2]
4          ⟨combDup⟩^Y ← ⟨0⟩^Y
5          for j = 1 ... i do
6              ⟨c2⟩^Y ← ⟨sortedCycles⟩^Y[j][2]
7              for k = 2 ... cLen do
8                  ⟨duplicate⟩^Y ← ⟨1⟩^Y
9                  for l = 1 ... cLen do
10                     ⟨same⟩^Y ← ⟨c1⟩^Y[l] == ⟨c2⟩^Y[(l + k) mod cLen]
11                     ⟨duplicate⟩^Y ← ⟨duplicate⟩^Y ∧ ⟨same⟩^Y
12                 ⟨combDup⟩^Y ← ⟨combDup⟩^Y ∨ ⟨duplicate⟩^Y
13         ⟨sortedCycles⟩^Y[i][1] ← ⟨isDuplicate⟩^Y ? ⟨0⟩^Y : ⟨sortedCycles⟩^Y[i][1]
14     return kNNSort(⟨sortedCycles⟩^Y, |unique|)
```

Subprotocol D.8: Subprotocol `removeDuplicates` removes all duplicate cycles.

Duplicate Removal. Subprotocol D.8 iterates over all (sorted) cycles and removes all duplicates—that is cycles "seen" before. This results in $|\text{unique}| = \lfloor \frac{|\text{cycles}|}{\text{cLen}} \rfloor$ cycles remaining. The removal works by setting the cycle weight to 0, hence, assuring—after another kNN-sort that these "zeroed" cycles are never included in a solution. The serial structure and the number of comparisons and MUX gates necessary form a circuit with high multiplicative depth. Because of that, it is evaluated in $\mathcal{Y}$.

Subprotocol D.9: Subprotocol `#TotalCycles` calculates the total number of cycles. All inputs are publicly known, hence, this protocol is performed locally in clear text.

```
1 Function #TotalCycles():
2     |allCycles| ← |pairs|
3     for i = 1, ..., cLen − 1 do
4         |allCycles| ← |allCycles| · (|pairs| − i)
5     return |allCycles|
```

Total Number of Cycles. Subprotocol D.9 computes the maximum number of cycles in the compatibility graph, based on the number of vertices. As only vertex disjointed cycles are allowed, each vertex can be member of only one cycle set. The parameters |pairs| and cLen are publicly known, hence, this calculation can be performed locally and in clear text.

D.1.3 Subprotocols for Solution Evaluation

Subprotocol D.10: Subprotocol `disjointSet` checks, whether a given set of cycles is vertex disjointed.

```
1  Function disjointSet(⟨cycles⟩^B, ⟨cCycle⟩^B, count):
2      ⟨disJ⟩^B ← ∅
3      for i = 1 ... count do
4          ⟨c⟩^B ← ⟨cycles⟩^B[i][2]
5          for j = 1 ... cLen do
6              for k = 1 ... cLen do
7                  ⟨tmp⟩^B ← ⟨c⟩^B[j] == ⟨cCycle⟩^B[k]
8                  ⟨disJ⟩^B.append(⟨tmp⟩^B)
9          ⟨disJ⟩^B ← ORTree(⟨disJ⟩^B)
10     return ¬⟨disJ⟩^B[1]
```

Disjoint Cycles. Subprotocol D.10 checks a given set of cycles for disjointness. To reduce the circuit depth, the final OR-fold of bits is performed in a tree structure. Because of this optimization, the circuit is most efficiently evaluated in $\mathcal{B}$.

Maximum Set. Finally, Subprotocol D.11 aggregates the cycle weighs in a possible solution set. This aggregated weight is the optimization target, as it represents an solution set-wide transplantation success probability. Another invocation of a kNN-sort returns the (locally) optimal solution set—that is the set with the highest aggregated weight. As in the circuits using kNN-sorting before, the high circuit depth leads to a most efficient evaluation in $\mathcal{Y}$.

