PATDISCOVER: Privacy-Preserving Discoverability of Patients

Jan Pennekamp*,§, Johannes Lohmöller*, Niels Pressel*, Sandra Geisler†, Felix M. Mottaghy‡, Klaus Wehrle*
*Communication and Distributed Systems, RWTH Aachen University, Germany · {lastname}@comsys.rwth-aachen.de
†Data Stream Management and Analysis, RWTH Aachen University, Germany · geisler@cs.rwth-aachen.de
‡Nuclear Medicine, RWTH Aachen University Hospital, Germany · fmottaghy@ukaachen.de
§Center for Computational Life Sciences, RWTH Aachen University, Germany

***Abstract*—Sourcing medical experience in finding effective patient-treatment strategies is challenged by strict privacy requirements these days. Specifically, the cross-institutional discovery of "similar" patients based on certain attributes, e.g., to align treatment strategies or collect clinical expertise on rare diseases, is currently either impossible, impractical, or it exposes sensitive data. Addressing this research gap, we propose PATDISCOVER, which is a fully homomorphic encryption-based design that supports multiple attribute types of medical importance, such as Enum, Range, and Distance. This way, institutions may compose and submit complex queries to several other institutions to discover relevant patients elsewhere. We evaluate PATDISCOVER extensively using real-world patient data from nuclear medicine and demonstrate its adequate performance, scalability, precision, and security for real-world use. In conclusion, our work enables the privacy-preserving discoverability of patients for various applications in healthcare (research) and beyond.**

***Index Terms*—information security; healthcare; rare diseases; distributed analysis**

1. Introduction

Improving the individual treatment of patients is a continuous challenge despite the shift to data-driven, evidence-based healthcare [1], irrespective of the disease or symptoms at hand. Indeed, various advances have been made, with electronic health records (EHRs) and related concepts being used more widely nowadays [2], [3]. These approaches facilitate a better and scalable monitoring of, for example, (personalized) treatment strategies, among other benefits. Accounting for the data's sensitivity [4]–[6], the desire for data sovereignty [7], and the expectation of a consent mechanism [8] is crucial in this context and must strictly be considered at all times. Hence, indexing, discovering, and exchanging patient data face multiple vital challenges.

A common approach to deal with the sensitivity of data and privacy of individuals is to aggregate information, e.g., by considering k-anonymity [9], applying differential privacy [10], or by utilizing distributed concepts like the personal health train [11]. By design, these approaches only give insights into patient statistics while abstracting individual attributes, thereby ensuring patient privacy. This aggregation level of detail only covers a subset of relevant applications in healthcare. However, both for research and treatment, knowing about "similar" patients across institutions is critical to, for example, assess an individual's likely response to specific drugs [12], [13], analyze patient trajectories under specific conditions [14], or reach out to compatible patients and/or practitioners in charge concerning new developments in the field [15]. Specifically, with rare diseases such as brain tumors [16], patient data is often scarce. This scarcity is accentuated by a limited number of cases per institution, making it challenging to utilize (traditional) statistical approaches on local data. We thus exemplarily motivate our research based on this use case in nuclear medicine.

Due to the benefits, there is a huge interest in identifying "similar" patients globally to strengthen the foundation for clinical and research decision-making. Similar patients must express specific features for being eligible for relevance. Given the sensitive nature of such data and the imperativeness of reliably protecting patient privacy, simply storing expressive patient data in unprotected central repositories is not an option. Thus, the domain of healthcare is in urgent need of a privacy-preserving alternative for the decentralized discovery of similar patients across institutions.

Conceptually, various building blocks come to mind when designing a solution for enhanced privacy, and indeed, related work mirrors this diversity. Secure multi-party computation, as explored in prior work [17], [18], offers robust privacy but requires significant precautions to ensure data availability while scaling poorly in settings with numerous institutions. In contrast, advanced centralized approaches like ObliDB [19] or databases employing searchable encryption [20]–[22] were shown to leak information on an individual level, making them unsuitable for our context. Recent encrypted database systems [23], [24] generally solve these privacy issues but implement a superset of the required query functionality, inducing non-negligible performance overheads. Lastly, approaches based on homomorphic encryption (HE) either suffer from limited query expressiveness [4], [25] or introduce minor but non-tolerable privacy leaks [26], highlighting a pressing and significant gap in research.

In this paper, we close this gap by proposing our performant, conceptually-centralized, HE-based approach,

PATDISCOVER, designed for the privacy-preserving discovery of patients across institutions. Specifically, we consider pressing requirements that currently hinder real-world use, i.e., data availability, scalability, query expressiveness, and privacy preservation. We opted for a straightforward design with reliable guarantees that focuses on the setting at hand. PATDISCOVER is versatile and scales to generic forms of patient discovery, i.e., beyond our evaluation in nuclear medicine, as well as other attribute-based queries with comparable requirements.

Contributions. Our primary contributions are as follows.

- PATDISCOVER is a privacy-preserving design for cross-institutional patient discovery that supports rich queries, promising personalized treatment plans for patients, among other benefits.
- Based on prior work, we carefully derive HE-based operators for efficient, approximated equality and less-than comparisons.
- Ethically building upon real-world patient data from nuclear medicine, we validate PATDISCOVER's feasibility of indexing and discovering rare-disease patients privacy-preservingly.

Open Science. We open-source a prototype of PATDISCOVER [27] to ensure reproducibility and reusability.

Organization. The remainder of this paper is structured as follows. First, in Section 2, we present the need and requirements for practical and privacy-preserving patient discovery. Specifically, we also refer to a real-world use case from nuclear medicine. Second, in Section 3, we detail fully homomorphic encryption as well as relevant related work before introducing our privacy-preserving design, PATDISCOVER, in Section 4. In Section 5, we report on our implementation and evaluation, which covers both performance and security aspects. Afterward, we discuss PATDISCOVER's impact in Section 6 and conclude in Section 7.

2. Scenario and Problem Statement

In Section 2.1, we present our considered scenario in detail, outlining how information would ideally flow for improved clinical practice and research. We augment this introduction with a real-world use case. Subsequently, in Section 2.2, we formalize this setting by (i) stating the corresponding threat model and (ii) compiling an elaborate list of information-security-specific research challenges.

2.1. Patient Treatment-Affecting Data Scarcity

Especially for rare diseases, treatment and clinical care would benefit from joining patient data from several institutions, i.e., establishing cross-institutional collaboration [28]. Likewise, when an institution is recruiting patients with certain features for a study, the local cohort might be too small, mandating the incorporation of patients from other institutions [22]. Hence, without a discovery approach, we observe data scarcity for individual institutions despite "similar" patients being treated globally (low numbers overall).

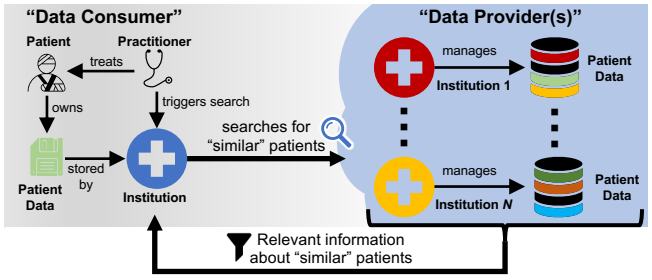

Figure 1. Using patient data, an institution wants to globally discover patients with "similar" features at other institutions.

Difficulty in Sharing Medical Data. A broad data foundation promises significant benefits for decision-making both in clinical practice and research. In addition to recent information, longitudinal data is also highly desired by practitioners since it may convey insights about patient trajectories. However, the discoverability of relevant patient data is severely hindered by two aspects. First, expressive identifiers might still be missing, for example, if a disease has not been (unambiguously) diagnosed or if existing frameworks like ICD-10 fail to capture crucial nuances accurately [29]. Second, privacy expectations and legislation restrict how patient data may be processed and shared—even within a single institution [30].

Involved Stakeholders. Assuming that these barriers have been overcome, the general setting of discovering patients for the sake of sharing medical data is as follows. We have an institution that is looking for "similar" patients and thus defines a list of attributes that should be satisfied precisely or approximately when considering relevancy. The expressiveness of the attributes depends on the use case, but they may cover matches in enumerations, Booleans, or range queries, among others. In practice, either a patient who is being treated or the parameters of a research project define what "similar" refers to in the specific context. Such a query would then be submitted internally or to third-party institutions for processing. Using the communicated attributes, these parties can check their *local* patient data for relevancy so that they can respond with the number or even a list of relevant patients.

Terminology from data ecosystems would translate the described roles as "data consumer", querying the data, and "data provider(s)", holding the data, respectively [31], as we visualize in Figure 1. The number of data consumers and providers is flexible ($N \geq 1$). For a practical realization, queries to different institutions should be independent of each other to not introduce processing dependencies.

Exemplary Use Case: Nuclear Medicine. Our work is motivated by a highly-specific use case in healthcare but also generalizes to other clinical contexts (cf. Section 6). Compared to other diseases, the oncological treatment of tumors is highly individual, depending on the tumor type, location, and progression, among other aspects. Thus, data on nuclear imaging and therapy is limited to "few" patients globally and annually, which are even further split across a small number of institutions or specialized practitioners.

TABLE 1. RELEVANT ATTRIBUTES IN THE AREA OF NUCLEAR MEDICINE.

Name	Type	Value Range
Diagnosis		
Active Tumor Tissue	Boolean	Yes, No
Tumor Progression	Boolean	Yes, No
Tumor Information		
1p/19q Codeletion	Boolean	Yes, No
IDH Wild-Type	Boolean	Yes, No
MGMT Promoter Meth.	Boolean	Yes, No
WHO Grade	Enum	I, II, III, IV
Tumor Position	Distance	$\{x \in \mathbb{R} \mid 10x \in \mathbb{Z} \wedge 0 \leq x \leq 4\}^3$
Tumor Type	Enum	Glioblastoma, Oligodendroglioma, Astrocytoma, Brain Metastasis
Treatment History		
Biopsy	Enum	Yes, No, Unknown
Boost Therapy	Enum	Yes, No, Unknown
CeTeG Protocol	Enum	Yes, No, Unknown
Chemotherapy	Enum	Yes, No, Unknown
FET-PET	Enum	Yes, No, Unknown
Radiotherapy	Enum	Yes, No, Unknown
Resection	Enum	Yes, No, Unknown
TMZ Therapy	Enum	Yes, No, Unknown
Patient Information		
Age	Range	$\{x \in \mathbb{N}_0 \mid x \leq 120\}$

Here, a single institution with elaborate patient data of around 8000 patients, covering all sorts of diseases (and irrespective of specific attributes), is already considered very extensive.

Nuclear medicine utilizes radioactive substances, so-called tracers, for diagnosis in PET-CT scanning [32]. Taken up by active tumor cells with specific features correlated to, e.g., receptor expression on channel activity or metabolism, the accumulated tracers emitting radiation are detectable by a PET scanner and thus help to identify the tumor's location and activity. Particularly, its application to brain tumors in the context of medical imaging is currently without substitute in clinical practice. Treating such rare diseases also highlights why patient data can be exceptionally scarce.

When conceptualizing these influences into distinct attributes of varying types, we can categorize them as follows.

