ATLAS: Datentreuhänder für anonymisierte Analysen in kommunalen Datenräumen

Teilvorhaben

Kryptographische Verfahren für eine pseudonyme und datenschutzfreundliche Datennutzung

Hasso-Plattner-Institut, Universität Potsdam

Verbundpartner

Hasso Plattner Institut, Universität Potsdam (HPI)
Technische Universität Berlin (TU Berlin)
Polyteia GmbH
SINE e.V.
KIProtect GmbH
Verband Region Rhein-Neckar (K.d.ö.R)(VRRN)

Förderkennzeichen

16KISA037

Projektlaufzeit

15.12.2022 – 14.12.2025 (36 Monate)

With funding from the:
Federal Ministry of Research, Technology and Space

Funded by the European Union
NextGenerationEU

Inhalt

1 Ziele des ATLAS-Projekts

Die öffentliche Verwaltung in Deutschland ist dezentral organisiert. Daten werden somit ebenfalls verteilt erhoben und gespeichert, was oft eine effiziente Entscheidungsfindung einschränkt. Insbesondere auf kommunaler Ebene wird datenbasierte Entscheidungsfindung durch zwei strukturelle Probleme maßgeblich eingeschränkt:

1. Daten werden anwendungsbezogen erhoben. Sogenannte Fachverfahren speichern ihre Daten getrennt voneinander in isolierten „Silos“. Vor einer Analyse müssen Daten aus mehreren dieser Silos mit hohem Aufwand zusammengeführt werden.
2. Beim Zusammenführen der Daten entstehen neue Bedenken, insbesondere bezüglich des Datenschutzes. Die daraus resultierenden rechtlichen Unklarheiten führen zu erhöhtem Projektaufwand und Verzögerungen.

Eine mögliche organisatorische Antwort ist die Einführung eines sogenannten Datentreuhänders. Das ist eine Institution, die Daten treuhänderisch zusammenführt, um sie für Analysezwecke verfügbar zu machen. Eine intuitive Umsetzung eines Datentreuhänders wirft jedoch eine Reihe von Bedenken hinsichtlich Datenschutz und Datensicherheit auf. Gerade im Falle personenbezogener kommunaler Daten besteht ein erhöhter Schutzbedarf, der nicht allein durch organisatorische Maßnahmen gedeckt werden kann. Aus rein softwaretechnischer Sicht können sensible Daten durch Sicherheitslücken oder andere Missbrauchsformen unbefugt zugänglich gemacht werden. Für den adäquaten und sicheren Betrieb eines Datentreuhänders sind daher kryptographische Sicherungsmechanismen, Analysen und Vorgehensmodelle erforderlich, um ein hohes technisches Schutzniveau für sensible Datenklassen herzustellen. Solche Schutzmaßnahmen sind notwendig, um die Erwartungen der Bürger:innen an einen Datentreuhänder zu erfüllen.

Ziel des Forschungsvorhabens ATLAS ist die Konzeption, Entwicklung und Implementierung von Infrastruktursoftware für den Betrieb eines sicheren und datenschutzfreundlichen Datentreuhänders im Kontext kommunaler Daten. Hierzu wurde ein interdisziplinäres Konsortium gebildet, das folgende übergeordnete Arbeitsschritte verfolgt:

1. Ein Use-Case-getriebenes Vorgehen, bei dem Anwendungsszenarien und Datenanalysen gemeinsam mit dem kommunalen Projektpartner definiert und entwickelt werden.
2. Die Anpassung, Optimierung und Weiterentwicklung kryptographischer und datenschutzfreundlicher Verfahren für einen einfachen und effizienten Einsatz.
3. Die formale Modellierung und Verifikation der Sicherheits- und Datenschutzeigenschaften des Datentreuhänders sowie Metriken zur Risikobestimmung.
4. Die Implementierung des Datentreuhänders als Open-Source-Software auf Basis der im Projekt erforschten Verfahren und definierten Protokolle.
5. Die experimentelle Integration und Anwendung der Datentreuhändersoftware in bestehende Datenanalysesoftware für kommunale Anwender:innen und Daten.

1.1 Ziele des Teilvorhabens

Ziel des Teilprojektes war die Entwicklung und Optimierung beweisbar sicherer kryptographischer Protokolle für eine pseudonyme und datenschutzfreundliche Datennutzung des im Projekt geplanten Datentreuhänder Systems. Dazu gehören die formale Modellierung der geforderten Sicherheits- und Privacy-Eigenschaften und die Entwicklung und Analyse praktikabler kryptographischer Protokolle, die diese Eigenschaften beweisbar unter wohldefinierten Annahmen erreichen, als auch die Umsetzung in prototypischen Implementierungen. Das Teilprojekt hat dabei Lösungen für zentrale als auch dezentrale Ansätze untersucht.

Zentralisierte und dezentrale Berechnungsmodelle

Der Projektplan sah eine systematische Untersuchung sowohl **zentralisierter** als auch **dezentralisierter** Datentreuhänder-Architekturen für datenschutzfreundliche Datenanalysen vor. Beide Paradigmen wurden parallel betrachtet, um ihre jeweilige Sicherheit, Effizienz und Anwendbarkeit auf reale kommunale Use Cases zu bewerten.

In der **zentralisierten Architektur** ist der Datentreuhänder direkt für die Datenbereitstellung zuständig; die Herausforderung besteht darin, das Risiko missbräuchlicher Zugriffe durch den Datentreuhänder selbst oder durch Datennutzende zu reduzieren. In dieser Architektur bildet ScrambleDB [Leh19] die technische Grundlage für eine sichere Datenkonsolidierung In ScrambleDB werden Daten zentral, jedoch in einer pseudonymen und nicht verknüpfbaren Form gespeichert; zum Wiederherstellen von Verknüpfungen ist die Autorisierung durch eine dritte Partei erforderlich. Der Forschungsschwerpunkt lag auf der Weiterentwicklung des ScrambleDB Protokolls um die Daten die bei der Zusammenführung der pseudonymen Datenpunkte entstehen, weiter zu minimieren. Ein weiteres Ziel war die Modellierung von Privacy-Leakage, die Definition geeigneter Risikometriken sowie die quantitative Bewertung von Datenschutzrisiken durch selektive Korrelation, u.a. in Zusammenarbeit mit der TU Berlin.

Im **dezentralen Setting** verbleiben die Daten in den Infrastrukturen der Kommunen, und Analysen werden durch direkte Interaktion dieser Infrastrukturen durchgeführt. Der Datentreuhänder ist in diesem Modell „virtuell“, und die Rolle einer dritten Partei beschränkt sich auf Bereitstellung von Software und grundlegender Kommunikationsinfrastruktur. Diese Architektur sollte auf generischen Secure-Multi-Party-Computation (MPC)Protokollen basieren; deren Untersuchung und Implementierung wurden überwiegend durch den Partner SINE e.V. durchgeführt. Dieser Ansatz bietet starke Privacy-Garantien, da eine Zentralisierung vermieden wird, allerdings ist er typischerweise mit erheblichem Rechen- und Kommunikationsaufwand verbunden. Ziel hierbei war die Effizienz der aktuellen Protokolle zu verbessern, u.a., durch Kombination mit den Pseudonymisierung-Ansätzen aus ScrambleDB.

Früh im Projektverlauf zeigte sich, dass dezentrale Ansätze für die Use-Case-Partner – vor allem aus regulatorischer und datenschutzrechtlicher Sicht – deutlich attraktiver sind. Gleichzeitig wurde der Betrieb einer interaktiven und rechenintensiven Infrastruktur als Herausforderung identifiziert. Diese Erkenntnis führte dazu, dass die HPI-Forschungsgruppe

den Fokus in Richtung dezentraler Modelle und insbesondere auf deren Effizienz und Deployability verlagerte. Eine daraus entstandene Forschungsrichtung ist die Untersuchung **hybrider Berechnungsmodelle**, die eine Teilzentralisierung ohne Privacy-Verlust erlauben, indem Datenverarbeitung von Vertrauen und Schlüsselbesitz entkoppelt wird. In diesem Modell stellt der Datentreuhänder zentrale Infrastruktur für Datenanalysen bereit, bleibt jedoch gegenüber den verarbeiteten Daten kryptographisch „blind".