```
1  Function findMaximumSet(⟨cyclesSets⟩^𝒱, ⟨cycleW⟩^𝒱):
2      ⟨weights⟩^𝒱 ← ∅
3      ⟨tmp⟩^𝒱 ← ∅
4      for i = 1, 2 do
5          ⟨weights⟩^𝒱.append(⟨0⟩^𝒱)
6          ⟨sets⟩^𝒱 ← ∅
7          for j = 1 ... |unique| do
8              ⟨vertices⟩^𝒱 ← ∅
9              for l = 1 ... cLen do
10                 ⟨vertices⟩^𝒱.append(⟨|pairs|⟩^𝒱)
11             ⟨tmp⟩^𝒱.append(⟨vertices⟩^𝒱)
12         ⟨sets⟩^𝒱.append(⟨tmp⟩^𝒱)
13     for i = 1 ... |unique| do
14         ⟨weights⟩^𝒱[2] ← ⟨cycleW⟩^𝒱[i]
15         ⟨sets⟩^𝒱[2] ← ⟨cycleSets⟩^𝒱[i]
16         ⟨sel⟩^𝒱 ← ⟨weights⟩^𝒱[2] > ⟨weights⟩^𝒱[1]
17         ⟨tmp1⟩^𝒱 ← ⟨weights⟩^𝒱[2]
18         ⟨tmp2⟩^𝒱 ← ⟨weights⟩^𝒱[1]
19         ⟨weights⟩^𝒱[2] ← ⟨sel⟩^𝒱 ? ⟨tmp2⟩^𝒱 : ⟨tmp1⟩^𝒱
20         ⟨weights⟩^𝒱[1] ← ⟨sel⟩^𝒱 ? ⟨tmp1⟩^𝒱 : ⟨tmp2⟩^𝒱
21         for j = 1 ... |unique| do
22             ⟨tmp1⟩^𝒱 ← ⟨sets⟩^𝒱[2][j]
23             ⟨tmp2⟩^𝒱 ← ⟨sets⟩^𝒱[1][j]
24             ⟨sets⟩^𝒱[2][j] ← ⟨sel⟩^𝒱 ? ⟨tmp2⟩^𝒱 : ⟨tmp1⟩^𝒱
25             ⟨sets⟩^𝒱[1][j] ← ⟨sel⟩^𝒱 ? ⟨tmp1⟩^𝒱 : ⟨tmp2⟩^𝒱
26     return (⟨weights⟩^𝒱[1], ⟨sets⟩^𝒱[1])
```

Subprotocol D.11: Subprotocol `findMaximumSet` calculates the aggregate weight of the solution sets and returns the maximum set.

D.2 Detailed Benchmark Results

Table D.2 to D.4 show the complete benchmark results for all three network settings (A: LAN with 10 Gbit/s, B: LAN with 1 Gbit/s, C: WAN) and a cycle length of $L = 2$. Table D.5 and D.6 show the results for a cycle length of $L = 3$.

Table D.7, presents the benchmark results of both reduced medical compatibility factor set and the full set. This benchmark was performed in LAN and WAN network settings.

Table D.2: Comparison of the communication costs and setup and online runtimes of SPIKE for the three networking settings A: LAN with 10 Gbit/s, B: LAN with 1 Gbit/s, C: WAN, for cycle length $L = 2$. This table contains both the aggregated total results and the results of Phase 1 (Compatibility Matching).

Pairs	Comm. [MiB]		Setup Phase [s]			Online Phase [s]		
Pairs	Setup	Online	A	B	C	A	B	C
Total								
2	0.1	0	0.021	0.021	0.78	0.04	0.039	2.1
4	1.1	0.1	0.052	0.051	1.7	0.075	0.08	3.1
6	3.4	0.3	0.1	0.11	2.5	0.15	0.15	4.3
8	5.6	0.4	0.13	0.17	3	0.17	0.18	4.4
10	12.7	0.8	0.22	0.24	4	0.28	0.29	5.8
12	19.5	1	0.37	0.34	4.4	0.46	0.37	6.6
14	55.8	2.3	0.61	0.68	7.4	0.8	0.88	12
16	95.4	3.4	0.94	1.1	11	1.2	1.3	15
18	159	5.1	1.4	1.6	15	1.8	1.9	18