- *Diagnosis* captures, for example, whether the tumor is active, how it has progressed so far, and other symptoms.
- *Tumor Information* expresses, e.g., tumor type, its position, etc., but also biopsy data, such as specific gene mutations of relevance for the treatment, as well as morphological and metabolic information that can be further disentangled by so-called radiomics data mining of medical images (CT, MRI, or PET).
- *Treatment History* records (past) therapies, e.g., chemotherapy.
- *Patient Information* provides additional insights; most frequently, age and gender are of interest here.

These attributes (summarized in Table 1) provide experts with a data-driven basis for tailoring treatment strategies to individual tumors and patients, e.g., when balancing the toxicity and efficacy of chemotherapy. Thus, the ability to discover "similar" patients across institutions is essential for improving treatment strategies and patient outcomes.

2.2. Obstacles to Overcoming the Data Scarcity

As a prerequisite for assessing prior work, we now define the threat model and associated research challenges.

Threat Model. Our setting features at least three types of entities. In addition to (i) the data consumer and (ii) one or multiple data providers, we also need to account for (iii) snooping third parties, which must not get access to any sensitive patient data. When utilizing a (distributed) data ecosystem for processing and/or storage, (iv) all parties operating the ecosystem must be considered as well. Overall, privacy must be considered for queries, processed data, and returned results. Exposing side-channel information to anyone is acceptable if and only if sensitive patient data is not affected, i.e., it must remain confidential at all times.

Given the setting with healthcare institutions, we assume that authorization is in place, i.e., only authorized data consumers and providers may participate. Accordingly, patients are not expected to interact with any party other than their treating practitioner. In light of these restrictions, we assume a semi-honest attacker model and expect entities not to collude with each other. Participating institutions are bound by legislation, practitioners take the Hippocratic Oath, and researchers are guided by scientific integrity. However, under our threat model, they still aim at maximizing the information gained from participating in PATDISCOVER.

Research Challenges. Within this setting (cf. Figure 1) and under consideration of the threat model, we identify four challenges with close relevance to information security.

Primary Challenge: ►**C1**: SENSIBLE INDEXING. Most importantly, the discovery mechanism must adequately account for the information's sensitivity, i.e., any processing of patient data and disease details may not violate the patients' privacy and right to confidentiality at any point in time. This challenge arises in three dimensions: Data privacy, query privacy, and result privacy, highlighting the privacy requirements of both data consumers and data providers. Hence, a sensible indexing for patient data must address these dimensions. However, leaking certain side-channel information, like the frequency of queried attributes, may even be acceptable in practice (cf. discussed threat model) as long as it does not reveal sensitive patient data.

Applicability-Specific Challenges: In addition to the core aspect of privacy preservation, three additional challenges are crucial for ensuring the practicability and value of any proposed design. ►**C2**: UTILITY. This aspect expresses that patient discovery also comes with usability requirements. On the one hand, practitioners must be able to apply the approach in real-world clinical contexts (ease of use), i.e., they may not be overwhelmed by its complexity. On the other hand, the expressiveness of queries is crucial to support highly-specialized inquiries. Otherwise, the design cannot express the intricacies of rare diseases. ►**C3**: AVAILABILITY. The benefits of any discovery tool further correlate with the availability of indexed patient data. Consequently, queries must be possible at all times, indicating that the design should have a mechanism that deals with data providers being offline/unavailable (temporarily). ►**C4**: PERFORMANCE.

Finally, any secure approach is only practical if it addresses the performance requirements: Data consumers expect results to be returned in a timely manner. Accordingly, the discovery protocol must excel in the following three scalability dimensions: number of patients, involved institutions (data providers), and combinations of queried attributes.

Looking at the bigger picture, compatibility and interoperability with large-scale initiatives, such as the European Health Data Space [33] or EHRs [2], is a noble goal for increasing the likelihood of widespread use. Nonetheless, we consider this aspect to be a technical task (with the need for standardization) rather than a pressing research challenge and thus defer it for the sake of concision.

Takeaway. *The attribute-based discovery of "similar" patients is a challenging endeavor, given the strict confidentiality requirements. Regardless, it promises significant benefits in clinical contexts, especially when treating (or researching) rare diseases like brain tumors.*

3. Background and State of the Art

In response to the identified challenges, we first survey solutions within our design space in Section 3.1. Subsequently, in Section 3.2, we study the feasibility of related work for privacy-preserving patient discovery and point out a lack of practical approaches. Finally, in Section 3.3, we introduce the technical foundation of our design, namely, homomorphic encryption.

3.1. Discussing the Design Space

Directly modifying sensitive data requires careful consideration of both privacy and utility (cf. **C2**). A variety of strategies have been developed to address this challenge, including classical models such as k-anonymity [9], distributed approaches like the personal health train [11], and mechanisms based on differential privacy [10]. The latter has seen significant refinement, with Garfinkel et al. [34], [35] striving for improved real-world deployments. At the same time, protecting against re-identification of individuals remains a central concern [36]. The limited expressions per feature, e.g., for specific tumor types or treatments, seemingly make the setting a good candidate to apply differential privacy on. However, its goal—protecting individuals—is in stark contrast to our setting, which specifically requires and builds upon identifying "similar" patients, i.e., individuals.

Beyond anonymization and differential privacy, a number of alternative approaches have been proposed. Relying on a trusted third party represents a straightforward option, though one that introduces a potential single point of failure. Trusted execution environments provide stronger guarantees and have been successfully applied in medical contexts [37], albeit at the cost of specific hardware requirements that may hinder broader use. Blockchain-based solutions [38]–[40] present another promising direction but face challenges regarding regulatory compliance and alignment with data protection requirements such as the GDPR [41]. Recent work [42] has further pointed out a gap between promise and practice when applying these technologies in real-world applications. Regardless, software-based approaches, such as homomorphic encryption and searchable encryption, emerge as promising building blocks that provide strong confidentiality guarantees without necessarily relying on trusted intermediaries or specialized hardware [43], offering a reasonable middle ground for real-world deployments.

3.2. Related Work

Fundamentally, prior work utilizes well-established concepts to achieve information security: Homomorphic encryption (HE) [4], [23]–[26], [44]–[46], searchable encryption (SE) [20]–[22], secure multi-party computation (SMC) [17], [18], [47], and trusted execution environments (TEEs) [19]. Some [17], [22], [26] of these approaches even have an explicit focus on clinical research contexts. General frameworks [48], [49] for creating cryptographic schemes, e.g., to implement encrypted databases, complement the picture.

Table 2 places these approaches in the context of the raised research challenges (Section 2.2). Indeed, most approaches satisfy the confidentiality requirements (**C1**). Notably, one HE-based approach [26] requires plaintext access to the query. In contrast, ObliDB [19] leaks (intermediate) result sizes, and approaches [20]–[22] that utilize searchable encryption *locally* reveal access patterns, compromising result privacy. Except for ObliDB, many designs [4], [17], [18], [21], [22], [25], [26], [44] trade off the expressiveness of queries (cf. **C2**) with information security, constraining the feasibility for highly-specialized inquiries. Secure multi-party computation-based concepts [17], [18], [47] fail to satisfy the need for availability (**C3**). General-purpose encrypted database systems [19], [23], [24], [45], [46] implement rich capabilities, such as sorting, relational query support, or grouping, which mandate, e.g., scheme switching and thus induce performance penalties and implementation complexity. Finally, the reliance on complex technical building blocks constrains their performance across the board (**C4**).

Even though various approaches have been proposed, they only inadequately fit to the challenges at hand: Prior work either lacks sufficient availability and scalability [4], [17], [18], [47], leaks confidential information [19]–[22], [26], or has inadequate query support [25]. Thus, we are in general need of new approaches that convincingly tackle the aforementioned research challenges (cf. Section 2).

3.3. Preliminaries

As we detail in Section 4.1, our proposed approach, PATDISCOVER, builds on an indexing using encrypted patient data to account for confidentiality needs. Thus, we introduce the relevant background information next.

Homomorphic Encryption (HE). This specialized encryption enables function evaluation on encrypted inputs without the need to decrypt any of the involved ciphertexts in the process. HE schemes differ in supported features, which allows for a granular classification into partially homomorphic (PHE), somewhat homomorphic (SWHE), leveled fully

TABLE 2. PAST (SECURE) APPROACHES HAVE NOTICEABLE ISSUES WITH THE REQUIRED EXPRESSIVENESS IN PATIENT DISCOVERY SETTINGS.

Approach	Building Block	Data Privacy (cf. C1)	Query Privacy (cf. C1)	Result Privacy (cf. C1)	Expressiveness (cf. C2)	Availability (C3)	Perform. (C4)
CryptDB (2011) [20]	HE, SE	●	●	◕	◕	●	◕
Cao et al. (2013) [21]	SE	●	●	◕	◑	●	◕
Yasuda et al. (2013) [25]	HE	●	●	●	◔	●	◕
Sepheri et al. (2015) [18]	SMC	●	●	●	◑	○	◕
Raisaro et al. (2017) [26]	HE	●	○	●	◑	●	◕
Yuan et al. (2017) [22]	SE	●	●	◕	◑	●	◕
Ziegeldorf et al. (2017) [4]	HE	●	●	●	◑	●	◑
Conclave (2019) [47]	SMC	●	●	●	◑	◔	◕
MedCo (2019) [17]	SMC, HE	●	◕	●	◕	◑	◕
ObliDB (2019) [19]	TEE	◕	◕	◕	●	●	●
PEGASUS (2021) [44]	HE	●	●	◕	◕	●	◑
HEDA (2022) [45]	HE	●	●	◕	◕	●	◑
He^{3}DB (2023) [46]	HE	●	●	◕	◕	●	◑
ArcEDB (2024) [23]	HE	●	●	◕	◕	●	◕
Engorgio (2025) [24]	HE	●	●	◕	◕	●	◕
PATDISCOVER *(this work)* (2025)	HE	●	◕	●	◕	●	◕

homomorphic (LFHE), and fully homomorphic (FHE) [50]. While PHE cannot express arbitrary functions, SWHE does not provide guarantees regarding the function evaluation. In contrast, LFHE ensures the correct computation of chained arbitrary operations up to a concrete (multiplicative) depth. FHE removes these limitations at the expense of computational complexity (the concept is called *bootstrapping*) and increasing ciphertext sizes. Hence, if sufficiently expressive, LFHE schemes promise better efficiency in most settings compared to more powerful FHE schemes.

HE Cryptosystems. In the last decades, research has proposed and evolved several HE schemes [51], [52]. In PATDISCOVER, we utilize the BFV [53], [54] and CKKS [55] schemes given their good fit. BFV enables precise calculations using modular arithmetic on integers, although the modulus choice constrains inputs and outputs, with larger moduli resulting in longer evaluation runtimes. In contrast, CKKS supports approximate arithmetics over complex numbers but introduces inaccuracies when chaining operations.