Insgesamt zielt das Teilvorhaben darauf ab, eine rigorose und zugleich praktisch tragfähige Grundlage für datenschutzfreundliche Datentreuhänder-Architekturen zu schaffen. Durch die Kombination kryptographischer Pseudonymisierung, formaler Sicherheitsgarantien, quantitativer Risikobewertung sowie zentralisierter, dezentraler und hybrider Berechnungsmodelle ermöglicht es Datenanalysen, die mit strengen Datenschutzauflagen vereinbar sind. Die Ergebnisse werden als Open-Source-Komponenten entwickelt und durch konkrete kommunale Use Cases getrieben, was sowohl wissenschaftliche Relevanz als auch praktische Anwendbarkeit sicherstellt.

2 Projektorganisation und -durchführung

2.1 Finanzierung und Ressourcen

Die dem Teilvorhaben zugewiesenen Ressourcen wurden überwiegend für Personalkosten eingesetzt. Die Personalmittel finanzierten eine wissenschaftliche Mitarbeitendenstelle, die die zentralen wissenschaftlichen Arbeiten des Projekts abdeckte, insbesondere kryptographische Modellierung, Protokollentwicklung und formale Sicherheitsanalyse. Zusätzlich wurde eine studentische Hilfskraft finanziert, um Forschungs- und Implementierungsaufgaben zu unterstützen. Sachmittel wurden für Konferenzreisen verwendet, auf denen Projektergebnisse präsentiert wurden und zur Dissemination, Diskussion und Validierung der Forschungsergebnisse in der Fachcommunity beitrugen. Eine detaillierte Liste der präsentierten wissenschaftlichen Ergebnisse findet sich in Abschnitt 4.

2.2 Beschreibung der Arbeitspakete

Dieser Abschnitt gibt einen Überblick über die Arbeiten des Projekts und hebt die Beiträge des HPI hervor. Das Projekt war in sieben – teilweise parallel laufende – Arbeitspakete gegliedert.

AP1 – Anforderungsanalyse

Leitung: Polyteia **Beteiligte Partner:** VRRN, HPI, TU Berlin, KIProtect, SINE

AP1 legt das konzeptionelle Fundament des Projekts, indem kommunale Use Cases in konkrete Systemanforderungen übersetzt werden. Im Fokus stehen das Verständnis realer Anwendungsszenarien, die Identifikation relevanter Datenquellen sowie die Ableitung übergeordneter technischer und datenschutzbezogener Anforderungen. Dieses Arbeitspaket

stellt sicher, dass nachfolgende technische Entwicklungen fest in praktischen Anforderungen und Stakeholder-Erwartungen verankert sind.

Ergebnisse: Das HPI brachte sich aktiv in die frühe konzeptionelle Phase des Projekts ein, insbesondere durch die Teilnahme an gemeinsamen Workshops mit allen Projektpartnern und kommunalen Stakeholdern. Diese Aktivitäten dienten der Identifikation und Schärfung relevanter Use Cases, der Klärung technischer Möglichkeiten und Grenzen sowie der Übersetzung anwendungsgetriebener Anforderungen in abstrakte Systemanforderungen. In dieser Phase stellte das HPI sicher, dass sowohl zentralisierte (ScrambleDB-basierte) als auch dezentrale (MPC-basierte) Architekturansätze von Beginn an systematisch berücksichtigt wurden.

Diese frühe Einbindung etablierte ein gemeinsames Verständnis des Systemumfangs und beeinflusste die nachfolgenden Aktivitäten zur Architekturentwicklung und Sicherheitsmodellierung. Insbesondere zeigte sich früh, dass für die identifizierten Use-Case-Anforderungen ein dezentraler Ansatz – oder zumindest eine stark dezentrale Systemarchitektur – passend ist, um geltende Datenschutzvorgaben einzuhalten. Gleichzeitig ergaben die Diskussionen, dass die Begrenzung von Rechen- und Kommunikationsaufwand eine zentrale praktische Anforderung ist: Kommunen würden erhebliche Schwierigkeiten haben, Lösungen mit übermäßigem Ressourcenbedarf zu betreiben.

In diesem Kontext beobachtete das Konsortium zudem ein klares Interesse der Stakeholder an Cloud-nativen Architekturen, die Helferdienste nutzen, um die Betriebskomplexität zu reduzieren. Daraus ergab sich eine zusätzliche Designanforderung: traditionell Peer-to-Peer-basierte Techniken – wie Secure Multi-Party Computation – mit Deployment-Modellen zu vereinbaren, die zentrale oder teilzentrale Helfer-Komponenten einbeziehen, ohne starke Sicherheits- und Privacy-Garantien aufzugeben.

AP2 – Modellierung von Sicherheitsanforderungen

Leitung: HPI **Beteiligte Partner:** Polyteia, TU Berlin, KIProtect, SINE
AP2 definiert die formalen Sicherheits- und Privacy-Ziele des Gesamtsystems und seiner Kernkomponenten. Es liefert eine strukturierte Abstraktion der Systemfunktionalität und etabliert präzise Sicherheitsziele unter klar spezifizierten Angreifermodellen. Dieses Arbeitspaket legt die theoretische und konzeptionelle Grundlage für spätere kryptographische Design-, Implementierungs- und Evaluationsarbeiten im Projekt.

Ergebnisse: Das HPI leitete und prägte die formale Modellierung der Sicherheits- und Datenschutzanforderungen für das ATLAS-System. Diese umfasste die Definition abstrakter Systemfunktionalitäten und Angreifermodelle, die unabhängig von konkreten Implementierungsentscheidungen sind.

Zu Beginn konzentrierte sich das HPI vor allem auf die Spezifikation integritätsbezogener Sicherheitsanforderungen für datenschutzfreundliche Datentreuhänder-Architekturen, mit besonderem Fokus auf Authentisierung und Verifizierbarkeit in verteilten und ausgelagerten

Settings. Motiviert durch die Beobachtung, dass Datentreuhänder die Datenverarbeitung an externe oder semi-vertrauenswürdige Parteien delegieren, analysierte das HPI, wie Integritäts- und Authentizitätsgarantien erreicht werden können, ohne Ziele wie Datensparsamkeit und Privacy zu kompromittieren. Dies führte zu einer formalen Klärung der erforderlichen Sicherheits- und Privacy-Eigenschaften und zeigte Grenzen bestehender Ansätze auf [LÖ24].

Anschließend identifizierte das HPI eine grundlegende Privacy-Herausforderung in ScrambleDB die bei einem Einsatz in dezentralen Anwendungen deutlich verstärkt wird: die Join-Funktionalität, d.h., das Zusammenfügen korrelierter Daten aus verteilten Datensätzen würde in einem dezentralen Setting deutlich mehr Informationen offenbaren als beabsichtigt. Die Beobachtung führte zur Formulierung einer dedizierten Multi-Party-Private-Join (MPPJ) Funktionalität. Dieses formale Sicherheitsmodell lieferte die Grundlage für die anschließende Forschung an konkreten Protokolldesigns und ScrambleDB-Erweiterungen, die MPPJ praktisch sicher realisieren können. Das Protokoll wurde im Rahmen von AP4 entwickelt und gemeinsam mit dem formalen Sicherheitsmodell in [LMS26] publiziert.

AP3 – Modellierung der Risikobewertung

Leitung: TU Berlin **Beteiligte Partner:** HPI, KIProtect, SINE
AP3 adressiert die systematische Bewertung technischer Datenschutzrisiken. Ziel ist es, quantitative Metriken und Risikoprofile zu entwickeln, die eine strukturierte Evaluation der in der Datentreuhänder-Architektur inhärenten Privacy-Risiken erlauben. Das Arbeitspaket verbindet kryptographisches Systemdesign mit messbarer Risikoanalyse und liefert einen analytischen Rahmen für fundierte Designentscheidungen.

Ergebnisse: Das HPI unterstützte die Entwicklung von Methoden zur Datenschutz-Risikobewertung, insbesondere mit Blick auf ScrambleDB-basierte Datentreuhänder-Architekturen. Das HPI stand dazu mit Partnern der TU-Berlin im aktiven Austausch, welche das Thema federführend bearbeitete, und half bei der Einordnung kryptographischer Verfahren und gab Feedback zu den Ergebnissen.