20	412.1	11.8	2.9	4.9	34	4.2	7.4	30
22	617.8	16.6	4.2	5.2	47	6.3	6.4	36
24	823.3	21.1	5.5	6.7	64	8.4	8.5	42
26	1,104.8	27	7.2	8.7	81	11	11	49
28	1,281.6	30.2	8.3	10	93	13	13	53
30	1,608.3	36.5	10	13	120	17	17	59
32	2,202.9	48.3	14	19	150	24	24	71
34	2,999.7	63.8	18	22	200	33	33	85
36	3,971.7	82.2	24	26	260	44	43	100
38	5,036.2	101.8	29	35	320	57	57	120
40	6,394	126.6	37	45	400	75	75	140
Phase 1: Compatibility Matching								
2	0	0	0.0071	0.0065	0.31	0.015	0.015	0.85
4	0.1	0	0.0093	0.0087	0.42	0.016	0.015	0.85
6	0.2	0	0.012	0.013	0.52	0.017	0.017	0.85
8	0.4	0	0.016	0.016	0.62	0.019	0.018	0.84
10	0.6	0	0.02	0.021	0.62	0.021	0.021	0.85
12	0.8	0	0.026	0.025	0.65	0.024	0.024	0.85
14	1.2	0	0.031	0.032	0.72	0.028	0.028	0.86
16	1.5	0	0.036	0.038	0.75	0.034	0.031	0.86
18	1.9	0	0.047	0.045	0.82	0.033	0.033	0.86
20	2.4	0	0.053	0.054	0.85	0.039	0.039	0.88
22	2.9	0.1	0.055	0.065	0.87	0.045	0.046	0.88
24	3.4	0.1	0.071	0.073	0.9	0.05	0.049	0.89
26	4	0.1	0.075	0.083	1	0.051	0.056	0.9
28	4.6	0.1	0.077	0.085	1	0.059	0.06	0.91
30	5.3	0.1	0.081	0.088	1.1	0.068	0.067	0.97
32	6.1	0.1	0.084	0.09	1.1	0.071	0.069	0.98
34	6.8	0.1	0.087	0.092	1.1	0.079	0.083	0.97
36	7.7	0.1	0.093	0.099	1.2	0.085	0.089	0.97
38	8.6	0.2	0.093	0.11	1.2	0.091	0.098	0.99
40	9.5	0.2	0.094	0.11	1.2	0.1	0.1	1

Table D.3: Comparison of the communication costs and setup and online runtimes of SPIKE for the three networking settings A: LAN with 10 Gbit/s, B: LAN with 1 Gbit/s, C: WAN, for cycle length $L = 2$. This table contains the results of Phases 2 and 3 (Cycle Computation and Evaluation).

Pairs	Comm. [MiB]		Setup Phase [s]			Online Phase [s]		
Pairs	Setup	Online	A	B	C	A	B	C
Phase 2: Cycle Computation								
2	0	0	0.0099	0.0099	0.43	0.013	0.012	0.75
4	0.2	0	0.013	0.013	0.54	0.014	0.014	0.76
6	0.4	0.1	0.02	0.02	0.83	0.017	0.018	0.76
8	0.9	0.1	0.028	0.031	1	0.021	0.021	0.85
10	1.7	0.2	0.043	0.047	1.2	0.024	0.027	0.77
12	2.8	0.3	0.06	0.059	1.3	0.033	0.031	0.79
14	4.3	0.4	0.082	0.087	1.6	0.034	0.04	0.8
16	6.2	0.5	0.1	0.12	1.8	0.047	0.048	0.82
18	8.6	0.7	0.12	0.12	1.8	0.048	0.054	0.84
20	11.6	0.8	0.13	0.14	2	0.063	0.061	0.87
22	15.3	1	0.13	0.17	2	0.075	0.072	0.9
24	19.6	1.2	0.15	0.19	2.9	0.078	0.083	1.1
26	24.6	1.5	0.17	0.23	3.2	0.088	0.1	1.3
28	30.5	1.7	0.19	0.27	5	0.1	0.11	2.4
30	37.2	2	0.22	0.3	4.8	0.11	0.12	2
32	44.8	2.3	0.24	0.35	5.7	0.12	0.14	2.2
34	53.4	2.6	0.27	0.42	6.5	0.13	0.15	2.3