Specialized (HE) Features. Various (general) concepts provide additional amenities when working with HE, such as cross-scheme optimizations and scheme interoperability. Here, we focus on batching, proxy re-encryption, and scheme switching. ►*Batching* is a ciphertext encoding technique where multiple plaintext values are packed into a single ciphertext. This packing enables SIMD [51], which reduces the computational overhead of using HE. ►*Key switching* allows for changing the secret key of a ciphertext [56]. ►*Proxy re-encryption* allows a third party (proxy) to "key-switch" a ciphertext that is destined for one party to an altered ciphertext for another party [57]. The proxy does not even require access to the encrypted plaintext data when performing the key-switching. ►*Scheme switching* is a concept for transforming ciphertexts from one HE scheme to another, allowing for taking advantage of the individual strengths [44]. However, the switching operation is costly.

4. Design

Having established the scenario and corresponding research challenges, we now focus on PATDISCOVER—our new approach for establishing privacy-preserving discovery of patients. After providing the intuition (Section 4.1), we highlight the attribute types PATDISCOVER currently supports before comparing different realizations in Section 4.3. Finally, we discuss the creation of queries (Section 4.4).

4.1. Design Overview and Processing Sequence

PATDISCOVER utilizes HE to account for the sensitivity of handled patient data (cf. **C1**). In this light, our design further builds on proxy re-encryption (cf. Section 3.3) and a distribution of competencies to achieve adequate data, query, and result privacy for practical use. That is, a key manager is responsible for the key management, while data storage, handling, and processing take place at a discovery server, ensuring the availability of patient data (**C3**). In particular, PATDISCOVER supports complex queries that may consist of multiple attributes of different types to feature-rich expressiveness (cf. **C2**). By design, submitting patient data and querying for "similar" patients requires institutions to be authenticated. The exact registration mechanism underlying this authentication is out of scope for this paper given our focus on the discovery. However, authentication should be straightforward to implement in settings with well-known, registered healthcare and research institutions.

Involved Entities. We are dealing with four types: First, we have data-providing institutions to which we refer as *data custodians*. In PATDISCOVER, we do not refer to them as data providers because they only provide pre-processed information, i.e., various attributes per patient that have been extracted from the original patient data. Second, we refer to institutions looking for "similar" patients as *clients*, i.e., they submit the queries. In practice, institutions may take on the roles of both data custodians and clients over time. Finally, the *key manager* and the *discovery server* are responsible for privacy-preservingly discovering relevant results.

General Workflow. As we illustrate in Figure 2, PATDISCOVER consists of two independent workflows for importing and querying information, respectively. On the one hand (Ⓐ–Ⓓ), data custodians provide homomorphically-encrypted, pre-processed information (i.e., attributes per patient) to the discovery server for indexing. On the other hand (①–⑧), clients create and submit queries for evaluation to the discovery server, which eventually returns the relevant "matches". Such matches correspond to pseudonyms of "similar" patients at a specific institution. These details then enable clients to trigger the next step (⑨), i.e., contacting

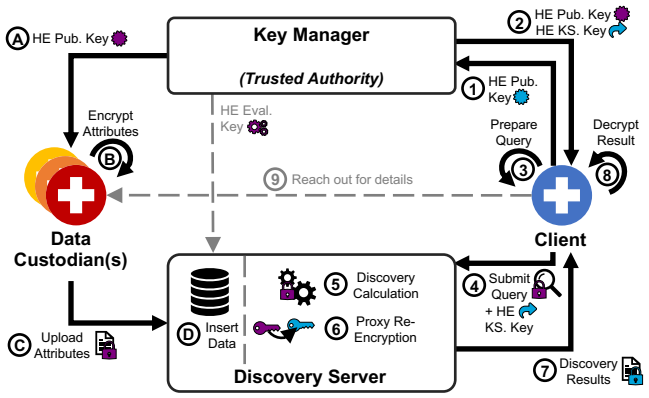

Figure 2. PATDISCOVER utilizes computations on FHE ciphertexts at the discovery server to discover relevant "matches".

identified institutions to request the relevant patient data out of band. We consider Step ⑨ to be independent of the discovery mechanism PATDISCOVER facilitates, and thus, it is out of scope for this paper.

Setup. Apart from the mentioned registration, PATDISCOVER further requires the one-time distribution of certain keys. In particular, the key manager creates HE key material (private, public, and eval keys) and shares the eval key with the discovery server. Likewise, querying institutions, i.e., clients, also require HE key material (private and public) for later use in the protocol. Finally, PATDISCOVER relies on a globally agreed-upon pre-processing to dissect patient data into meaningful, use-case-specific attributes. With FHIR [58], ICD [29], or LOINC [59], various standards are available to provide such pre-processed data in a structured manner. However, given that the identification and interoperability of attributes is highly use-case-specific, we do not cover this largely orthogonal preparatory step in more detail as part of this paper. For our evaluation (Section 5), we later present an exemplary realization for our considered use case of "nuclear medicine" (cf. Section 2.1).

Patient Indexing. In this workflow, the data custodian first retrieves the key manager's HE public key (Step Ⓐ). Subsequently, in Step Ⓑ, the data custodian encrypts the pre-processed attributes of all patients (one HE ciphertext per attribute) using the retrieved key before uploading the HE ciphertexts along with a patient-unique pseudonym to the discovery server (Step Ⓒ). Finally, in Step Ⓓ, the discovery server stores the tuple (institution ID, patient pseudonym, list of ciphertexts). Each data-providing institution conducts this workflow independently. When submitting updates or indexing additional patients for discovery, the data custodian can simply repeat the Steps Ⓑ and Ⓒ to trigger Step Ⓓ.

Patient Discovery. In comparison to the indexing, the privacy-preserving discovery of "similar" patients is more complex. First, in Step ①, the client must share its HE public key with the key manager. In turn, as part of Step ②, with its HE private key, the key server creates a client-specific key-switching key, which later enables proxy re-encryption (cf. Section 3.3) at the discovery server. It forwards this key, as well as its HE public key, to the client. Afterward, the client prepares its query (more in Section 4.4) and encrypts it using the retrieved key (Step ③). In Step ④, the client submits the query and its key-switching key to the discovery server to trigger Step ⑤, i.e., the computations at the discovery server (more in Section 4.2), which depend on the key manager's HE eval key. Afterward (Step ⑥), using the key-switching key, the discovery server transforms the computed result HE ciphertexts before returning them to the client in Step ⑦. Finally, in Step ⑧, the client utilizes its HE private key to decrypt the received (key-switched) HE ciphertexts. As mentioned before, in Step ⑨, the discovered "matches", i.e., the list of "similar" patients (institution ID, patient pseudonym), are to be used by the client. In PATDISCOVER, queries are entirely independent of each other.

TABLE 3. PATDISCOVER SUPPORTS THE FOLLOWING ATTRIBUTE TYPES.

Type	Set	Query Data	Query Result
Boolean	$d \in \{0,1\}$	$q \in \{0,1\}$	$r = \begin{cases} 1 & \text{if } d = q \\ 0 & \text{otherwise} \end{cases}$
Enum	$d \in \mathbb{N}_f$	$q \in \mathbb{N}_f$	$r = \begin{cases} 1 & \text{if } d = q \\ 0 & \text{otherwise} \end{cases}$
Range	$d \in \mathbb{R}_f$	$q = (q_{\text{lb}}, q_{\text{ub}})^T \in \mathbb{R}_f^2$	$r = \begin{cases} 1 & \text{if } q_{\text{lb}} < d < q_{\text{ub}} \\ 0 & \text{otherwise} \end{cases}$
Distance	$d \in \mathbb{R}_f^3$	$q = (x, y, z, q_{\text{ud}})^T \in \mathbb{R}_f^4$ $q_p = (x, y, z)^T \in \mathbb{R}_f^3$	$r = \begin{cases} 1 & \text{if } \text{dist}(d, q_p) < q_{\text{ud}} \\ 0 & \text{otherwise} \end{cases}$

f indicates that the respective set is a finite subset.

4.2. Query Attributes and their Evaluation

As we have discussed in Section 2.1, the discovery of "similar" patients depends on attributes of different types. In accordance with these needs, we incorporate four types in PATDISCOVER (Table 3): Boolean, Enum, Range, and Distance. Collectively, they offer the required expressiveness for real-world queries (cf. **C2**).

PATDISCOVER addresses the confidentiality requirements by relying on HE. Hence, we have to express the attribute types using functions amenable to homomorphic evaluation. HE schemes support only a (varyingly) limited set of operations and differ in computational overhead and precision. We realize the homomorphic "comparisons" for an attribute value d and a query q as follows.

Boolean. With $d, q \in \{0,1\}$, this comparison relies on a simple formula.

$$\text{Boolean}(d, q) = d \cdot q + (1 - d) \cdot (1 - q) \tag{1}$$

Enum. Assuming a finite field $\mathbb{F}_p$ (p must be prime) with at most p distinct elements, we compare Enum attributes as follows.

$$\text{Enum}_{\text{precise}}(d, q) = 1 - (d - q)^{p-1} \mod p \tag{2}$$

Range. To check whether an item lies within defined bounds (lb, ub), we utilize a less-than operator LT from prior work [60].

$$\text{Range}_{\text{precise}}(d, (q_{\text{lb}}, q_{\text{ub}})) = \text{LT}(q_{\text{lb}}, d) \cdot \text{LT}(d, q_{\text{ub}}) \quad (3)$$

Distance. First, an element-wise comparison, i.e., $\forall i \in \{1, 2, 3\} : (d_i - q_i)^2$, is computed on HE ciphertexts before comparing the resulting sum, i.e., $\text{dist}()^2$, to the maximum upper distance ud.

$$\text{Distance}_{\text{precise}}(d, (q_{\text{pos}}, q_{\text{ud}})) = \text{LT}(\text{dist}(d, q_{\text{pos}})^2, q_{\text{ud}}) \quad (4)$$

Now, these computations *precisely* represent the attribute types from Table 3. However, they depend on costly multiplications, which could potentially add significant computational overhead. Thus, we also propose *approximated* attribute types with superior performance (**C4**) for integration in PATDISCOVER (cf. Section 5.3).

We make two adjustments: First, we rely on the idea of approximating the sign function ($\text{sgn}_{\text{approx.}}$) by chaining polynomials until the desired approximation precision is reached [61]. We choose the minimax approximation polynomials by Lee et al. [62] to build a composite polynomial that provides the desired precision and requires minimal multiplication operations. Additionally, we boost the precision of the composite polynomial by chaining a last polynomial f (cf. [61]), doubling the precision if the approximation error is already low [63]. To efficiently evaluate the composite polynomial, we utilize the baby-step/giant-step algorithm [64], which has been proposed before [65]. Second, we conceptualize an approximated less-than operator $\text{LT}_{\text{approx.}}$, which also builds on $\text{sgn}_{\text{approx.}}$.

$$\text{LT}_{\text{approx.}}(a, b) = \text{sgn}^2_{\text{approx.}}(b - a) \cdot \frac{\text{sgn}_{\text{approx.}}(b - a) + 1}{2} \quad (5)$$

These changes result in the following approximate comparisons.