AP4 – Weiterentwicklung der Technologien

Leitung: SINE **Beteiligte Partner:** HPI, TU Berlin, KIProtect
AP4 fokussiert die konzeptionelle und technische Weiterentwicklung der für den Datentreuhänder benötigten kryptographischen Verfahren und Privacy-Enhancing Technologies (PETs). Ziel ist es, kontrollierte, datenschutzfreundliche Analysen sensibler Daten – sowohl in zentralisierten als auch dezentralisierten Settings – bei gleichzeitig starken Sicherheitsgarantien zu ermöglichen. Dieses Arbeitspaket bildet den Hauptteil der Forschungsbeiträge des HPI ab.

Ergebnisse: Das HPI entwarf, analysierte und erweiterte kryptographische Protokolle für zentralisierte und dezentrale Datentreuhänder-Architekturen. Zu den zentralen Beiträge gehört die Erweiterung von ScrambleDB auf dezentrale bzw. hybride Umgebungen, mit stärkeren Sicherheitsgarantien als in der ursprünglichen zentralen Design-Formulierung und mit dem

daraus abgeleiteten Begriff von Multi-Party-Private-Join-Protokollen [LMS26]. Zudem wurden kryptographische Protokolle für dezentrale Berechnungen entwickelt, einschließlich einer neuen Konstruktion auf Basis vollhomomorpher Verschlüsselung (FHE) namens Helium [MCPT24]. Zudem wurden hybride Systemmodelle untersucht, die eine zentrale Verarbeitung verschlüsselter Daten mit einer dezentralen Schlüsselkontrolle kombinieren [MCPT24, MCNL24]. Darüber hinaus wurden Techniken zur Sicherstellung der Datenintegrität bei gleichzeitiger Wahrung der Privatsphäre erforscht und weiterentwickelt – sowohl in dezentralen Architekturen als auch im delegierten Datentreuhänder-Setting. [LÖ24, HDL+25, LÖ25, LNÖ25]. Sämtliche Arbeiten wurden auf internationalen wissenschaftlichen Konferenzen präsentiert und werden in Kapitel 3 ausführlicher vorgestellt.

AP5 – Implementierung und experimentelle Anwendung des Systems

Leitung: Polyteia **Beteiligte Partner:** VRRN, HPI, TU Berlin, KIProtect, SINE
AP5 überführt die konzeptionellen und kryptographischen Entwicklungen in ein nutzbares Softwaresystem. Ziel ist es, die entwickelten Technologien in eine Open-Source-Implementierung zu integrieren, die in bestehende kommunale Dateninfrastrukturen eingebettet werden kann. Das Arbeitspaket bereitet das System für den praktischen Einsatz in realen administrativen Umgebungen vor.

Ergebnisse: Die Implementierungsarbeiten am HPI konzentrierten sich auf die folgenden Hauptaufgaben:

- Implementierung des Multi-Party-Private-Join-Protokolls [LMS26].
- Fortführung der Entwicklung der Helium-Framework-Implementierung [MCPT24].
- Implementierung eines Prototyps für Private-Set-Intersection auf Basis der Helium-Implementierung [MCNL24].
- Unterstützung und Code-Review der von SINE durchgeführten ScrambleDB-Implementierung.

Verzögerungen im Einstellungsprozess im Konsortium verschoben die geplante Prototyp-Implementierung. Dadurch wurde der finale Systemprototyp später als vorgesehen fertiggestellt. Als Reaktion darauf lenkte das HPI die vorgesehenen Ressourcen in Richtung Implementierung zusätzlicher kryptographischer Bausteine um, insbesondere des Multi-Party-Private-Join-Protokolls. Dadurch konnten die Projektziele weiterhin erreicht werden, indem die technischen Grundlagen des Systems gestärkt und wiederverwendbare Komponenten für zukünftige Prototypen und Folgearbeiten bereitgestellt wurden.

AP6 – Verifizierung und Evaluierung

Leitung: Polyteia **Beteiligte Partner:** VRRN, HPI, TU Berlin
AP6 evaluiert das entwickelte System hinsichtlich Sicherheit, Datenschutz, Performance und praktischer Nutzbarkeit. Ziel ist es, die Tragfähigkeit der kombinierten Technologien unter realistischen Bedingungen zu bewerten und strukturierte Schlussfolgerungen zu Stärken sowie

Grenzen und zukünftige Anwendungspotenziale abzuleiten. Das Arbeitspaket unterstützt sowohl die technische Validierung als auch die strategische Verwertungsplanung.

Ergebnisse: Das HPI hat die Sicherheit aller entwickelten kryptographischen Verfahren durch formale mathematische Beweise nachgewiesen und verifiziert. Darüber hinaus führte das HPI ein Code-Review der von SINE entwickelten ScrambleDB-Implementierung durch und leistete damit einen Beitrag zur weiteren Stärkung der Softwarequalität. Obwohl ursprünglich vorgesehen war, im dritten Projektjahr vollständige Prototypsysteme und fortgeschrittene Spezifikationen zur Verifikation durch das HPI bereitzustellen, kam es bei der Fertigstellung dieser Spezifikationen zu Verzögerungen. Infolgedessen wurden die vorgesehenen Ressourcen seitens des HPI auf die Verifikation generischer Bausteine umgelenkt, einschließlich des oben genannten Code-Reviews.

AP7 – Projektmanagement und Koordination

Leitung: Polyteia **Beteiligte Partner:** VRRN, HPI, TU Berlin
AP7 stellt die Koordination über alle Projektpartner und Arbeitspakete hinweg sicher. Ziel ist es, strukturierte Zusammenarbeit, Abstimmung technischer Aktivitäten und die fristgerechte Erreichung von Meilensteinen zu unterstützen. Dieses Arbeitspaket liefert selbst keine technischen Ergebnisse, bildet jedoch die Grundlage für die effektive Durchführung des gesamten Projekts.

Ergebnisse: Das HPI nahm an regelmäßigen Projektmeetings und technischen Abstimmungen teil. Diese Beiträge stellten Konsistenz zwischen formaler Modellierung, Protokollentwicklung und Systemimplementierung über die Partner hinweg sicher.

3 Detaillierte Ergebnisse des Teilvorhabens

Dieser Abschnitt fasst die wesentlichen Ergebnisse zusammen, die das HPI im Rahmen des ATLAS-Projekts erzielt hat. Der Fokus lag auf der Entwicklung neuer kryptographischer Verfahren und der Implementierung von Open-Source-Artefakten für eine pseudonyme und datenschutzfreundliche Datennutzung in Datentreuhänder Systemen.

3.1 Wissenschaftliche Ergebnisse

Die im Projekt entwickelten Protokolle und Sicherheitsmodelle wurden in Form von peer-reviewed Publikationen auf internationalen Konferenzen publiziert. Die konkreten Forschungsergebnisse des HPI lassen sich dabei entlang von drei wesentlichen Themen kategorisieren, die wir im Folgenden vorstellen werden: (1) Verbesserungen des ScrambleDB Einsatzes, insbesondere bei Einsatz in hybriden Architekturen (2) Verbesserung der Effizienz von MPC-Techniken und (3) Techniken zur Sicherung der Datenintegrität in verteilten System.

ScrambleDB mit verteilten Datenquellen und hybrider Architektur

Ein zentraler technischer Ausgangspunkt des Teilvorhabens ist das ScrambleDB-Protokoll [Leh19], das eine kryptographisch kontrollierte Join-Operation auf pseudonymisierten Datensätzen erlaubt. ScrambleDB schützt Daten, indem relationale Tabellen in individuell pseudonymisierte Attribute zerlegt werden, die isoliert betrachtet nicht personenbezogen sind und in der Form nicht verknüpft werden können. Erst während der Ausführung einer konkreten Join-Query wird mittels kryptographischer Schlüssel diese Verknüpfung wiederhergestellt, aber nur für die konkrete Untermenge der Informationen, die in mehreren Eingabetabellen auftrat und innerhalb des Joins liegt. Während dieser Ansatz starken Schutz „at rest“ bietet und die Nutzbarkeit der Daten durch kontrollierte Joins erhält, wurde er ursprünglich für ein zentrales Setting entworfen, in dem alle Datenaufnahmen und Abfragen durch eine einzige Partei erfolgen. Eine direkte Erweiterung auf dezentrale Szenarien, bei denen die Daten verteilt auf mehreren Parteien liegen und wiederum von einer weiteren Partei aggregiert und korreliert abgefragt werden können, würde zu schwächeren Privacy-Garantien führen: Es können Informationen über Daten jenseits der am Join beteiligten Attribute abgeleitet werden. Daher war es ein zentrales Projektziel, Protokolle zu entwickeln, die optimale Privacy auch für solche Ansätze mit verteilter Datenhaltung erlauben.