36	63	3	0.3	0.48	7.1	0.13	0.16	2.3
38	73.7	3.4	0.34	0.55	7.9	0.14	0.17	2.3
40	85.6	3.8	0.38	0.63	8.8	0.16	0.18	2.4
Phase 3: Cycle Evaluation								
2	0.1	0	0.0023	0.0027	0.022	0.0086	0.0082	0.3
4	0.7	0.1	0.019	0.02	0.29	0.026	0.03	0.35
6	2.2	0.1	0.054	0.061	0.56	0.068	0.07	0.47
8	3.8	0.2	0.066	0.1	0.74	0.089	0.096	0.48
10	8.6	0.4	0.12	0.13	1.2	0.14	0.16	0.56
12	13.4	0.5	0.21	0.2	1.6	0.22	0.2	0.66
14	35	0.9	0.38	0.41	3.6	0.37	0.43	0.94
16	57.3	1.3	0.62	0.66	5.6	0.6	0.69	1.2
18	90.2	1.8	0.98	1	8.5	0.87	0.95	1.4
20	181.2	3.3	1.9	2.2	17	1.7	1.9	2.3
22	255.2	4.4	2.7	2.9	23	2.4	2.5	3
24	332.8	5.3	3.6	3.8	30	3.1	3.1	3.7
26	431.8	6.5	4.7	4.9	39	4	4	4.5
28	514.4	7.2	5.5	5.7	46	4.7	4.7	5.3
30	635.1	8.4	6.9	7.3	57	6	5.9	6.4
32	815.8	10.4	8.9	12	73	7.5	9.7	8
34	1,037.4	12.8	11	12	92	9.4	9.3	10
36	1,292.4	15.4	15	14	110	12	10	12
38	1,567.8	18	18	19	140	14	14	15
40	1,894.4	21.2	22	23	170	18	18	18

Pairs	Comm. [MiB]		Setup Phase [s]			Online Phase [s]		
Pairs	Setup	Online	A	B	C	A	B	C
Part 4: Solution Evaluation								
2	0	0	0.002	0.0016	0.0071	0.0038	0.0037	0.22
4	0.2	0	0.01	0.01	0.42	0.02	0.021	1.2
6	0.5	0.1	0.019	0.019	0.63	0.044	0.045	2.2
8	0.5	0.1	0.018	0.019	0.62	0.045	0.045	2.2
10	1.8	0.2	0.043	0.041	0.96	0.088	0.088	3.6
12	2.4	0.2	0.078	0.055	0.84	0.18	0.11	4.3
14	15.4	0.9	0.12	0.15	1.5	0.37	0.39	9.4
16	30.4	1.6	0.18	0.26	2.5	0.55	0.55	12
18	58.2	2.6	0.27	0.44	4	0.8	0.84	15
20	216.9	7.6	0.84	2.5	14	2.4	5.4	26
22	344.5	11.2	1.3	2	21	3.8	3.9	31
24	467.5	14.5	1.7	2.7	30	5.2	5.3	36
26	644.4	19	2.3	3.5	38	7.2	7.4	42
28	732.1	21.1	2.5	3.9	41	8.3	8.4	44
30	930.7	26	3.2	4.8	53	11	11	50
32	1,336.2	35.5	4.5	5.7	73	16	14	60
34	1,902.1	48.2	6.4	9.2	100	23	23	72
36	2,608.7	63.7	8.7	12	140	32	32	86
38	3,386.1	80.2	11	16	180	43	43	100
40	4,404.4	101.4	15	21	230	57	57	120

Table D.4: Comparison of the communication costs and setup and online runtimes of SPIKE for the three networking settings A: LAN with 10 Gbit/s, B: LAN with 1 Gbit/s, C: WAN, for cycle length $L = 2$. This table contains the result of Phase 4 (Solution Evaluation).

Table D.5: Comparison of the communication costs and setup and online runtimes of SPIKE for the three networking settings A: LAN with 10 Gbit/s, B: LAN with 1 Gbit/s, C: WAN, for cycle length $L = 3$. This table contains the aggregated total result and the results of Phases 1 and 2 (Compatibility Matching and Cycle Computation).