Enum Approx. In contrast to $\text{Enum}_{\text{precise}}$, this realization relies on the approximated sign function for improved performance.

$$\text{Enum}_{\text{approx.}}(d, q) = 1 - \text{sgn}^2_{\text{approx.}}(d - q) \quad (6)$$

Range Approx. For the range comparison, we only replace the "precise" LT operator with an approximated version ($\text{LT}_{\text{approx.}}$).

$$\text{Range}_{\text{approx.}}(d, (q_{\text{lb}}, q_{\text{ub}})) = \text{LT}_{\text{approx.}}(q_{\text{lb}}, d) \cdot \text{LT}_{\text{approx.}}(d, q_{\text{ub}}) \quad (7)$$

Distance Approx. The same holds for the distance comparison.

$$\text{Distance}_{\text{approx.}}(d, (q_{\text{pos}}, q_{\text{ud}})) = \text{LT}_{\text{approx.}}(\text{dist}(d, q_{\text{pos}})^2, q_{\text{ud}}) \quad (8)$$

Together, these precise and approximate homomorphic comparison strategies enable us to trade off performance and precision. Moving on, we theoretically analyze the computational complexity of implementing these comparisons in PATDISCOVER, which we later complement via experimental validation in Section 5.3.

TABLE 4. DIFFERENT COMPUTATIONAL COMPLEXITIES ARE FEASIBLE.

Type	HE Scheme	Multiply Operations	Multiplicative Depth	Pre-Processing
Boolean	BFV	2	1	✗
Enum Precise	BFV	16	16	✗
Range Precise	BFV	551	21	✗
Distance Precise	BFV	278	21	✗
Enum Approx.	CKKS	30	13	✓
Range Approx.	CKKS	97	18	✓
Distance Approx.	CKKS	66	25	✓

Multiplicative operations *only* count ciphertext-ciphertext multiplications, omitting plaintext-ciphertext multiplications.

4.3. Attribute Comparison

Looking at practical, real-world performance (**C4**), we notice that the precise formulas introduce a significant number of multiplicative operations and depth, as Table 4 summarizes. This situation is undesirable when dealing with HE (cf. Section 3.3). Hence, despite requiring a client to pre-process attributes (normalizing them to the interval $[0, 1]$), the lower number of multiplicative operations promises valuable performance improvements. In Section 5.3, we report on the measured performance of the different attribute types.

As we detail in Section B, we also experimented with an alternative precise realization that utilizes scheme-switching (cf. Section 3.3) to implement the attribute types. In particular, we switched ciphertexts from the CKKS [55] to FHEW [66] and vice versa, as demonstrated in prior work [44], to exploit the benefits of the individual HE schemes. This approach thus promises high precision at the expense of no support for batching and the need for (costly) scheme-switching. In FHEW, we relied on the large-precision homomorphic sign evaluation [67]. However, our measurements confirmed the superior performance of our presented approach without scheme-switching. Thus, we disregard this alternative.

4.4. Tree-Based Query Structure and Result(s)

Practical implementations for the discovery of "similar" patients mandate adequate query expressiveness (cf. **C2**). Accordingly, in PATDISCOVER, we introduce a tree-based query structure *per attribute type*, which clients create and which the discovery server processes. In particular, we support combining attributes using three operators, `AND`, `OR`, and `SUM`, which can be mixed as needed, offering diverse composition opportunities to querying institutions. The `SUM` operator simply adds ciphertexts ($a + b$) while the `AND` ($a \cdot b$) and `OR` ($a + b - a \cdot b$) are intended for Boolean comparisons. This way, modeling a rich set of queries is possible. In Section 5.2, we will present a real-world query example.

In response to the confidentiality requirements (cf. **C1**), the discovery server executes these operations on HE ciphertexts. Consequently, the complexity of the query also contributes to the overall multiplicative depths (due to the combination of the tree's height and depth) and count of ciphertext-ciphertext multiplications, impacting performance and precision alike. Hence, the chosen HE scheme must

accommodate these matters, account for the height of the created trees, and be configured accordingly (cf. Section 3.3).

As part of Step ⑧, the discovery server returns key-switched HE ciphertexts with the query results to the client. Each tree that has been part of a query is being treated separately. A query consists of at least as many trees as different attribute types that have been part of the query. Per tree, the discovery server returns one homomorphically-encrypted result per indexed patient, along with a mapping to the institution ID and the patient pseudonym. When using batching (cf. Section 3.3), multiple patients are recorded in a single HE ciphertext, introducing some negligible implementation complexity. After decryption, the client can verify whether the query result *per tree* matches its needs to eventually identify the institutions and pseudonyms of "similar" patients, disregarding the others. This information then serves as input for Step ⑨ (reaching out), exceeding the functionality PATDISCOVER provides.

Takeaway. *Our design,* PATDISCOVER*, focuses on the privacy-preserving discovery of "similar" patients across institutions while taking into account several critical challenges (cf. Section 2.2). It supports all attribute types that are relevant in this context (realized either precisely or as approximated counterparts) and even allows for expressing complex queries. The inherent use of HE is key to ensure the confidentiality needs when handling sensitive patient data.*

5. Realization and Evaluation

We now focus on assessing the performance and scalability (**C4**), precision, and security (cf. **C1**) of PATDISCOVER in detail. First, in Section 5.1, we introduce our open-sourced PATDISCOVER implementation [27] before reporting on our experimental setup, along with the sourced use case data in Section 5.2. Afterward, in Section 5.3, we present our evaluation results. Our analysis covers multiple perspectives: Scalability, runtime, data transmissions, memory needs, and loss of precision. After discussing these measurements, we focus on PATDISCOVER's security (guarantees) in Section 5.4.

5.1. Implementation

We implemented a prototype [27] of PATDISCOVER in C++. We utilize the OpenFHE library [68] to internally build on the BFV [53], [54] and CKKS [55] schemes (cf. Section 3.3). We further rely on the Intel HEXL [69] backend to accelerate the HE computations. Both HE schemes support batching, enabling SIMD-based performance improvements, and allow parallel processing of independent attributes. We utilize the NTL [70] library for implementing the LT operator, source Lattigo [63] for pre-computing minimax approximation polynomials, and avoid costly bootstrapping operations by using FHE schemes as leveled ones (LFHE).

Due to its limited resource needs, we configured SQLite [71] as the database backend. We implemented all networking functionality using two common libraries: protobuf [72] and gRPC [73]. All network communication is authenticated using self-signed X.509 certificates and is further protected using TLS 1.3 during transit.

5.2. Experimental Setup and Use Case Data

We configured our evaluation environment as follows.

Experimental Setup. We configure the security of our HE schemes according to the compiled HE standard [74] to achieve an equivalent of 128 bit security. We leave the default OpenFHE parameters for any remaining settings, promising good performance [75]. This way, we utilize the supported batching (cf. Section 3.3) and pack 32 768 and 65 536 plaintext values for BFV and CKKS, respectively, into a single HE ciphertext. We deployed all entities on a single server (2x Intel Xeon Silver 4116 and 196 GB RAM). Hence, all communication was local and not artificially constrained. We report means and 99 % confidence intervals over 30 runs.

Dataset: Patient Letters from Nuclear Medicine. To identify relevant attributes (cf. Table 3), a medical practitioner sourced ten real-world patient letters. In total, we ended up with five Boolean, ten Enum (at most four values), one Distance (tumor position), and Range (age) attribute(s) (cf. Section 3.3). Using this information, we created a generator for synthetic patient data that maintains semantic correctness and samples data per attribute uniformly at random. Given the data-independent processing of HE calculations, we simulate a large-scale application of up to 500 k patients while still accounting for the sensitivity of real-world patient data.

Representative Query. Together with a domain expert from nuclear medicine, we crafted a representative query (Figure 3) that queries for a specific tumor, in a specific position, in a young patient (between the ages of 20 and 40), who optionally has been treated with chemotherapy. It consists of four separate trees (one per attribute) with heights one, two, zero, and zero, respectively, and contains every attribute type (cf. Table 3), totaling seven attributes: (1) [Boolean] checks whether two attributes of tumor information hold (using the `AND` operator), (2) [Enum] first compares whether the tumor is of a specific type (`OR` operator) before applying the `SUM` operator to include whether the patient has received chemotherapy (3) [Range] simply conducts an age check (without any operator), and (4) [Distance] captures the tumor

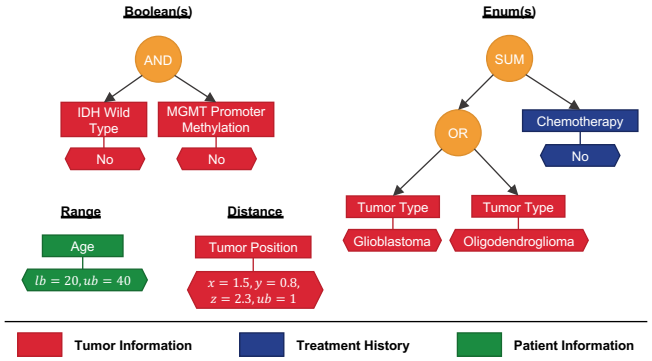

Figure 3. A visual illustration of our representative evaluation query that utilizes all attribute types while also combining multiple attributes across categories (`SUM` operator).

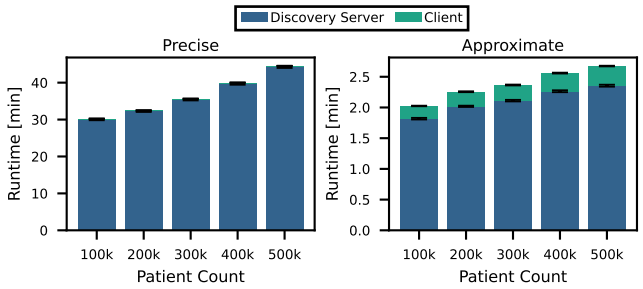

Figure 4. The computations scale linearly with the number of indexed patients. The runtime of the approximated attributes is significantly shorter compared to the precise ones, at the expense of (acceptable) decryption overhead at the client.

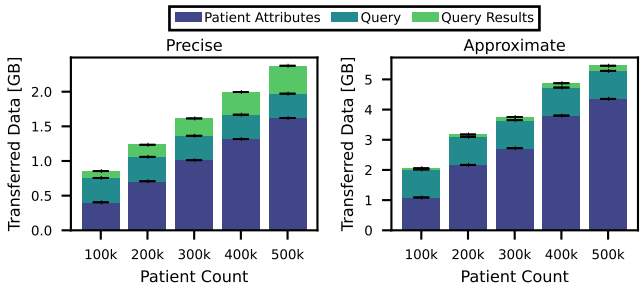

Figure 5. Approximated attributes add significant overhead to the transferred data of the precise attributes. The indexing of patient attributes results in large data transmissions.

position (again without any combining operator). Given that the Distance and Range attribute types occur only once (cf. Table 1), we do not perform any aggregation; instead, we return the results (1 or 0; cf. Table 3) per attribute. In contrast, our representative query combines multiple Enum and Boolean attributes, respectively, to mimic real-world use.

5.3. Performance Evaluation

In the following, we analyze the performance of PATDISCOVER in detail. After an initial assessment of the overall performance in realistic real-world settings, we look at the attribute types individually.

5.3.1. End-to-End Performance. As we illustrate in Figure 4, the runtime scales linearly with the number of indexed patients. The corresponding slopes are 0.000036 (precise) and 0.000002 (approx.). The respective R^2 scores are on par with 0.99, being nearly perfect. Overall, the approximated attribute types outperform the precise ones. However, relatively speaking, they have a higher client-side processing. At first glance, this processing takes up a large fraction of up to 11 % of the total processing. However, in absolute numbers, this decryption overhead is negligible when comparing both approaches: 6.3 s (precise) and 12 s (approx.) for 100 k patients.