Die angestrebte Funktionalität ist die eines *Multi-Party Private-Join* (MPPJ) Protokolls. Ein MPPJ-Protokoll ermöglicht es mehreren Datenquellen, für eine Empfängerpartei die Inner Joins über ihre jeweiligen Datensätze bereitzustellen, während so wenig Informationen wie möglich preisgegeben werden. Zum damaligen Zeitpunkt gab es kein Protokoll, das ein solches MPPJ direkt und effizient über das Zwei- oder Drei-Parteien-Setting hinaus ermöglichte. Alle zuvor bekannten Protokolle erreichten entweder eine schwächere (z. B. Multi-Party-Private-Set-Intersection) oder eine allgemeinere Funktionalität (z. B. Private-Join-Compute und generische Secure-Multi-Party-Computation) und waren dadurch erheblich komplexer.

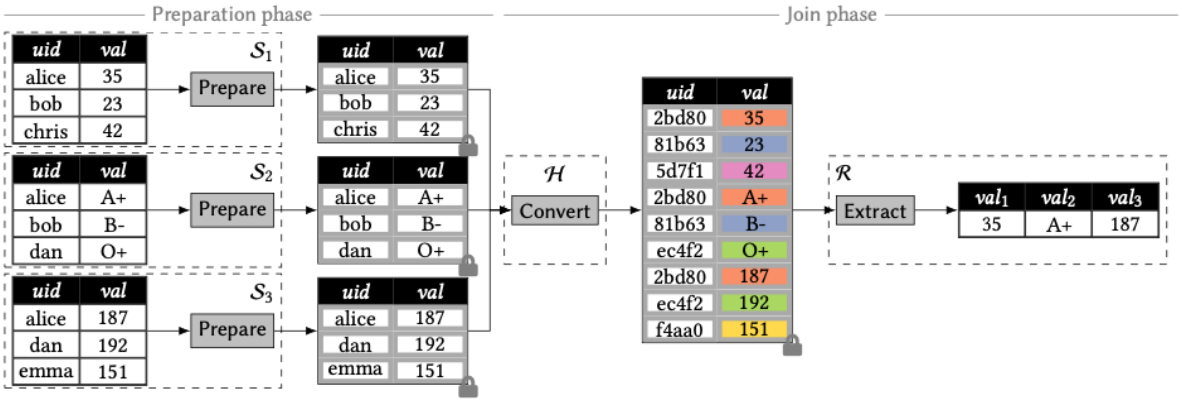
Daher untersuchte die HPI-Forschungsgruppe das generelle Problem privater Joins in Multi-Party-Setting. Die Arbeit wurde auf der PETS 2026 Konferenz angenommen und als [LMS26] publiziert. Eine Vorversion dieser Arbeit wurde zudem im Hauptevent des Forschungsclusters in Form eines Posters präsentiert. Eine zentrale Beobachtung von [LMS26] ist, dass sich eine dedizierte MPPJ Konstruktion mit nur einer Interaktionsrunde pro Partei realisieren lässt, was das Protokoll in der Praxis besonders einfach einsetzbar macht.

Multi-Party Private Join (MPPJ). Die zentrale Operation, die wir mit unserem MPPJ Protokoll berechnen wollen, ist ein *anonymer Join*. Ein solches Verfahren kombiniert n assoziative Tabellen, von denen jede eine Menge von Identifikatoren auf einen zugehörigen Wert abbildet, zu einer einzigen Tabelle. In dieser zusammengeführten Tabelle sind nur die Werte, für die der zugehörige Identifikator in allen Eingabetabellen vorhanden ist. Wichtig ist, dass die Ausgabetabelle korreliert (d.h. Werte mit übereinstimmenden Identifikatoren werden einander zugeordnet), aber zugleich anonym ist, d.h. sie enthält die Identifikatoren nicht mehr.

Wir interessieren uns für das Problem, anonyme Joins zwischen n verteilten Quellen zu berechnen, die den anonymen Join ihrer lokalen Datensätze einem Empfänger bereitstellen

möchten. Dabei soll der Empfänger keine Werte außerhalb des Joins erfahren, da diese typischerweise für die Studie oder Anwendung irrelevant sind. Außerdem darf der Join die eindeutigen Identifikatoren nicht offenlegen, da dies dem Ziel anonymer Joins widersprechen würde. Schließlich sollen die Quellen auch nichts über die Datensätze der jeweils anderen Quellen erfahren.

Das von uns entwickelte Multi-Party Private Join Protokoll setzt dieses Ziel um und ermöglicht es Quellen S1, ..., Sn, die jeweils eine Datenbanktabelle bestehend aus Datenzeilen (uid, val) besitzen, einem Empfänger R den anonymen Join über ihre Tabellen bereitzustellen, wobei auf die Unterstützung einer Helper-Partei H zurückgreift. Die Abbildung unten illustriert die Haupt-Operation unseres MPPJ-Protokolls.



Übersicht über unser MPPJ-Protokoll

MPPJ gewährleistet die folgenden Eigenschaften:

- **Source Privacy:** Die Quellen dürfen keine Information über die Eingabedaten anderer Quellen erfahren. Insbesondere sollen sie keine Information über die Schnittmenge erfahren, die implizit als Teil des Joins berechnet wird.
- **Helper Obliviousness:** Der Helper darf keine Information über die Input-Datensätze (abgesehen von "gutartigen" bzw. vertretbaren Leakage, wie der Größe der Inputs) und keine Information über die von ihm unterstützte Join-Berechnung erfahren.
- **Anonymous Join:** Der Empfänger erfährt nur die anonym zusammengeführten Werte val. Damit erfährt er weder die zugrunde liegenden Identifikatoren uid der zusammengeführten Werte noch Werte außerhalb des Joins, d. h. Werte, die zu Identifikatoren gehören, die nicht in allen Tabellen vorhanden sind.
- **One-Way Flow:** Das Protokoll erfordert keine Kommunikation der Quellen untereinander und nur einen einfachen, verschlüsselten Upload der Daten von den Quellen zum Helfer. Für n Datenbanken mit m Zeilen benötigt das Protokoll nur einen einzigen $O(m)$-Upload der Quellen zum Helper und einen einzigen $O(n \cdot m)$-Download vom Helper zum Empfänger. Der Helper ist vollständig "blind": er ermöglicht Effizienz und einfache Deployability, erfährt jedoch keine Information über die Join-Berechnung oder ihre Inputs.

Die Grundidee des Protokolls ist wie folgt: Wir lassen die Quellen lediglich eine verschlüsselte Version ihrer lokalen Daten unter Verwendung eines öffentlichen Schlüssels des Empfängers an den Helper hochladen. Anschließend verlassen wir uns darauf, dass der Helper eine weitere Verschlüsselungsschicht hinzufügt, wobei alle Dateneinträge unter uid-spezifischen Schlüssel verschlüsselt werden, die von R nur für Werte innerhalb des Joins wiederhergestellt werden kaan. Die größte Herausforderung besteht darin, diese zweite Verschlüsselungsschicht und die Schlüsselwiederherstellung hinzuzufügen, ohne dass Helper und Empfänger etwas über die zugrunde liegenden Identifikatoren erfahren. Dies erreichen wir durch eine sorgfältige Kombination (teilweise) homomorpher Verschlüsselung, wie etwa ElGamal, und Ideen aus (3-Parteien-) Oblivious Pseudorandom Functions (OPRF). Die OPRF wird – ähnlich wie in ScrambleDB – eingesetzt, um aus den Identifikatoren uid Pseudonyme nym abzuleiten die bei Umwandlung und Entschlüsselung später deterministisch gleich sind für gleiche Identifier, aber keine Rückschlüsse auf den Identifier selber zulassen. Zusätzlich nutzen wir die OPRF auch für die Schlüsselableitung der uid-spezifischen Schlüssel.