Pairs	Comm. [MiB]		Setup Phase [s]			Online Phase [s]		
Pairs	Setup	Online	A	B	C	A	B	C
Total								
3	0.5	0.1	0.029	0.028	0.97	0.054	0.056	2.2
5	4	0.3	0.096	0.11	2.2	0.13	0.15	2.9
7	19.7	0.7	0.26	0.27	4.1	0.3	0.34	4.3
9	52.6	1.5	0.63	0.66	7.3	0.63	0.73	5.3
11	182.5	3.3	2	2.1	19	1.9	2	9.4
13	1,215.8	16.3	12	13	110	12	12	34
15	2,084.4	22.8	21	23	180	19	20	45
17	5,428.5	58.1	52	56	440	54	54	93
18	9,537.2	107.6	88	95	740	100	100	150
Phase 1: Compatibility Matching								
3	0.1	0	0.0079	0.0075	0.32	0.015	0.015	0.85
5	0.1	0	0.011	0.01	0.42	0.016	0.016	0.85
7	0.3	0	0.014	0.014	0.52	0.018	0.018	0.85
9	0.5	0	0.019	0.018	0.61	0.019	0.02	0.85
11	0.7	0	0.023	0.024	0.64	0.023	0.023	0.86
13	1	0	0.029	0.029	0.72	0.025	0.024	0.86
15	1.3	0	0.034	0.036	0.74	0.029	0.029	0.86
17	1.7	0	0.041	0.043	0.77	0.035	0.034	0.87
18	1.9	0	0.042	0.049	0.82	0.035	0.034	0.87
Phase 2: Cycle Computation								
3	0.1	0	0.013	0.012	0.54	0.014	0.013	0.76
5	0.4	0	0.018	0.019	0.83	0.016	0.016	0.76
7	1.1	0.1	0.029	0.031	1	0.019	0.019	0.77
9	2.2	0.2	0.052	0.053	1.3	0.024	0.023	0.77
11	3.8	0.3	0.072	0.079	1.5	0.026	0.029	0.78
13	6.2	0.4	0.1	0.11	2	0.032	0.034	0.84
15	9.3	0.5	0.1	0.13	1.8	0.036	0.04	0.81
17	13.4	0.7	0.12	0.15	2	0.049	0.049	0.86
18	15.8	0.8	0.13	0.16	2.2	0.047	0.054	0.91

D.3 Fit Parameter

For all extrapolations of SPIKE's runtimes and all network settings the following power-function model was used:

$$f(x) = a \cdot x^b + c$$

The parameters in Tables D.8 and D.9 were computed in Matlab 2021a (9.10.0.1602286) with the Trust-Region algorithm with a maximum of 400 iterations. All coefficients are given with 95 % confidence bounds. The fitting was perfomed based on timings measured in milliseconds.

Pairs	Comm. [MiB]		Setup Phase [s]			Online Phase [s]		
Pairs	Setup	Online	A	B	C	A	B	C
Phase 3: Cycle Evaluation								
3	0.3	0	0.0061	0.0068	0.1	0.022	0.022	0.42
5	3.4	0.2	0.06	0.071	0.6	0.089	0.1	0.53
7	17.9	0.6	0.2	0.21	2	0.22	0.27	0.71
9	49.1	1.2	0.54	0.56	4.8	0.53	0.63	1.1
11	172.4	2.6	1.8	2	16	1.7	1.7	2.2
13	1,005.9	9.1	11	12	89	9.1	9.1	9.6
15	1,773.8	12.8	19	21	160	16	16	16
17	4,213.3	26.5	47	50	370	38	38	39
18	6,735.8	42	79	82	590	65	65	65
Phase 4: Solution Evaluation								
3	0	0	0.0024	0.0022	0.011	0.0038	0.0059	0.22
5	0.1	0	0.0083	0.0082	0.32	0.014	0.014	0.75
7	0.5	0.1	0.016	0.017	0.54	0.039	0.04	1.9
9	0.8	0.1	0.024	0.025	0.64	0.057	0.056	2.6
11	5.5	0.4	0.078	0.087	1	0.19	0.19	5.5
13	202.7	6.9	0.81	1.4	14	2.4	2.5	22
15	300	9.5	1.1	1.8	18	3.6	3.7	27
17	1,200.1	30.9	4.1	6	64	15	16	53
18	2,783.8	64.8	9.2	13	150	36	36	87

Table D.6: Comparison of the communication costs and setup and online runtimes of SPIKE for the three networking settings A: LAN with 10 Gbit/s, B: LAN with 1 Gbit/s, C: WAN, for cycle length $L = 3$. This table contains the results of Phases 3 and 4 (Cycle and Solution Evaluation).

Table D.7: Comparison of the setup and online runtimes of SPIKE for the reduced medical factor compatibility matching and the full set in the LAN and WAN networking setting A: LAN with 10 Gbit/s, C: WAN.