►*Runtime.* Looking at the total runtime, we identify two crucial observations. First, even when looking for "similar" patients in a large set of 500 k indexed patients, despite the HE-induced overhead, the query concludes in less than 45 min (3 min for the approximated variant). Approximated variants are faster than their precise counterparts (except for Enums), as we further detail in Section 5.3.4. We consider this performance to be well-suited, especially when dealing with rare diseases and the discovery of respective patients, where new patients at a single institution surface infrequently and with a certain delay. Second, the majority of the computations are with the discovery server. The processing by the key server is negligible. Consequently, scaling out or scaling up the discovery server are convincing strategies to address potential performance bottlenecks.

►*Storage and Networking.* Similar observations hold for the storage and networking performance (Figure 5). Different from the runtime, the approximated variant introduces some overhead when compared to the precise one. The approximated attributes trade off runtime benefits for additional network load, with the size required for the HE eval keys being notably larger for the approximate variants (precise: 554 MB and approx.: 914 MB). Especially in constrained networks, this difference is impactful. Fortunately, this observation highlights that PATDISCOVER can be configured according to use-case-specific requirements.

Locally, data transmitted for the query (including its result) must only be stored temporarily. Hence, only the indexed patient attributes add a permanent storage burden on the discovery server. We consider a measured storage of 0.41 GB (precise) and 1.1 GB (approx.) per 100 k patients as well acceptable for real-world use.

►*Memory.* Likewise, the maximum memory consumption for our largest evaluation setting with 500 k indexed patients is 52 GB and 18 GB for the precise and approximated variant, respectively. These numbers can also be reduced when executing fewer HE computations in parallel, increasing the runtime. Thus, PATDISCOVER's memory needs comply easily with off-the-shelf server hardware.

Result: PATDISCOVER*'s performance (runtime, storage, networking, and memory requirements) is satisfactory for real-world settings.*

5.3.2. Scalability Assessment. Complementing our measurements, we now discuss the three scalability dimensions of interest (cf. **C4**).

►*Number of Patients.* Our performance measurements already highlight the linear scaling of PATDISCOVER concerning the number of indexed patients. Since our implementation processes large numbers of patients in batches (not to be confused with batching in HE schemes) and these evaluations have been using our experimental setup to capacity, we are confident that even larger settings would also confirm the reported, near-perfect linear trends.

►*Involved Institutions.* By design, the number of involved institutions (data providers) does neither influence the processing of attributes nor queries at the discovery server (Steps Ⓓ and ⑤–⑦) because all uploaded patient attributes are encrypted with the same public key (✹). The introduction of additional institutions thus mainly impacts the distribution phase, where the key server distributes its public key (✹) during the initial setup (Step Ⓐ). This process imposes only a minimal overhead on the patient indexing. The situation is

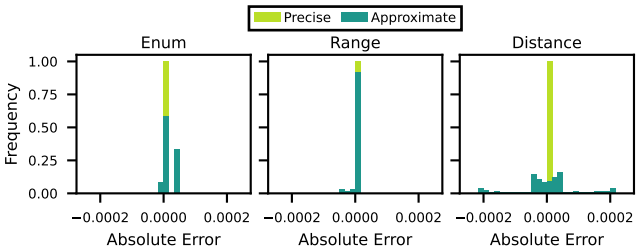

Figure 6. The precision of the conducted HE computations is good across attribute types, with absolute errors being small.

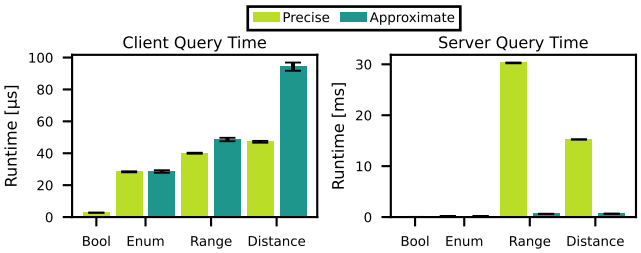

Figure 7. While precise attributes burden primarily the server, approximated attributes put processing on the client as well.

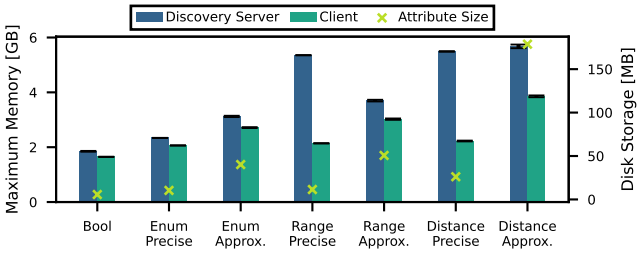

Figure 8. Memory needs and storage usage are not excessive.

similar for an increasing number of querying clients. While the operation of the discovery server is strictly identical for each query (Steps ④–⑦), each "new" client has to interact with the key manager at least once (Steps ① and ②), resulting in minor processing at the key server. If necessary, the key server scales horizontally. We thus conclude that the number of involved institutions has no practical relevance for PATDISCOVER's scalability properties.

►*Combinations of Queried Attributes.* The query complexity is another relevant scalability dimension. First, after varying the number of attributes in a single query, we identified a linear correlation with the runtime, which is highly desirable for practical scalability in real-world settings. Certainly, the attribute types come with different individual processing times, as we carry out in more detail (Section 5.3.4). Second, we also studied the influence of the query height (cf. Section 4.4). Moreover, when increasing the height of a perfect binary tree, we confirmed an exponential growth in the measured runtime, which follows from the exponential growth of the number of contained attributes in such an expanded query.

Result: While the number of involved data providers barely impacts PATDISCOVER*'s scalability properties by design, the number of indexed patients correlates linearly with the runtime and storage requirements. Likewise, during patient discovery, the runtime of a query scales linearly with the number of attributes. Both dimensions allow for parallel processing. In contrast and as expected, the query height has an exponential influence on the runtime.*

5.3.3. Validity of Results. Since we are dealing with an HE scheme, which might lose precision (i.e., CKKS; BFV is exact), we assess this dimension by comparing the ciphertext calculation to its plaintext counterpart. As evaluation, we compute >100 000 runs per attribute type, as we summarize in Figure 6. The precise building blocks that build on BFV do not show any issues. In contrast, the approximated variants reveal only minor deviations. However, even in the worst case, the absolute error remains below 0.00022 per single attribute computation. Accordingly, we did not encounter any situation where the HE-based "matching" results were skewed, i.e., we do not observe any false negatives or false positives. Hence, the approximation and approximate arithmetics the CKKS scheme introduces (cf. Section 3.3) do not skew or impact PATDISCOVER's results. Thus, we conclude that, with our HE configuration (schemes), all attribute types provide sufficient precision for practical, real-world use in healthcare.

Result: The performance of the attribute types and their realization (precise vs. approximated) differs greatly. The most commonly-used attribute types (Boolean and Enum) perform best, while range and distance attributes introduce longer runtimes and require more memory.

5.3.4. Attribute Type Comparison. After focusing on the overall performance and having established a satisfactory performance for practical, real-world application in healthcare, we now investigate the differences between the attribute types and their realizations (precise vs. approximated). In Figure 7, we show the processing time per attribute type for the client and the discovery server. The pre-processing for the approximated variants at the client is negligible, with less than 2 ms for a single attribute. In contrast, the discovery server's processing times merit attention since, despite batching improvements, execution is required for each indexed patient. Overall, approximated variants (except Enum) compute faster than precise ones, with Range and Distance attributes taking significantly longer to process than Boolean and Enum types.

A brief look at the consumed memory and storage across the different attribute types confirms the moderate requirements, as Figure 8 shows. The discovery server's memory usage is higher than the client's usage. Again, the rather uncommon attribute types (Range and Distance) exhibit larger memory needs. The storage needs per attribute type are mostly on par; only the approximated distance type introduces a six-fold increase. These numbers confirm our conclusions on end-to-end performance (cf. Section 5.3).

5.4. Security Discussion

PATDISCOVER operates on the foundation of secure cryptographic primitives, their correct configuration, and TLS-authenticated communication. We further assume that

no entity has been compromised by a third party, i.e., entities do not act maliciously beyond the capabilities defined in the semi-honest attacker model (cf. Section 2). Having this situation in mind, we now assess how PATDISCOVER fulfills the privacy-preservation aspects **C1** captures.

We have designed PATDISCOVER to reliably protect sensitive patient data in settings with semi-honest adversaries. Given this threat model, it addresses the outlined privacy requirements, provided that no collusion takes place. In medical settings, the required non-collusion assumption holds as (a) institutions, practitioners, and researchers are well-known and authenticated at all times, and (b) their reputation and law-abidance are crucial to their operations.

Privacy Preservation. Given that all entities are authenticated, a curious key manager or discovery server could analyze meta information like access patterns. As a countermeasure, data custodians and clients could, in theory, index dummy data or submit dummy queries, respectively. The discovery server is further aware of the mapping between uploaded ciphertexts and their origin, i.e., the respective data custodian. However, we consider corresponding insights to be marginal as they do not compromise any patient data.

►*Data Privacy.* In line with its intended purpose, clients can query for "similar" patients across institutions by comparing relevant attributes. By design, all sensitive patient data is recorded in abstracted attributes, which are only processed and queried in an encrypted form. Thus, PATDISCOVER ensures data privacy. For stronger, long-term privacy guarantees, the discovery server could periodically apply proxy re-encryption to update the underlying encryption key, mitigating the impact of using a persistent key.

►*Query Privacy.* Just like searchable encryption [76], PATDISCOVER's underlying design is also prone to revealing statistical information: While the precise configuration of attributes in the query is homomorphically encrypted, the queried attributes are visible for the discovery server, opening a side channel. Hence, it may record and analyze the frequency of queried attributes, attribute combinations, or query composition (i.e., the query trees). However, the exact values are hidden from the server. Nonetheless, the discovery server could track which attributes are queried by a specific client, i.e., an institution, identifying its main focus area. In terms of more complex queries that make use of operators, the discovery server can further observe frequent combinations of operators and attribute types, indicating potential (inter)dependencies. Consequently, given the exposed side-channel information, PATDISCOVER does not ensure "query privacy" to the fullest extent. However, this minor limitation is compliant with the discussed privacy requirements of our setting (cf. **C1**) because the exposed side-channel information does not reveal detailed insights into the query's origin and goals due to the low number of alternatives, even over time.

►*Result Privacy.* Since all computations depend on HE, they conceal control flows, while results remain encrypted and indistinguishable, accessible only via a client-specific authenticated key. With clients' reputations at stake, there is no incentive to share keys. Thus, PATDISCOVER adequately protects sensitive results.

Data Injection and Extraction. As all indexed patient attributes and queries are homomorphically encrypted, institutions could potentially submit irrelevant or misleading data. However, false-data injection by data custodians would be noticed during Step ⑨ when clients initiate the out-of-band information exchange. In this case, the data custodian may learn information about the query, i.e., queried attributes and their composition (cf. Section 4.4), slightly impairing the query privacy. Nonetheless, we consider the likelihood (and profits) of such an attack on PATDISCOVER to be low.