MPPJ-Protokoll Idee. Der wesentliche Ablauf des Protokolls ist dabei wie folgt (für Details verweisen wir auf [LMS26]):

Prepare (Quellen S): Jede Datenquelle verarbeitet ihre Tabelle zunächst lokal. Für jeden Tabelleneintrag (uid, val) wird der Nutzwert val mit dem öffentlichen Schlüssel bpk des Empfängers verschlüsselt. Zusätzlich wird der zugehörigen Identifier uid (bzw. die gehashte Version $H(uid)$) separat unter bpk verschlüsselt, was die Vorstufe der OPRF darstellt. Die Quellen übermitteln ausschließlich diese verschlüsselten Daten an die Helper-Partei. Eine direkte Interaktion zwischen den Quellen ist nicht erforderlich.

Convert (Helper H): Der Helper übernimmt anschließend die zentrale, aber komplett blinde Vorbereitung für die Zusammenführung. Er versieht die bereits unter dem Empfängerschlüssel verschlüsselten Werte mit einer zusätzlichen kryptographischen Schutzschicht. Diese zweite Schicht wird so konstruiert, dass ihre erfolgreiche Entfernung nur dann möglich ist, wenn ein Identifier tatsächlich in allen beteiligten Tabellen vorkommt. Technisch wird dies durch eine Kombination aus partiell homomorpher Verschlüsselung und OPRF-basierten Mechanismen zur blinden Schlüsselableitung realisiert.

Insbesondere transformiert der der Helper die verschlüsselten Identifier $H(uid)$ in verschlüsselte Pseudonyme $nym = H(uid)^k$ für einen Schlüssel k der für die Protokolldurchführung vom Helper frisch erstellt wurde. Die verbleibende zentrale Herausforderung ist das weitere Verarbeiten der verschlüsselten val Daten: Der Empfänger soll nur Werte entschlüsseln können, die tatsächlich im Join enthalten sind. Unser Protokoll erreicht dies durch eine zweite OPRF-Nutzung zur Generierung uid-spezifischer One-Time Pads. Für jedes Tupel zieht der Helper ein frisches per-Tupel-Secret s und verschlüsselt den Wert val unter einem symmetrischen Schlüssel $H(s)$, während er separat eine Verschlüsselung von s unter einem Schlüssel pad erstellt. Diesen Schlüssel pad soll der Empfänger aber nur dann erhalten, wenn ein vollständiger Join über alle Quellen hinweg existiert. Dazu berechnet der Helper ein additives Secret-Sharing und hängt für jede Quelle einen Share von pad an, sodass pad nur dann rekonstruierbar ist, wenn alle n Shares für dasselbe Pseudonym vorliegen. Hierzu

nutzen wir die Key-Homomorphie der Hash-DH-OPRF, d.h. ziehen uid-spezifische Pseudonyme in die Berechnung ein. Das stellt sicher, dass der Empfänger genau dann (und nur dann), wenn er n Tupel mit demselben nym findet, pad rekonstruieren, das zugehörige s entmaskieren und die Werte entschlüsseln kann; andernfalls bleiben die Werte maskiert und nicht entschlüsselbar.

Der entscheidende Effekt dieses Designs ist, dass die Join-Logik implizit in der Schlüsselstruktur verankert ist. Für jeden Identifier entsteht faktisch ein gemeinsamer Entschlüsselungskontext, der nur bei vollständiger Übereinstimmung aller Parteien rekonstruierbar ist. Fehlt ein Beitrag einer Quelle, bleibt das entsprechende Ergebnis für den Empfänger kryptographisch unentschlüsselbar. Nach Abschluss der Verarbeitung übermittelt der Helper die resultierenden Ciphertexte an den Empfänger.

Extract (Empfänger R): Der Empfänger entschlüsselt die Ciphertexte zunächst mit seinem geheimen Schlüssel bsk, entfernt also zuerst die innere Verschlüsselung Schicht. Er erhält dadurch Tuple von Pseudonymen und weiteren Ciphertexten und Key Shares. Die Pseudonyme wurden durch den Helper konsistent abgeleitet, d.h. uid −Werte die in verschiedenen Quellen vorkamen, wurden konsistent und deterministisch in den gleichen Pseudonym-Wert nym überführt. Damit erhält der Empfänger konsistente Pseudonyme und kann Tupel nach Gleichheit von nym gruppieren – analog zu ScrambleDB, nun jedoch über mehrere Quellen hinweg. Anschließend kann der Empfänger die assoziierten Daten entschlüsseln: für alle Tuple deren uid in allen Quellen vorkam, hat der Empfänger alle Shares von pad erhalten. Damit kann er den vollen Schlüssel s rekonstruieren und anschließend die verschlüsselten Daten entschlüsseln, d.h., erfolgreich ist dies ausschließlich für diejenigen Datensätze, die echte Join-Treffer darstellen. Der Empfänger erhält somit direkt die korrekt ausgerichteten Wertetupel der Schnittmenge, jedoch ohne Zugriff auf die zugrunde liegenden Identifier und ohne Informationen über Einträge außerhalb der Schnittmenge.

Das Protokoll wurde formal modelliert und seine Sicherheit unter Standardannahmen im passiven Angreifermodell bewiesen. Das Protokoll garantiert, dass jede Partei nur die ihr bestimmte Information erhält: Der Empfänger lernt ausschließlich das Join-Ergebnis, der Helper verarbeitet zu jedem Zeitpunkt nur verschlüsselte Daten, und die einzelnen Quellen erhalten keinerlei Einblick in fremde Datensätze.

Hinsichtlich der Effizienz zeichnet sich der Ansatz dadurch aus, dass die Datenquellen nur eine einmalige Upload-Phase durchführen müssen und keine aufwändigen paarweisen MPC-Interaktionen zwischen allen Parteien erforderlich sind. Die Hauptlast der Verarbeitung liegt beim Helper, wodurch das Verfahren gut auf eine größere Anzahl von Datenquellen skalieren kann. Insgesamt stellt das Protokoll damit einen praktisch orientierten und datenschutzstarken Ansatz zur Berechnung anonymer Multi-Party-Joins dar, der insbesondere für organisationsübergreifende Datenkooperationen geeignet ist.

MPPJ-Implementierung & Evaluation. Zusätzlich zur Protokollentwicklung wurde eine Open-Source-Implementierung und eine umfangreiche Performance-Evaluierung bereitgestellt.

Experimentelle Ergebnisse zeigen einen klaren Vorteil gegenüber alternativen Protokollen. Die folgenden zwei Tabellen fassen diese Resultate zusammen.

Vergleich zwischen unserem MPPJ-Protokoll und einem generischen MPC-Protokoll (MP-SPDZ/ATLAS) für n = 2 Quellen und m Zeilen pro Tabelle.

m	MP-SPDZ/ATLAS		π_{MPPJ} (% of MP-SPDZ/ATLAS)	
	Time (s)	Comm (MiB)	Time (s)	Comm (MiB)
10^1	0.12	1.06	0.18 (150.478%)	0.01 (0.744%)
10^2	10.18	90.73	0.21 (2.067%)	0.08 (0.087%)
10^3	1027.97	9072.27	0.49 (0.048%)	0.79 (0.009%)

Benchmarks unseres MPPJ-Protokolls für n Quellen und m Zeilen pro Tabelle sowie Vergleich mit einem Private-Join-Compute-Protokoll. Die Join-Größe ist auf 80 % der Tabellenzeilen festgelegt.

n	Time in seconds					Total Communication in MiB (Percentage of IDCloak)				
	$m=1353$	$m=1700$	$m=19735$	$m=45211$	$m=253680$	$m=1353$	$m=1700$	$m=19735$	$m=45211$	$m=253680$
2	0.59	0.68	5.36	11.95	66.43	1.07 (59%)	1.34 (15%)	15.55 (60%)	35.61 (98%)	199.83 (83%)
3	0.78	0.92	7.83	17.60	98.24	1.60 (42%)	2.01 (10%)	23.32 (39%)	53.42 (64%)	299.75 (54%)
4	0.95	1.14	10.28	23.25	130.13	2.13 (35%)	2.68 (8%)	31.09 (30%)	71.23 (49%)	399.67 (41%)
5	1.12	1.36	12.76	28.94	162.24	2.66 (31%)	3.35 (6%)	38.86 (24%)	89.04 (40%)	499.58 (33%)
6	1.32	1.57	15.24	34.69	194.65	3.20 (28%)	4.02 (5%)	46.64 (21%)	106.84 (34%)	599.50 (28%)