Pairs	Comm. [MiB]		Setup Phase [s]		Online Phase [s]	
Pairs	Setup	Online	A	C	A	C
Reduced Medical Factor Set						
2	0.1	0	0.0084	0.34	0.045	3
50	14.9	0.3	0.14	1.7	0.26	3.4
100	59.8	1.1	0.29	4.4	0.81	4.4
150	134.7	2.5	0.55	8.5	1.9	5.8
200	239.5	4.4	0.91	15	3.8	7.7
250	374.4	6.9	1.4	23	6.4	11
300	539.2	9.9	2	31	9.4	14
350	734	13.4	2.5	41	14	20
400	958.8	17.5	3.2	53	18	26
450	1,213.6	22.1	4.2	65	25	32
500	1,498.3	27.3	5.3	80	31	37
550	1,813.1	33	6.3	96	38	48
600	2,157.8	39.3	7.2	110	45	56
650	2,532.5	46.1	9	130	53	64
Full Medical Factor Set						
2	0.1	0	0.013	0.88	0.047	3.4
50	44	11.8	0.51	4.6	1	5.2
100	177.1	47.1	1.3	14	4.7	12
150	399.2	105.9	2.8	29	12	24
200	710.5	188.3	5.1	48	22	41
250	1,110.9	294.3	7.6	71	35	64
300	1,600.4	423.8	12	100	51	92
350	2,179.1	576.8	14	140	66	120
400	2,846.8	753.4	18	180	86	160
450	3,603.7	953.5	23	230	110	200
500	4,449.6	1,177.2	28	280	140	250
550	5,384.7	1,424.4	35	340	170	300
600	6,408.9	1,695.2	41	410	200	350
650	7,522.2	1,989.5	48	480	240	420

Table D.8: Fit parameters for the total runtime of SPIKE with cycle length $L = 3$

Parameter	LAN	Confidence Bounds	WAN	Confidence Bounds
a	5.437×10^{-6}	$(-1.1162 \times 10^{-5}, 2.249 \times 10^{-2})$	0.000259	$(-0.000441, 0.00096)$
b	8.484	(7.392,9.576)	7.658	(6.718, 8.598)
c	1,721	(−3,373,7,815)	1.335×10^4	$(-8{,}887, 3.3559 \times 10^4)$
r^2	0.9907		0.9915	
RMSE	7,167		3.042×10^4	

Table D.9: Fit parameters for the comparison of SPIKE and the state-of-the-art with cycle length $L = 3$

Parameter	LAN, 1 Gbit/s	Confidence Bounds
a	1.032×10^{-5}	$(-9.383 \times 10^{-7}, 2.159 \times 10^{-5})$
b	8.285	(7.906,8.663)
c	329.3	(−1,314,1,973)
r^2	0.9994	
RMSE	2,079	

Chapter E
Record Linkage Supplementary Material

E.1 Full Benchmark Tables

Database	Comm. [MiB]			Setup Phase [s]			Online Phase [s]		
Size	#Rounds	Setup	Online	A	B	C	A	B	C
GMW circuit variant									
1	370	2.7	0	0.047	0.1	0.85	0.11	0.17	19
10	530	25.2	0.4	0.19	0.42	1.5	0.15	0.24	27
25	570	62.5	1	0.35	1.9	2.1	0.17	0.24	29
100	650	248.8	3.9	1.2	10	5.5	0.23	0.37	33
250	690	621.2	9.8	3	27	12	0.32	0.48	36
1,000	770	2,483.4	39.3	12	110	44	0.63	1.7	40
2,500	850	6,207.9	98.3	29	270	110	1.3	4.2	45
10,000	930	24,830.6	393	120	1,100	450	3.9	17	53
25,000	970	62,076.2	982.5	300	2,700	1,100	8.8	44	66
GMW circuit variant with arithmetic conversions									
1	266	0.6	0.1	0.018	0.036	0.72	0.052	0.054	13
10	330	5.5	0.7	0.097	0.15	1.4	0.072	0.072	16
25	346	13.5	1.7	0.18	0.29	1.6	0.093	0.094	17
100	378	53.7	6.7	0.43	1.7	2.5	0.17	0.17	18
250	394	133.9	16.8	0.87	5.3	4	0.29	0.3	19
1,000	426	555.2	47.1	3	23	11	0.77	0.87	22
2,500	458	1,394.1	119.5	7.3	60	25	1.6	1.9	27
10,000	490	5,577.4	459.4	28	240	96	6.1	8.2	48
25,000	506	13,917.9	1,150.3	69	610	240	15	23	88
Yao circuit variant									
1	5	0	2.4	0.055	0.09	0.15	0.065	0.092	0.85
10	5	20.3	3.8	0.26	0.3	0.67	0.2	0.34	1.8
25	5	52.8	7.6	0.61	0.61	1.1	0.38	0.7	2.8
100	5	227.3	42.3	2.1	2.2	2.5	1.2	2.5	8.7
250	5	576.8	119.2	4.9	4.6	5.2	2.9	6.4	20
1,000	5	1,905.1	558.8	17	19	19	14	25	76
2,500	5	4,762.1	1,430.8	43	52	44	35	64	190
10,000	5	19,538.8	5,729.7	170	200	170	140	280	750
Yao circuit variant with arithmetic conversions									
1	40	0.1	0.6	0.017	0.033	0.4	0.022	0.026	1.7
10	76	1.2	5.8	0.084	0.1	0.94	0.099	0.11	3.1
25	85	11.1	6.4	0.18	0.21	1.2	0.16	0.2	3.5
100	103	52.3	17.7	0.63	0.7	2	0.42	0.59	5.4
250	112	139	40.2	1.3	1.9	3	0.93	1.3	8.6
1,000	130	554.4	231.1	4.6	10	7	3.8	5.9	24
2,500	148	1,412.2	546.2	12	27	16	9	16	54
10,000	166	4,860.8	2,285.6	42	110	60	39	67	200

Table E.1: Full comparison of the setup and online runtimes for the three networking configurations from Figure 5.5, for *varying database sizes*, and all four circuit protocol variants.