In contrast, a client submitting garbage (false) queries has little appeal and little impact as well. First, clients most likely have to pay for queries, thus misuse is a costly endeavor. Second, returned results do not reveal any patient information or details beyond the attribute "matches". Hence, a client could only learn a rough distribution and mapping of patients and specific attributes across institutions. Thus, the respective client's misbehavior adds little value. However, excessively submitting clever-crafted queries will eventually reveal, potentially more insightful, frequency statistics about specific attributes of indexed patients. If required by the data custodians, the discovery server could implement a rate-limiting mechanism to mitigate such frequency attacks.

Entity Misbehavior. Within the semi-honest attacker model, entities will stick to protocols. However, given the sensitive data involved, we briefly discuss the implications of potential misbehavior and show that no direct privacy breaches are to be expected. In addition to these institution-specific attack vectors, the key manager and discovery server could misbehave (apart from colluding; cf. Section 2.2). However, such behavior would not reveal any insights for them. On the one hand, misbehavior by the key manager would either break computations at the discovery server or interfere with the client-side decryption of the results. Apart from adding no incentive for the key manager, such actions are detectable. Entities would thus experience a loss of reputation. On the other hand, incorrect computations by the discovery server are most likely noticeable as well. While returning false positives is trivially detectable in Step ⑨, false negatives are not. However, if local patient data is returned with inaccuracies or when known information about indexed patients changes over time while revealing semantic discrepancies, clients will also notice such misbehavior. Given that the discovery server cannot learn anything from these manipulations, we consider the attack likelihood as very low. Lastly, the construction of our approximated attribute types exposes an attack vector, which could, in theory, enable the retrieval of sensitive patient data, as we elaborate on in Section C. Fortunately, precision losses in HE schemes render practical attacks unlikely.

Server Operators. For the key manager, we envision a public health association—funded through membership fees—running this service. Its operation barely introduces any computational load (limited to the generation of a key-switching key per client; Step ②) or storage requirements. Only its HE key material (for each attribute type) must be stored permanently. The cumulative sum of both types of keys yields 0.59 GB (precise) and 1.2 GB (approx.).

The operation of the discovery server could be funded through fees for querying or dedicated subscription models. By design, apart from observing some minor query insights (see above), it cannot learn anything due to not having access to any sensitive patient information (it only processes HE ciphertexts). Hence, any third party could take over the computations. However, in a setting where collusion is a reasonable threat (contradicting our threat model), a more reputable third party should be considered, as we discuss next. One example could be a government-operated deployment.

Implications of Collusion. In settings with colluding entities (contradicting our healthcare scenario), some limitations apply. ►*Discovery Server and Key Manager.* This collusion would enable decryption of all indexed attributes—an inherent limitation of distributing competencies. Two independent, reputable operators might mitigate this risk, although PATDISCOVER cannot prevent it. Still, detailed patient data is only locally available at the data custodians. ►*Discovery Server and Client.* They could also decrypt arbitrary data and, seemingly legitimately, trigger Step ⑨. Moving proxy re-encryption (Step ⑥) to the key manager would mitigate this threat at the cost of overhead. ►*Other Collusions.* Collusion between the discovery server and data custodians, or the key manager and any institution, poses minimal threats as they lack direct access to encrypted data.

This short assessment explains why PATDISCOVER is not suitable for settings with stronger adversaries that are absent in our considered scenario. As we also indicate in Table 2, in light of its features (expressiveness, availability, and performance), our design offers competitive privacy preservation compared to related work.

Takeaway. *Our assessment of* PATDISCOVER *demonstrates a privacy-preserving approach with promising performance and scalability (**C4**). Its resource requirements match what is reasonably available at the involved institutions. Our security discussion confirms our design choices. Participating parties can only deduce minor (insensitive) details, indicating fulfillment of the targeted privacy preservation (**C1**). Thus, we conclude that* PATDISCOVER *is readily available to privacy-preservingly discover "similar" patients across institutions.*

6. Discussion and Future Work

After focusing on the technical dimension, we assess the bigger picture, i.e., PATDISCOVER's contribution to the targeted setting of healthcare treatment and research.

Impact. Most importantly, our novel privacy-preserving design for cross-institutional patient discovery opens the door for healthcare improvements. Current data sharing [77], [78] is limited by privacy concerns [30] and difficulties implementing solutions compliant with data protection laws [11], [17]. Our work addresses these shortcomings by enabling queries on pre-processed patient data without storing unprotected sensitive information globally, elegantly decoupling patient discovery from subsequent sharing of (rich and unfiltered) data. Revised consent forms [79] could immediately establish the corresponding legal foundation for such a practice. By focusing on patient attributes extracted from comprehensive clinical letters rather than just EHRs or billing reports, PATDISCOVER might even reduce biases inherent in ICD codes, which often fail to capture nuances needed for individualized treatment [29], [80], [81]. Especially when dealing with a disease that is grouped in a coarse ICD code, this situation is highly problematic.

While this paper emphasized application in rare diseases, our prototype's performance indicates scalability to broader settings, such as patient recruitment for drug trials. PATDISCOVER could also contribute to establishing more diverse clinical trial cohorts without compromising privacy, introducing significant benefits. Nonetheless, its application is not restricted to the domain of healthcare.

Research Challenges. Looking at the outlined challenges (cf. Section 2), PATDISCOVER complies with the majority. It provides a sensible indexing approach (**C1**) without sacrificing any of the other aspects. Arguably in the setting of rare diseases, *query privacy* is only secondary to patients who urgently require treatment. They commonly favor the prospect of treatment over privacy preservation. Hence, real-world deployments can even tolerate minor privacy leaks in such contexts. Moreover, subtleties in the expressiveness (cf. **C2**) of queries require evaluation in field studies. Given that FHE can compute arbitrary functions, future enhancements might be able to address this aspect, i.e., it is not a limitation of PATDISCOVER's design. Lastly, computational performance (**C4**) is eventually a use case- and deployment-specific matter. However, since approaches that do not account for the confidentiality requirements and sensitivity of patient data cannot be put to real-world use, we consider this challenge as secondary. Future developments in the area, like ASIC-based hardware acceleration [82], further promise to invalidate this building block-specific aspect.

Optimizing for Performance or Network Traffic. Based on our evaluation (cf. Figures 4 and 5), precise and approximated attribute types yield identical results despite minor precision differences (Fig. 6). Consequently, operators can thus optimize for performance (approximated types) or reduced data transfer (precise types), with the optimal choice depending on the environment.

Counterintuitive Performance. Our evaluation also highlights that the approximated Enum attribute type does not add any benefits over its precise counterpart. At comparable runtime, it merely introduces larger ciphertexts, additional memory requirements, and less precision (cf. Section 5.3). Hence, interestingly, and different from the other attribute types, the precise variant is generally to be preferred when utilizing the Enum attribute type.

Limitations. Despite its outlined benefits, we see two main limitations. First, PATDISCOVER currently cannot merge results from different attribute types at the discovery server. Clients are thus required to carefully craft queries and handle post-processing. Second, it lacks data quality guarantees, allowing potential indexing of arbitrary information by malicious actors, wasting meaningful resources until misbehavior becomes apparent in Step ⑨—though this risk is minimal given our setting with reputable institutions.

For applications beyond healthcare, the query privacy

side channels we identified (cf. Section 5.4) may become an issue in contexts with differing privacy requirements. We consider addressing these limitations as orthogonal research, since our targeted healthcare setting does not require them.

Future Work. Diverse research could build upon our work. First, *design improvements* could include merging different attribute types, pre-filtering results before returning them to clients, and adding homomorphic signatures [83] or verifiable computations [84] for data integrity. Second, *(minor) implementation-specific improvements* might explore adaptations like optimizing rescaling [61] for better level consumption, though they would likely yield only modest gains. Third, extending support for *other attribute types* beyond numerical values, such as image data [85], would enhance functionality. Fourth, *real-world exploration* by deploying PATDISCOVER across institutions, including an evaluation of downstream processes while also considering data-sharing policies, audit trails, and consent frameworks, would enable researchers to study clinical impact, usability, and suitable business models. Fifth, legal scholars should *assess regulatory compliance*, i.e., check whether PATDISCOVER conforms to cross-institutional collaboration regulations as well as general legislation and data protection laws. Finally, exploring *applications beyond healthcare* could leverage our FHE-based approach in other privacy-sensitive domains.

Takeaway. PATDISCOVER *has the potential to unlock significant benefits and bring amenities to the healthcare domain. It promises to broadly improve patient treatment and research activities alike by enhancing patient discovery*

7. Conclusion

The global discovery of patients and their data, especially across institutions, is particularly challenged by privacy requirements, driven both by expectations and legislation. Despite the expected potential for (immediate) patient treatment and research, in today's medical landscape, institutions cannot simply discover "similar" patients, primarily due to a lack of privacy-preserving approaches. Promising designs from related work, unfortunately, feature shortcomings in terms of query expressiveness, performance, and/or privacy preservation. Our novel design, PATDISCOVER, tackles this research gap and enables attribute-based cross-institutional patient discovery in a privacy-preserving manner. Since we strive for real-world practicability, we opted for simplicity in our design. That is, we focused on satisfying the lower limit privacy requirements for the setting at hand and considered differing needs to be orthogonal work. Building on FHE-based operators, we are able to efficiently support various attribute types, which may also be combined to express complex queries for practical, real-world deployments. Our extensive evaluation builds on real-world patient data from nuclear medicine and thus mimics a distributed setting with multiple institutions and thousands of patients. Its results highlight that PATDISCOVER is indeed practical, even beyond the use case of discovering patients with a "similar" rare disease. We look forward to the impact PATDISCOVER may have on healthcare in the long run.

Acknowledgments

We are grateful to Ina Berenice Fink and the anonymous reviewers of this paper for their thorough feedback, which allowed us to further improve our paper. This work has received funding from the Klaus Tschira Boost Fund, a joint initiative of GSO – Guidance, Skills & Opportunities e.V. and Klaus Tschira Stiftung.

References

[1] B. Djulbegovic and G. H. Guyatt, "Progress in evidence-based medicine: a quarter century on," *The Lancet*, vol. 390, no. 10092, pp. 415–423, 2017.

[2] R. S. Evans, "Electronic Health Records: Then, Now, and in the Future," *Yearbook of Medical Informatics*, vol. 25, no. S 01, pp. S48–S61, 2016.

[3] P. Anantharaman, R. b. Shapiro, V. Varadharaju, and M. E. Locasto, "A Study of Interoperability in Electronic Health Record Software," in *Proceedings of the 2024 Workshop on Cybersecurity in Healthcare (HealthSec '24)*. ACM, 2023, pp. 53–60.

[4] J. H. Ziegeldorf, J. Pennekamp, D. Hellmanns, F. Schwinger, I. Kunze *et al.*, "BLOOM: BLoom filter based Oblivious Outsourced Matchings," *BMC Medical Genomics*, vol. 10 (Suppl 2), 2017.

[5] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, "Big healthcare data: preserving security and privacy," *Journal of Big Data*, vol. 5, no. 1, 2018.