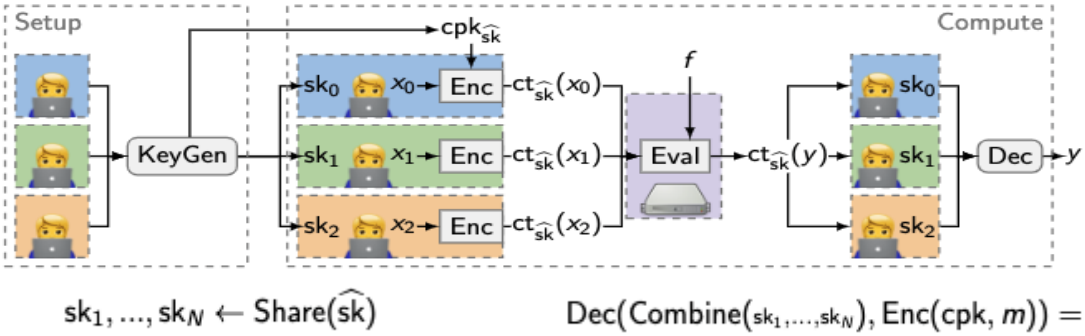
Wir konnten beobachten, dass die Implementierung unseres MPPJ-Protokolls zwischen 1,02x und 20x geringere Kommunikationskosten benötigt als IDCloak [CNS+25] – eines der jüngsten Private-Join-Compute-Protokolle (ohne Berechnung über dem Join) – für 2 bis 6 Parteien und Datenbankgrößen von 1,5K bis 250K Einträgen.

Effiziente MPC-Protokolle im hybriden Setting

Im Projekt wurde Multi-Party Computation auf Basis von Threshold-Fully-Homomorphic-Encryption als vielversprechender Ansatz untersucht, um generische MPC mit geringen Anforderungen an die teilnehmenden Parteien zu ermöglichen. In threshold-FHE-basierter MPC (siehe Abbildung unten) generieren Dateninhaber gemeinsam kryptographisches Schlüsselmaterial und verschlüsseln ihre Inputs unter einem kollektiven Public Key, sodass Berechnungen direkt auf Ciphertexts ausgeführt werden können. Die Entschlüsselung von Ergebnissen wird kryptographisch über Threshold-Mechanismen kontrolliert, während der Großteil von Berechnung und Kommunikation an einen honest-but-curious Helfer delegiert werden kann. Dies führt zu geringer Interaktionskomplexität für die Teilnehmenden, diese müssen nur einen kleinen, persistenten Zustand halten können.

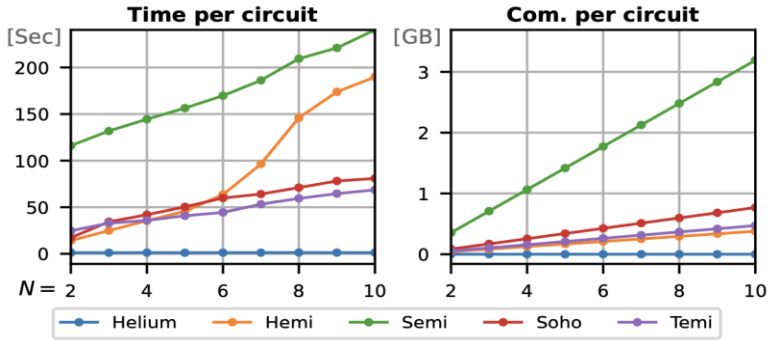
Das Helium-Framework geht zentrale Herausforderungen an, die bislang den Praxiseinsatz von FHE-basierter MPC limitiert haben, insbesondere die Unterstützung ressourcenbeschränkter Teilnehmender und Robustheit unter Netz-Churn. Zu diesem Zweck reformuliert die Arbeit die Ausführung FHE-basierter MPC-Protokolle systematisch in nicht-monolithische, neustartfähige Sub-Protokolle mit öffentlich aggregierbaren Transkripten. Es wird ein neuer Ausführungsmechanismus eingeführt, der Korrektheit, Sicherheit und “Lebendigkeit” selbst bei häufigen Verbindungsabbrüchen und Node-Ausfällen sicherstellt, ohne auf

nicht-kryptographische Annahmen wie Trusted Hardware oder Nichtabsprache der Parteien zurückzugreifen. Die Ergebnisse wurden in [MCPT24] veröffentlicht.



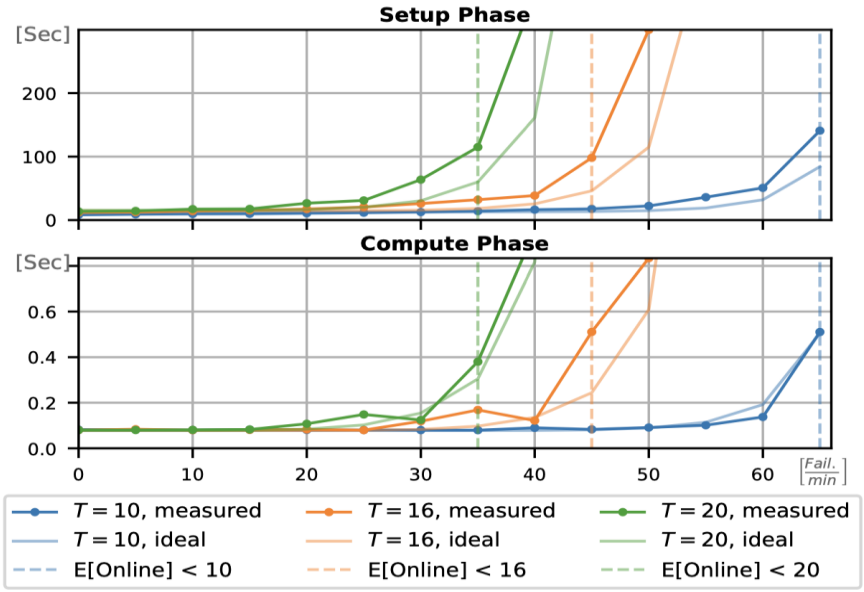
Übersicht über das Threshold-FHE-basierte MPC-Protokoll. KeyGen und Dec sind Mehrparteienprotokolle, Enc und Eval sind die üblichen (lokalen) FHE-Operationen.

Das resultierende Framework Helium ist die erste Open-Source-Implementierung generischer FHE-basierter MPC, die Churn toleriert und Lightweight Clients unter Standardannahmen unterstützt. Experimentelle Ergebnisse zeigen, dass Helium skalierbare MPC mit Dutzenden Parteien bei geringen Speicher- und Bandbreitenanforderungen ermöglicht und gegenüber secret-sharing-basierten MPC-Frameworks (ohne Churn) Performance-Verbesserungen um Größenordnungen erzielt.



Vergleich der pro Partei anfallenden Rechenzeit (links) und Kommunikationskosten (rechts) einer 512×512 Matrix-Vektor-Multiplikation für N Parteien mit mehreren passiv-sicheren Protokollen, die in MP-SPDZ implementiert sind. Mittelwert über 10 Durchläufe.

Darüber hinaus treibt Helium den Stand der Technik in generischer MPC voran, indem es Netz-Churn explizit unterstützt, inklusive wiederholter Disconnections und Reconnections teilnehmender Knoten. Im Gegensatz zu bestehenden ausfalltoleranten MPC-Protokollen, die getrennte Parteien typischerweise als dauerhaft verloren behandeln, erlaubt Helium Knoten, einem laufenden Setup wieder beizutreten und an neu ausgestellten Schaltkreis-Berechnungen aktiv mitzuwirken. “Lebendigkeit” wird durch eine T-out-of-N-Threshold-FHE-Konstruktion erreicht: Sicherheit gilt, solange höchstens T−1 Parteien korrumpiert sind, und Fortschritt ist möglich, sobald mindestens T Parteien online sind. Die Abbildung unten zeigt Heliums Performance bei unterschiedlichen Churn-Raten und belegt stabile Durchsatz- und Latenzeigenschaften selbst in hochdynamischen Netzwerken.