E.2 Field Configuration

Table E.2 displays the default configuration of Mainzelliste and following that *Mainzelliste Secure EpiLinker* (MainSEL). The comparison field indicates either "Equality" (Eq.) or "Bloom-Dice" (B.D.) comparison and the weight w is calculated according to $w = \log\left((1-e)/f\right)$.

Field name	Type	Comparison	Frequency f	Error Rate e	Weight w	Bitlength
First Name	String	B.D.	0.000235	0.01	12.04	500
Surname	String	B.D.	0.0000271	0.008	15.16	500
Birth name	String	B.D.	0.0000271	0.008	15.16	500
Day of birth	Integer	Eq.	0.0333	0.005	4.90	5
Month of birth	Integer	Eq.	0.0833	0.002	3.58	4
Year of Birth	Integer	Eq.	0.0286	0.004	5.12	11
ZIP code	String	Eq.	0.01	0.04	6.58	40
City	String	B.D.	0.01	0.04	6.58	500

Table E.2: Default EpiLink field configuration used in the reported benchmarks.

Acknowledgements

First and foremost, I would like to thank my Ph.D. supervisor Kay Hamacher, without whom this work would not have been possible. Thank you for the opportunity to conduct my research in your group. This gratitude also extends to my co-supervisor Thomas Schneider, without the valuable guidance and feedback from both of you, and of course, the funding of my research, all of my academic achievements would not be possible.

Many thanks to all former and current colleagues in the Computational Biology & Simulation group. Thank you for the fascinating discussions and activities during four wonderful years in which I have learned a lot. I thank all my collaborators who made the projects what they are and whose involvement was always enriching. In particular, many thanks to Alex, Sebastian, and Daniel, who have always been there for me in questions about cryptography and academic matters, and to Uli and Maren who helped me in the medical realm.

My academic career was significantly shaped by Felix and Barbara – many thanks for that.

I owe thanks to my family Heike and Lena, Eva, Ben, Klaus-Dieter, Anette, David, Lisa, Lio, Hanna, and Friedhelm for their support, patience, and encouragement.

I am very grateful for the people who proofread this dissertation: Eva, Heike, Lara, and Steffen.

And finally, a huge thanks to my friends who put up with me during this time (in alphabetical order): Andy, Heiko, Kathi, Lara, Laura, Nathali, Sebastian, Simon, Steffen, and Tobi.

Thank you all!

Ehrenwörtliche Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit entsprechend den Regeln guter wissenschaftlicher Praxis selbstständig und ohne unzulässige Hilfe Dritter angefertigt habe.

Sämtliche aus fremden Quellen direkt oder indirekt übernommenen Gedanken sowie sämtliche von Anderen direkt oder indirekt übernommenen Daten, Techniken und Materialien sind als solche kenntlich gemacht. Die Arbeit wurde bisher bei keiner anderen Hochschule zu Prüfungszwecken eingereicht. Zu einem vorherigen Zeitpunkt ist noch keine Promotion versucht worden.

Die eingereichte elektronische Version stimmt mit der schriftlichen Version überein.

Datum und Unterschrift