[6] Z. Rahmani, N. Shahini, N. Gat, Z. Yun, Y. Jiang *et al.*, "Privacy-Preserving Collaborative Genomic Research: A Real-Life Deployment and Vision," in *Proceedings of the 2024 Workshop on Cybersecurity in Healthcare (HealthSec '24)*. ACM, 2023, pp. 85–91.

[7] P. Hummel, M. Braun, M. Tretter, and P. Dabrock, "Data sovereignty: A review," *Big Data & Society*, vol. 8, no. 1, p. 2053951720982012, 2021.

[8] S. Wiertz and J. Boldt, "Evaluating models of consent in changing health research environments," *Medicine, Health Care and Philosophy*, vol. 25, no. 2, pp. 269–280, 2022.

[9] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[10] S. L. Garfinkel, *Differential Privacy*, 1st ed. MIT Press, 2025.

[11] O. Beyan, A. Choudhury, J. Van Soest, O. Kohlbacher, L. Zimmermann *et al.*, "Distributed Analytics on Sensitive Medical Data: The Personal Health Train," *Data Intelligence*, vol. 2, no. 1-2, pp. 96–107, 2020.

[12] G. Adam, L. Rampášek, Z. Safikhani, P. Smirnov, B. Haibe-Kains, and A. Goldenberg, "Machine learning approaches to drug response prediction: challenges and recent progress," *NPJ Precision Oncology*, vol. 4, no. 1, p. 19, 2020.

[13] G. Asharov, S. Halevi, Y. Lindell, and T. Rabin, "Privacy-preserving search of similar patients in genomic data," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 4, pp. 104–124, 2018.

[14] R. A. Hughes, K. Tilling, and D. A. Lawlor, "Combining Longitudinal Data From Different Cohorts to Examine the Life-Course Trajectory," *American Journal of Epidemiology*, vol. 190, no. 12, pp. 2680–2689, 2021.

[15] C. Zahren, S. Harvey, L. Weekes, C. Bradshaw, R. Butala *et al.*, "Clinical trials site recruitment optimisation: Guidance from Clinical Trials: Impact and Quality," *Clinical Trials*, vol. 18, no. 5, pp. 594–605, 2021.

[16] K. A. McNeill, "Epidemiology of Brain Tumors," *Neurologic Clinics*, vol. 34, no. 4, pp. 981–998, 2016.

[17] J. L. Raisaro, J. R. Troncoso-Pastoriza, M. Misbach, E. S. Gomes de Sá, J. André *et al.*, "MedCo: Enabling Secure and Privacy-Preserving Exploration of Distributed Clinical and Genomic Data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 4, pp. 1328–1341, 2019.

[18] M. Sepehri, S. Cimato, and E. Damiani, "Privacy-preserving query processing by multi-party computation," *The Computer Journal*, vol. 58, no. 10, pp. 2195–2212, 2015.

[19] S. Eskandarian and M. Zaharia, "ObliDB: Oblivious Query Processing for Secure Databases," *Proceedings of the VLDB Endowment*, vol. 13, no. 2, pp. 169–183, 2019.

[20] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting Confidentiality with Encrypted Query Processing," in *Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP '11)*. ACM, 2011, pp. 85–100.

[21] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2013.

[22] J. Yuan, B. Malin, F. Modave, Y. Guo, W. R. Hogan *et al.*, "Towards a privacy preserving cohort discovery framework for clinical research networks," *Journal of Biomedical Informatics*, vol. 66, pp. 42–51, 2017.

[23] Z. Zhang, S. Bian, Z. Zhao, R. Mao, H. Zhou *et al.*, "ArcEDB: An Arbitrary-Precision Encrypted Database via (Amortized) Modular Homomorphic Encryption," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security (CCS '24)*. ACM, 2024, pp. 4613–4627.

[24] S. Bian, H. Pan, J. Hu, Z. Zhang, Y. Fu *et al.*, "Engorgio: An Arbitrary-Precision Unbounded-Size Hybrid Encrypted Database via Quantized Fully Homomorphic Encryption," in *Proceedings of the 34th USENIX Security Symposium (SEC '25)*. USENIX Association, 2025, pp. 8441–8460.

[25] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Secure Pattern Matching using Somewhat Homomorphic Encryption," in *Proceedings of the 5th ACM Workshop on Cloud Computing Security Workshop (CCSW '13)*. ACM, 2013, pp. 65–76.

[26] J. L. Raisaro, J. G. Klann, K. B. Wagholikar, H. Estiri, J.-P. Hubaux, and S. N. Murphy, "Feasibility of Homomorphic Encryption for Sharing I2B2 Aggregate-Level Data in the Cloud," *AMIA Summits on Translational Science Proceedings*, vol. 2017, no. 1, pp. 176–185, 2017.

[27] J. Pennekamp, J. Lohmöller, N. Pressel, S. Geisler, F. M. Mottaghy, and K. Wehrle, "PatDiscover: Privacy-Preserving Discoverability of Patients," https://github.com/COMSYS/PatDiscover, 2025.

[28] S. Kölker, F. Gleich, U. Mütze, and T. Opladen, "Rare Disease Registries Are Key to Evidence-Based Personalized Medicine: Highlighting the European Experience," *Frontiers in Endocrinology*, vol. 13, 2022.

[29] K. J. O'Malley, K. F. Cook, M. D. Price, K. R. Wildes, J. F. Hurdle, and C. M. Ashton, "Measuring Diagnoses: ICD Code Accuracy," *Health Services Research*, vol. 40, no. 5p2, pp. 1620–1639, 2005.

[30] W. N. Price and I. G. Cohen, "Privacy in the age of medical big data," *Nature Medicine*, vol. 25, no. 1, pp. 37–43, 2019.

[31] S. Geisler, C. Cappiello, I. Celino, D. Chaves Fraga, A. Dimou *et al.*, "From Genesis to Maturity: Managing Knowledge Graph Ecosystems Through Life Cycles," *Proceedings of the VLDB Endowment*, vol. 18, no. 5, pp. 1390–1397, 2025.

[32] V. Kapoor, B. M. McCook, and F. S. Torok, "An Introduction to PET-CT Imaging," *Radiographics*, vol. 24, no. 2, pp. 523–543, 2004.

[33] R. Raab, A. Küderle, A. Zakreuskaya, A. D. Stern, J. Klucken *et al.*, "Federated electronic health records for the European Health Data Space," *The Lancet Digital Health*, vol. 5, no. 11, pp. E840–E847, 2023.

[34] S. L. Garfinkel, J. M. Abowd, and S. Powazek, "Issues Encountered Deploying Differential Privacy," in *Proceedings of the 2018 Workshop on Privacy in the Electronic Society (WPES '18)*. ACM, 2018, pp. 133–137.

[35] S. L. Garfinkel and P. Leclerc, "Randomness Concerns when Deploying Differential Privacy," in *Proceedings of the 19th Workshop on Privacy in the Electronic Society (WPES '2020)*. ACM, 2020, pp. 73–86.

[36] S. L. Garfinkel, "De-Identification of Personal Information," NISTIR 8053, 2015.

[37] J. Lohmöller, R. Matzutt, J. Loos, E. Vlad, J. Pennekamp, and K. Wehrle, "Complementing Organizational Security in Data Ecosystems with Technical Guarantees," in *Proceedings of the 1st Conference on Building a Secure and Empowered Cyberspace (BuildSEC '24)*. IEEE, 2024, pp. 49–56.

[38] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "MedBlock: Efficient and Secure Medical Data Sharing Via Blockchain," *Journal of Medical Systems*, vol. 42, no. 8, 2018.

[39] V. Ramani, T. Kumar, A. Bracken, M. Liyanage, and M. Ylianttila, "Secure and Efficient Data Accessibility in Blockchain Based Healthcare Systems," in *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM '18)*. IEEE, 2018, pp. 206–212.

[40] J. Castillo and Q. Chen, "MediLink: A Secure Blockchain Framework for Multi-Institutional Healthcare," in *Proceedings of the 2024 Workshop on Cybersecurity in Healthcare (HealthSec '24)*. ACM, 2023, pp. 61–68.

[41] A. Hasselgren, P. K. Wan, M. Horn, K. Kralevska, D. Gligoroski, and A. Faxvaag, "GDPR Compliance for Blockchain Applications in Healthcare," arXiv:2009.12913, 2020.

[42] J. Lohmöller, H. Jeon, J. Hentschel, K. Wehrle, and J. Pennekamp, "Between Promise and Practice: Challenges and Misperceptions of Applying Privacy Enhancing Technologies in Business Contexts," in *Proceedings of the 59th Hawaii International Conference on System Sciences (HICSS '26)*. University of Hawaii at Manoa, 2026.

[43] S. L. Nita and M. I. Mihailescu, *Advances to Homomorphic and Searchable Encryption*, 1st ed. Springer, 2023.

[44] W.-j. Lu, Z. Huang, C. Hong, Y. Ma, and H. Qu, "PEGASUS: Bridging Polynomial and Non-polynomial Evaluations in Homomorphic Encryption," in *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP '21)*. IEEE, 2021, pp. 1057–1073.

[45] X. Ren, L. Su, Z. Gu, S. Wang, F. Li *et al.*, "HEDA: Multi-Attribute Unbounded Aggregation over Homomorphically Encrypted Database," *Proceedings of the VLDB Endowment*, vol. 16, no. 4, pp. 601–614, 2022.

[46] S. Bian, Z. Zhang, H. Pan, R. Mao, Z. Zhao *et al.*, "HE3DB: An Efficient and Elastic Encrypted Database Via Arithmetic-And-Logic Fully Homomorphic Encryption," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)*. ACM, 2023, pp. 2930–2944.

[47] N. Volgushev, M. Schwarzkopf, B. Getchell, M. Varia, A. Lapets, and A. Bestavros, "Conclave: secure multi-party computation on big data," in *Proceedings of the 14th European Conference on Computer Systems (EuroSys '19)*. ACM, 2019.

[48] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan *et al.*, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[49] Z. Espiritu, E. A. Markatou, and R. Tamassia, "Time- and Space-Efficient Aggregate Range Queries over Encrypted Databases," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, no. 4, pp. 684–704, 2022.

[50] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.

[51] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. P. Fitzek, and N. Aaraj, "Survey on Fully Homomorphic Encryption, Theory, and Applications," *Proceedings of the IEEE*, vol. 110, no. 10, pp. 1572–1609, 2022.

[52] S. Savvides, D. Khandelwal, and P. Eugster, "Efficient Confidentiality-Preserving Data Analytics over Symmetrically Encrypted Datasets," *Proceedings of the VLDB Endowment*, vol. 13, no. 8, pp. 1290–1303, 2020.

[53] Z. Brakerski, "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP," in *Proceedings of the 32nd Annual Cryptology Conference (CRYPTO '12)*. Springer, 2012, pp. 868–886.

[54] J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," Cryptology ePrint Archive 2012/144, 2012.

[55] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in *Proceedings of the 23rd International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '17)*. Springer, 2017, pp. 409–437.

[56] J. Kun, "Key Switching in LWE," https://www.jeremykun.com/2022/08/29/key-switching-in-lwe/, 2022 (accessed September 25, 2025).