Latenz der Setup- und Compute-Phasen für steigende Ausfallraten und unterschiedliche Threshold-Werte T**. Die Anzahl der Knoten beträgt** N = 30**, und jeder Knoten folgt einem unabhängigen markovschen Failure-Recovery-Prozess mit einer festen Wiederverbindungszeit von 20 s.** E[Online] **ist die erwartete Anzahl online befindlicher Parteien im Gleichgewicht. Die Idealzeit ist die Latenz ohne Churn, dividiert durch** Pr[Online ≥ T]**.**

Wir zeigen, dass Helium Churn-tolerant ist: Für Ausfallraten unterhalb der E[Online] ≥ T-Schwelle (gestrichelte Linien) liegt die Latenz nahe am idealen Wert. Zudem ist die Latenz, sobald Pr[Online ≥ T] nahe null ist, nahezu linear in der Ausfallrate mit sehr kleiner Steigung. Das liegt daran, dass nur wenige Parteiausfälle tatsächlich einen Sub-Protokoll-Ausfall verursachen (etwa weil abstürzende Parteien ihre Nachrichten bereits geliefert haben oder im aktuellen Protokollabschnitt gar nicht beteiligt sind). Diese Beobachtung wird dadurch gestützt, dass der Faktor mit T ansteigt (was die Wahrscheinlichkeit erhöht, dass eine gegebene Partei an einem gegebenen Protokoll beteiligt ist).

Helium wird als erweiterbares Open-Source-Software-Framework entwickelt, das auf der quelloffenen Lattigo-Bibliothek für homomorphe Verschlüsselung aufbaut. Seine modulare Architektur ist auf Wiederverwendung und Erweiterbarkeit ausgelegt und erleichtert die Integration zusätzlicher Protokolle, Ausführungsmodelle und anwendungsspezifischer Funktionalitäten über den Umfang des Projekts hinaus. Dadurch wird erwartet, dass Helium eine wachsende Bandbreite an Use Cases in datenschutzfreundlicher Datenanalyse und sicherer verteilter Berechnung unterstützt.

Das Framework und seine zugrunde liegenden Konzepte wurden als Poster auf der IEEE EuroS&P 2025 präsentiert [Mou25].

Darüber hinaus wurde ein Helium-basierter Use Case, der an den Zielen von ATLAS ausgerichtet ist, in Form eines **Multi-Party-Private-Set-Intersection-Protokolls** untersucht. Diese Konstruktion nutzt das Helfer-unterstützte Ausführungsmodell in ähnlicher Weise wie das MPPJ-Protokoll, arbeitet jedoch im strengeren passiven MPC-Setting. Im Gegensatz zu

Ansätzen, die auf Nichtabsprache-Annahmen beruhen, erfordert das Protokoll keine Vertrauensaufteilung zwischen Helfer und Output-Empfänger und liefert damit stärkere und standardnähere Sicherheitsgarantien (unterstützt jedoch – anders als MPPJ – keine assoziierten Werte). Dieses Ergebnis wurde als Poster auf zwei Konferenzen präsentiert:

Zusammen mit dem Helium-Framework etablieren diese Ergebnisse FHE-basierte MPC als tragfähige Grundlage für praktische, low-overhead und dezentrale datenschutzfreundliche Berechnung im Kontext des ATLAS-Projekts.

Datenintegrität in dezentralen Settings

Datenschutzfreundliche Authentisierung ist ein zentraler Baustein in einer verteilten System-Architektur. In einem dezentralen Setting, wie den hier untersuchten Datentreuhändern, sind Vertrauen und Verantwortung inhärent verteilt, und diese Struktur muss sich auch in den Authentisierungsmechanismen widerspiegeln – insbesondere im Design digitaler Signaturverfahren. Zu diesem Zweck untersuchten wir Multi-Signatur-Verfahren, die es mehreren Parteien erlauben, Daten durch Aggregation individueller Signaturen und Public Keys in eine kurze aggregierte Signatur bzw. einen aggregierten Public Key gemeinsam zu authentisieren. Wir identifizierten die spezifischen Sicherheits- und Privacy-Eigenschaften, die für solche Verfahren in ad-hoc- und privacy-sensitiven Umgebungen erforderlich sind. Wir modellierten die identifizierten Sicherheits- und Privacy-Eigenschaften, zeigten, dass bestehende Verfahren diese Anforderungen nicht (vollständig) erfüllen, und entwickelten Erweiterungen, die die gewünschten Eigenschaften beweisbar erreichen. Diese Arbeit wurde in [LÖ24] publiziert. Aufbauend auf [LÖ24] demonstriert [HDL+25], wie sich solche datenschutzfreundlichen Signaturschemata auf Basis schwächerer und insbesondere quantensicherer Annahmen realisieren lassen. In [LÖ25] zeigten wir, wie sich Multi-Signaturen in ausfallsicherere Threshold-Signatur-Verfahren umwandeln lassen. Die Arbeit [LNÖ25] zeigt, wie Signaturen nicht nur verteilt, sondern auch blind berechnet werden können, was den Datenschutz weiter erhöht.

3.2 Software-Ergebnisse

ATLAS legte bezüglich Software-Artefaktes Wert auf Reproduzierbarkeit und Wiederverwendbarkeit. Das HPI trug dazu durch Feedback und Code-Review der im Konsortium entwickelten ScrambleDB- und Polytune-Implementierungen bei. Darüber hinaus entwickelte das HPI mehrere eigenständige Software-Artefakte, die zentrale kryptographische Bausteine implementieren. Die unten aufgeführten Implementierungen unterstützen die Reproduzierbarkeit der wissenschaftlichen Ergebnisse und liefern wiederverwendbare Komponenten für zukünftige Forschung und Systementwicklung.

`https://github.com/hpicrypto/mppj`
Dieses Repository enthält die Referenzimplementierung des HPI für das

Multi-Party-Private-Join (MPPJ) Protokoll aus [LMS26] als Go-Modul. Das Paket implementiert zentrale Typen und Funktionen zur Durchführung privater Joins in einem Multi-Party-Setting, wie in der Protokollspezifikation beschrieben. Es stellt eine API für Kodierung, Streaming und Austausch verschlüsselter Zeilendaten zwischen Teilnehmenden und Helper-Server über gRPC-Schnittstellen bereit.

https://github.com/ChristianMct/helium

Helium ist ein Prototyp des MPC-Frameworks auf Basis von Multiparty Homomorphic Encryption (MHE), wie in [MCPT24] beschrieben. Es ist in Go implementiert und bietet eine Schnittstelle zur homomorpher Multi-Party-Ausführung Circuits; es verwaltet die zugrunde liegenden MHE-Protokolle sowie den Netzwerktransport via gRPC. Die meisten kryptographischen Operationen werden durch die Lattigo-Bibliothek unterstützt. Das Framework ermöglicht Helfer-unterstützte Ausführung und damit MPC unter ressourcenbeschränkten Teilnehmenden mit Unterstützung eines honest-but-curious Servers. Beispielanwendungen – etwa Multi-Party-Vektor-Multiplikation – demonstrieren die Nutzung des Frameworks für Ende-zu-Ende-Abläufe sicherer Berechnung.

3.3 Publikationsübersicht

Die Projektergebnisse wurden in 7 Konferenzbeiträgen sowie in 4 Poster-Präsentationen veröffentlicht. Eine Übersicht der Konferenzen, auf denen die jeweiligen Arbeiten vorgestellt wurden, ist nachfolgend aufgeführt:

Konferenzbeiträge:

[LMS26] *Privacy Enhancing Technologies Symposium (PETS) 2026.*
[LNÖ25] *IACR Eurocrypt 2025.*
[LÖ25] *IACR Public-Key Cryptography 2024.*
[HDL+25] *IACR Public-Key Cryptography 2024.*
[MCPT24] *ACM Conference on Computer and Communications Security (CCS) 2024.*
[LÖ24] *IACR Public-Key Cryptography 2024.*
[GL23] *Privacy Enhancing Technologies Symposium (PETS) 2023.*

Poster:

[LMS25] *Kongress des Forschungsnetzwerk – Anonymisierung (AnoSiDat) 2025*
[Mou25] *10th IEEE European Symposium on Security and Privacy (IEEE EuroS&P) 2025*
[MCNL24] *ACM Conference on Computer and Communications Security (CCS) 2024 und* Nationale Konferenz IT-Sicherheitsforschung 2025

4 Verwertungsplan und Ausblick

4.1 Wissenschaftliche Verwertung und erwarteter Nutzen

Die im Projekt erzielten Ergebnisse liefern substanziellen wissenschaftlichen Nutzen, indem sie wiederverwendbare Modelle, Methoden und kryptographische Bausteine für datenschutzfreundliche Datenanalyse in dezentralen und zentralisierten Datentreuhänder-Architekturen etablieren. Diese Ergebnisse sind direkt in weiterer wissenschaftlicher Arbeit nutzbar, da sie auf abstrakter und technologieunabhängiger Ebene formuliert sind und dadurch in eine breite Palette von Forschungskontexten jenseits der ursprünglichen Anwendungsszenarien übertragbar sind.

Die wissenschaftliche Verwertbarkeit wird durch eine systematische Dissemination der Ergebnisse über internationale peer-reviewte Publikationen und Präsentationen auf führenden Konferenzen sichergestellt – sowohl während der Projektlaufzeit als auch im Zeitraum nach Projektabschluss. Zum Zeitpunkt der Berichterstellung sind alle wissenschaftlichen Publikationen angenommen; das jüngste Ergebnis [LMS26] wird auf dem Privacy Enhancing Technologies Symposium im Juli 2026 präsentiert.

Insgesamt stärkte und erweiterte das Projekt die Kernkompetenzen des HPI in Kryptographie und Privacy-Enhancing Technologies und lieferte eine nachhaltige wissenschaftliche Grundlage für Anschlussforschung.

4.2 Technische und wirtschaftliche Verwertung und erwarteter Nutzen

Aus technischer und wirtschaftlicher Sicht sind die Projektergebnisse als modulare und wohldefinierte Komponenten verwertbar, die sich in fortgeschrittene Softwaresysteme zur datenschutzfreundlichen Datenverarbeitung integrieren lassen. Die entwickelten kryptographischen Protokolle und Systemabstraktionen sind mit Blick auf praktische Anwendbarkeit konzipiert; die kritischsten davon wurden hinsichtlich Sicherheit und Performance in realistischen Systemsettings validiert.

Die enge Zusammenarbeit mit den Projektpartnern ermöglicht den Transfer dieser Ergebnisse in konkrete Softwarelösungen, insbesondere im Kontext von Datentreuhänder-Architekturen für kommunale und administrative Daten. Die Nutzbarkeit der Ergebnisse wird durch ihre Ausrichtung an realen Use Cases, durch Prototyp-Implementierungen sowie durch ihre Kompatibilität mit bestehenden Dateninfrastrukturen unterstützt. Dadurch können die Projektpartner auf den Ergebnissen für weitere Produktentwicklung und Praxiseinsatz aufbauen, ohne grundlegende Redesigns vornehmen zu müssen.

Über den unmittelbaren Projektkontext hinaus liefern die Ergebnisse eine technologische Basis für Folge-Entwicklungen, einschließlich Erweiterungen in Richtung hybrider Systemmodelle und zusätzlicher Anwendungsdomänen. Dies schafft günstige Voraussetzungen für eine

längerfristige wirtschaftliche Verwertung – etwa durch Integration in bestehende Produkte, durch Folge-Innovationsprojekte oder durch Adoption in breiteren Datenökosystemen. Damit trägt das Projekt zur Stärkung technologischer Expertise und Innovationsfähigkeit im Bereich datenschutzfreundlicher Datenanalyse bei.

4.3 Ausblick

Das ATLAS Projekt führte zur Entwicklung neuer kryptographischer Konzepte und Protokolle für datenschutzfreundliche Datenanalyse. Diese Ergebnisse bilden eine solide und wiederverwendbare Grundlage für weitere Forschung und Entwicklung im Bereich sicherer Datentreuhänder und Privacy-Enhancing Technologies. Gleichzeitig lieferte ATLAS wertvolle Einsichten in die praktischen Herausforderungen beim Transfer fortgeschrittener kryptographischer Techniken in reale administrative Umgebungen. Obwohl es gelang, einen funktionalen Prototyp des Datentreuhänders zu implementieren und auszurollen, geschah dies mit einem einzigen Use-Case-Partner und in einer späteren Projektphase als ursprünglich geplant. Dadurch waren Möglichkeiten zur iterativen Verfeinerung des Prototyps und zu systematischem Feedback der akademischen Partner auf Basis mehrerer Deployment-Zyklen begrenzt. Diese Erfahrungen haben sowohl technische als auch organisatorische Einschränkungen verdeutlicht, die für eine breitere Adoption adressiert werden müssen.

Ein naheliegender nächster Schritt ist die weiterführende Untersuchung von Architektur Optionen zwischen vollständig zentralisierten und vollständig dezentralisierten Datentreuhänder-Modellen. Insbesondere sind hybride Lösungen weiter zu erforschen, die die Effizienz zentraler Datenverarbeitung mit der Vertrauensminimierung dezentraler Ansätze kombinieren. Dazu gehören Architekturen, die auf partiell vertrauenswürdigen oder oblivious Helfer-Entitäten beruhen – wie im Multi-Party Private Join (MPPJ)-Ansatz – sowie die Integration solcher Komponenten in End-to-End-Workflows für Datenanalyse.

Literaturverzeichnis

[CNS+25] Shuyu Chen, Guopeng Lin, Haoyu Niu, Lushan Song, Chengxun Hong, Weili Han. IDCloak: A Practical Secure Multi-party Dataset Join Framework for Vertical Privacy-preserving Machine Learning. https://arxiv.org/abs/2506.01072 2025.

[Dwo+06] *Cynthia Dwork. Differential Privacy. Automata, Languages and Programming (ICALP) 2006.*

[GL23] *Tarek Galal, Anja Lehmann. Privacy-Preserving Outsourced Certificate Validation. Privacy Enhancing Technologies Symposium (PETS) 2023.*

[HDL+25] *Calvin Abou Haidar, Dipayan Das, Anja Lehmann, Cavit Özbay, Octavio Perez-Kempner. Privacy-Preserving Multi-Signatures: Generic Techniques and Constructions Without Pairings. Public-Key Cryptography (PKC) 2025.*

[Leh19] *Anja Lehmann. ScrambleDB: Oblivious (Chameleon) Pseudonymization-as-a-Service. Privacy Enhancing Technologies Symposium (PETS) 2019.*

[LMS25] *Anja Lehmann, Christian Mouchet, Andrey Sidorenko. Poster: ATLAS Project – Multi-party Private Join. Kongress des Forschungsnetzwerk – Anonymisierung (AnoSiDat) 2025*

[LMS26] *Anja Lehmann, Christian Mouchet, Andrey Sidorenko. Multi-party Private Join. Privacy Enhancing Technologies Symposium (PETS) 2026*

[LNÖ25] *Anja Lehmann, Phillip Nazarian, Cavit Özbay. Stronger Security for Threshold Blind Signatures. IACR Eurocrypt 2025*

[LÖ24] *Anja Lehmann, Cavit Özbay. Multi-Signatures for Ad-hoc and Privacy-Preserving Group Signing. IACR Public-Key Cryptography (PKC) 2024.*

[LÖ25] *Anja Lehmann, Cavit Özbay. Commit-and-Prove System for Vectors and Applications to Threshold Signing. Public-Key Cryptography (PKC) 2025.*

[MCNL24] *Christian Mouchet, Sylvain Chatel, Lea Nürnberger, and Wouter Lueks. Poster: Multiparty Private Set Intersection from Multiparty Homomorphic Encryption. ACM Conference on Computer and Communications Security (CCS) 2024.*

[MCPT24] *Christian Mouchet, Sylvain Chatel, Apostolos Pyrgelis, and Carmela Troncoso. Helium: Scalable MPC among Lightweight Participants and under Churn. ACM Conference on Computer and Communications Security (CCS) 2024.*

[Mou25] *Christian Mouchet. Poster: Helium: A Framework for FHE-based MPC. 10th IEEE European Symposium on Security and Privacy (IEEE EuroS&P) 2025.*