[57] Y. Polyakov, K. Rohloff, G. Sahu, and V. Vaikuntanathan, "Fast Proxy Re-Encryption for Publish/Subscribe Systems," *ACM Transactions on Privacy and Security*, vol. 20, no. 4, 2017.

[58] M. Ayaz, M. F. Pasha, M. Y. Alzahrani, R. Budiarto, and D. Stiawan, "The Fast Health Interoperability Resources (FHIR) Standard: Systematic Literature Review of Implementations, Applications, Challenges and Opportunities," *JMIR Medical Informatics*, vol. 9, no. 7, 2021.

[59] S. M. Huff, R. A. Rocha, C. J. McDonald, G. J. E. De Moor, T. Fiers *et al.*, "Development of the Logical Observation Identifier Names and Codes (LOINC) Vocabulary," *Journal of the American Medical Informatics Association*, vol. 5, no. 3, pp. 276–292, 1998.

[60] I. Iliashenko and V. Zucca, "Faster homomorphic comparison operations for BGV and BFV," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 3, pp. 246–264, 2021.

[61] J. H. Cheon, D. Kim, and D. Kim, "Efficient homomorphic comparison methods with optimal complexity," in *Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '20)*. Springer, 2020, pp. 221–256.

[62] E. Lee, J.-W. Lee, J.-S. No, and Y.-S. Kim, "Minimax Approximation of Sign Function by Composite Polynomial for Homomorphic Comparison," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 6, pp. 3711–3727, 2021.

[63] C. V. Mouchet, J.-P. Bossuat, J. R. Troncoso-Pastoriza, and J.-P. Hubaux, "Lattigo: a Multiparty Homomorphic Encryption Library in Go," in *Proceedings of the 8th Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC '20)*. HomomorphicEncryption.org, 2020, pp. 64–70.

[64] S. Halevi and V. Shoup, "Faster Homomorphic Linear Transformations in HElib," in *Proceedings of the 38th Annual International Cryptology Conference (CRYPTO '18)*. Springer, 2018, pp. 93–120.

[65] J.-P. Bossuat, C. Mouchet, J. Troncoso-Pastoriza, and J.-P. Hubaux, "Efficient Bootstrapping for Approximate Homomorphic Encryption with Non-sparse Keys," in *Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '21)*. Springer, 2021, pp. 587–617.

[66] L. Ducas and D. Micciancio, "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second," in *Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '15)*. Springer, 2015, pp. 617–640.

[67] Z. Liu, D. Micciancio, and Y. Polyakov, "Large-Precision Homomorphic Sign Evaluation Using FHEW/TFHE Bootstrapping," in *Proceedings of the 28th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '22)*. Springer, 2022, pp. 130–160.

[68] A. Al Badawi, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli *et al.*, "OpenFHE: Open-Source Fully Homomorphic Encryption Library," in *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC '22)*. ACM, 2022, pp. 53–63.

[69] F. Boemer, S. Kim, G. Seifu, F. D. de Souza, V. Gopal *et al.*, "Intel HEXL," https://github.com/intel/hexl, 2021.

[70] V. Shoup, "NTL: A Library for doing Number Theory," https://libntl.org, 2001.

[71] SQLite, "SQLite," https://www.sqlite.org/, 2000.

[72] Google LLC, "Protocol Buffers - Google's data interchange format," https://github.com/protocolbuffers/protobuf, 2008.

[73] gRPC, "gRPC - An RPC library and framework," https://grpc.io/, 2015.

[74] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser *et al.*, "Homomorphic Encryption Standard," in *Protecting Privacy through Homomorphic Encryption*, 1st ed. Springer, 2021, ch. 2, pp. 31–62.

[75] J. Takeshita, N. Koirala, C. McKechney, and T. Jung, "HEProfiler: An In-Depth Profiler of Approximate Homomorphic Encryption Libraries," Cryptology ePrint Archive 2024/1059, 2024.

[76] C. Liu, L. Zhu, M. Wang, and Y.-a. Tan, "Search pattern leakage in searchable encryption: Attacks and new construction," *Information Sciences*, vol. 265, pp. 176–188, 2014.

[77] A. Cumyn, J.-F. Ménard, A. Barton, R. Dault, F. Lévesque, and J.-F. Ethier, "Patients' and Members of the Public's Wishes Regarding Transparency in the Context of Secondary Use of Health Data: Scoping Review," *Journal of Medical Internet Research*, vol. 25, no. 1, 2023.

[78] M. Cuggia and S. Combes, "The French Health Data Hub and the German Medical Informatics Initiatives: Two National Projects to Promote Data Sharing in Healthcare," *Yearbook of Medical Informatics*, vol. 28, no. 01, pp. 195–202, 2019.

[79] J. Lohmöller, J. Pennekamp, and K. Wehrle, "Toward Technically Enforceable Consent in Healthcare Research," in *Proceedings of the 9th Privacy Platform (Interdisciplinary Annual Conference) 2024*. Fraunhofer ISI, 2024.

[80] N. Haffer and S. Thun, "Incorrect and Sex-Inconsistent Mapping of Disorders: Identification of Sex Biases in the ICD-10, ICD-11 and SNOMED CT and How to Work Around Them," in *Proceedings of the 34th Medical Informatics Europe Conference (MIE '24)*, vol. 316. IOS Press, 2024, pp. 1458–1462.

[81] R. Husemann, H. F. Wiegand, and L. P. Hölzel, "The new structure and concept of the ICD-11 in the area of mental disorders," *Nervenheilkunde*, vol. 43, no. 04, pp. 160–166, 2024.

[82] J. Zhang, X. Cheng, L. Yang, J. Hu, X. Liu, and K. Chen, "SoK: Fully Homomorphic Encryption Accelerators," *ACM Computing Surveys*, vol. 56, no. 12, 2024.

[83] R. Gay and B. Ursu, "On Instantiating Unleveled Fully-Homomorphic Signatures from Falsifiable Assumptions," in *Proceedings of the 27th IACR International Conference on Practice and Theory of Public-Key Cryptography (PKC '24)*. Springer, 2024, pp. 74–104.

[84] Z. Zhou, Y. Li, Y. Wang, Z. Yang, B. Zhang *et al.*, "ZHE: Efficient Zero-Knowledge Proofs for HE Evaluations," in *Proceedings of the 2025 IEEE Symposium on Security and Privacy (SP '25)*. IEEE, 2025, pp. 3328–3346.

[85] C. Guo, J. Jia, K.-K. R. Choo, and Y. Jie, "Privacy-preserving image search (PPIS): Secure classification and searching using convolutional neural network over large-scale encrypted medical images," *Computers & Security*, vol. 99, 2020.

Appendix A.
Real-World Patient Data and Deployment

Our evaluation of PATDISCOVER features a real-world use case from the area of nuclear medicine that builds on mimicking patient data. Thus, we also need to discuss both ethical and deployment considerations.

Ethics Statement. A medical practitioner identified relevant attributes (cf. Table 3) in a small set of 10 real-world patient letters from a university hospital. The patients consented to their information being used for research. To account for the sensitivity of the data, we decided to implement a synthetic attribute generator, which generates realistic patient data, i.e., attributes. Since our approach does not depend on statistical distributions in the data, we disregarded this dimension and only focused on semantic correctness on a patient level. Consequently, we did not require additional consent or IRB approval to conduct our research. No sensitive data was ever disclosed to third parties.

Deployment Considerations. While we tested PATDISCOVER on real-world data, putting PATDISCOVER into practical deployment on a global scale involves ethical, legal, and regulatory considerations. Although we provide the technical groundwork, addressing these broader issues is beyond our scope. Still, we want to briefly discuss deployment-related considerations in the following.

From a technical standpoint, data updates and modifications need to be handled. In PATDISCOVER, institutions can update existing records anytime by re-encrypting and retransmitting the relevant data batches to the server. Likewise, new records can be dynamically appended. Another important consideration concern erroneous records and heterogeneity caused by data collection systems. By combining the SUM and less-than operators, PATDISCOVER supports query flexibility, permitting slight errors in specific attributes within multi-attribute queries. Nevertheless, input data heterogeneity remains a broader, orthogonal issue relative to our discovery approach. Lastly, competing data standardization initiatives exist, including ICD [29] or LOINC [59] for general health records. In radiology, radiomics approaches extract numeric metrics from images and reports suitable as attributes for PATDISCOVER. However, standardization within these radiomics methods is currently lacking, as the relevant communities have yet to establish consensus.

Appendix B.
Further Details on Applied Operators

To complement our design overview (Section 4.3), we briefly elaborate on the operators that we use while designing our required attribute types.

Precise Less-Than Operator. As detailed in Section 4.2, we use the following univariate interpolation of the less-than function by Iliashenko and Zucca [60] on a finite field $\mathbb{F}_p$ for an odd prime p:

$$\mathrm{LT}(a,b) = \frac{p+1}{2}(a-b)^{p-1} + \sum_{i=1,odd}^{p-2} c_i \cdot (a-b)^i \quad (9)$$

where $c_i = \sum_{j=1}^{\frac{p-1}{2}} j^{p-1-i}$.

Minimax Approximations of the Sign Function. For the approximated Enum attribute type, we use a composite polynomial of degrees 3, 5, 7, 7. For the approximated Range attribute type, we select a composite polynomial of degrees 7, 7, 7, 13 and finally chain the precision boost, as described in Section 4.2. For the approximated Distance matching, we apply a composite polynomial of degrees 7, 7, 7, 7, 9, 9, 15. By relying on this configuration, we end up with a sign approximation with sufficient precision.

Discarded Scheme-Switching Approach. As mentioned in Section 4.3, we also experimented with an alternative scheme-switching approach to realize the precise attribute types. In this alternative approach, we first computed all calculations up to the less-than operators in CKKS [55] and then switched the CKKS ciphertext to many FHEW [66] ciphertexts (one per packed plaintext). On the FHEW ciphertexts, we then conducted the large-precision sign function evaluation proposed by Liu et al. [67]. Once the sign function evaluation concluded, we switched the FHEW ciphertexts back into one CKKS ciphertext and completed the computation and aggregation operations in CKKS.

Although the approach is embarrassingly parallel, it unfortunately does not scale well to large patient numbers. Our evaluation of the corresponding implementation revealed that queries on around 4000 patients already took about one hour to process, i.e., far longer than our reported runtimes (Section 5.3). Therefore, we discarded this scheme-switching approach and replaced it with the discussed large polynomial evaluation in BFV [53], [54] (Section 4.3).

Appendix C.
Exploiting Approximated Attribute Types

The chosen construction of our approximated attribute types introduces a (theoretical) attack vector to PATDISCOVER. Specifically, a malicious client—the attacker—can exploit his knowledge of how the sign function is approximated in our design to submit well-crafted queries during patient discovery. By querying values outside the normalized range ($[0,1]$), he may infer original patient data from the approximated sign function. The attack builds on applying the inverse of the approximation function to the discovery server's computed result. Fortunately, the inherent loss of precision when computing complex arithmetic operations in CKKS complicates the extraction of accurate patient data in practice. Hence, the HE scheme's peculiarities render the attack unlikely in practice.

The described attack vector only manifests in the approximated attribute types. The precise variants are robust.