

# A Multiparty Homomorphic Encryption Approach to Confidential Federated Kaplan–Meier Survival Analysis

Narasimha Raghavan Veeraragavan<sup>1</sup>, Svetlana Boudko<sup>2</sup>,  
Jan Franz Nygård<sup>1,3</sup>

<sup>1</sup>Cancer Registry of Norway, Norwegian Institute of Public Health,  
Oslo, Norway.

<sup>2</sup>Norwegian Computing Center, Oslo, Norway.

<sup>3</sup>Department of Physics and Technology, The Arctic University of  
Norway, Tromsø, Norway.

Contributing authors: [Narasimha.Raghavan@kreftregisteret.no](mailto:Narasimha.Raghavan@kreftregisteret.no);  
[svetlana@nr.no](mailto:svetlana@nr.no); [Jan.Nygard@kreftregisteret.no](mailto:Jan.Nygard@kreftregisteret.no);

## Abstract

The proliferation of real-world healthcare data has substantially expanded opportunities for collaborative research, yet stringent privacy regulations hinder the pooling of sensitive patient records in a single location. To address this dilemma, we propose a *multiparty homomorphic encryption-based* framework for *privacy-preserving federated Kaplan–Meier survival analysis*, surpassing existing methods by offering native floating-point support, a detailed theoretical model, and explicit mitigation of reconstruction attacks.

Compared to prior work, our framework provides a more comprehensive analysis of noise growth and convergence, guaranteeing that the *encrypted federated survival estimates closely match centralized (unencrypted) outcomes*. Formal utility-loss bounds demonstrate that as aggregation and decryption noise diminish, the encrypted estimator converges to its unencrypted counterpart. Extensive experiments on the NCCTG Lung Cancer and a synthetic Breast Cancer dataset confirm that the *mean absolute error (MAE) and root mean squared error (RMSE) remain low*, indicating only negligible deviations between encrypted and non-encrypted federated survival curves. Log-rank tests further reveal *no significant difference* between federated encrypted and non-encrypted analyses, thereby preserving statistical validity. Additionally, an in-depth reconstruction-attack evaluation shows that smaller federations (2–3 providers) with overlapping data

are acutely vulnerable, a challenge our multiparty encryption *effectively neutralizes*. Larger federations (5–50 sites) inherently degrade reconstruction accuracy, yet encryption remains prudent for maximum confidentiality.

Despite an overhead factor of 8–19× compared to non-encrypted computation, our results show that threshold-based homomorphic encryption is *feasible for moderate-scale deployments*, balancing security needs with acceptable runtime. By furnishing robust privacy guarantees alongside high-fidelity survival estimates, this framework significantly advances the state of the art in secure, multi-institutional survival analysis.

**Keywords:** Kaplan–Meier Survival Analysis, Federated Analytics, Threshold Homomorphic Encryption, Privacy-Preserving Technologies

## 1 Introduction

In today’s data-driven world, the healthcare sector stands to benefit greatly from advanced data sharing, analytics, and artificial intelligence. Nevertheless, the sensitive nature of medical data demands robust privacy and security assurances. In response, federated learning and privacy-preserving cryptographic techniques such as homomorphic encryption have emerged as key enablers for multi-institutional collaboration, ensuring compliance with stringent regulations. Among these methods, federated Kaplan–Meier survival analysis has garnered attention for its effectiveness in estimating patient survival probabilities, a critical element in clinical research.

Recent advances in Multiparty Homomorphic Encryption (MHE) [1] and Fully Homomorphic Encryption (FHE) [2] enable computations on encrypted data without intermediate decryption, significantly reducing privacy risks. The FAMHE framework [1] demonstrated scalability in federated survival analysis with integers, supporting up to 96 providers. In parallel, Geva *et al.* [2] showcased multiparty FHE for privacy-preserving analysis of oncological datasets, including Kaplan–Meier and log-rank tests.

However, despite these milestones, several challenges persist. The FAMHE framework proposed in [1] lacks native floating-point arithmetic, limiting its use for real-world datasets requiring scaled survival times and advanced statistical workflows. While Geva *et al.* [2] provided empirical results, they did not offer a detailed theoretical analysis of noise growth, utility loss, and scalability across large federations. Moreover, neither baseline explicitly addresses vulnerabilities to reconstruction attacks in federated environments, particularly when data overlap occurs among providers.

**Data overlap** refers to scenarios where patient records appear—either partially or fully—at multiple institutions. In real-world healthcare, such overlap can arise when patients move between hospitals, receive referrals from specialists, or otherwise share treatment plans across sites. Whenever a substantial fraction of patient information is shared across sites, an adversarial party can exploit that knowledge to “subtract out” its own local data from the global aggregates, thereby inferring other sites’ private counts or statistics. Consequently, greater overlap typically increases the risk

of *subtraction-based* (reconstruction) attacks, as there are fewer truly unique patient records to obscure the contributions from other institutions.

To address these gaps, we propose a novel framework for federated Kaplan–Meier survival analysis leveraging multikey CKKS encryption and a star-based topology for distributed key sharing. Our approach provides a rigorous theoretical foundation for analyzing utility loss and noise growth, thereby ensuring both privacy and scalability. Further, we evaluate the method’s robustness against reconstruction attacks, quantify privacy risks, and demonstrate its capacity to scale to large federations.

## 1.1 Summary of Contributions

In comparison with earlier baselines [1, 2], our work achieves several notable advancements in multiparty homomorphic encryption-based, privacy-preserving federated Kaplan–Meier analysis:

### 1. Stronger Privacy Guarantees

- **Reconstruction Attack Mitigation:** While prior studies largely demonstrated the feasibility of federated approaches, they did not deeply address vulnerabilities stemming from overlapping datasets. By incorporating multikey CKKS encryption and quantifying the risks of reconstruction attacks, this work explicitly tackles and *mitigates* a critical privacy gap that previous methods left underexplored.

### 2. Robust Floating-Point Support

- **CKKS Integration:** Whereas prior frameworks [1] frequently depended on integer-based homomorphic encryption schemes (e.g., BFV or BGV), our approach leverages *CKKS for native floating-point operations*, enabling more precise survival times, log-rank tests, and confidence intervals. This obviates the need for coarse numeric approximations inherent in integer-only schemes.

### 3. Comprehensive Theoretical Framework

- **Noise Modeling and Convergence:** Earlier approaches mostly relied on empirical evaluations or partial analyses of noise growth. Here, we develop *formal noise-growth models, utility-loss bounds, and convergence guarantees* for homomorphically encrypted survival estimates, thereby extending both the rigor and breadth of existing methods.

### 4. Empirical Generalizability

- **Diverse Datasets & Scenarios:** Beyond single or toy datasets, this work applies its method to both a smaller real-world dataset (NCCTG Lung Cancer) [3] [4] and a larger synthetic breast cancer dataset [5]—under various federation sizes and overlap conditions. Such extensive testing *demonstrates generalizability* and highlights performance trends previously under-documented.

### 5. Measured Scalability

- **Realistic Performance Assessments:** While prior studies [1] have shown federations up to specific sizes (e.g., 96 providers), our work offers a *detailed performance analysis* across up to 50 clients, revealing how computational overhead scales. This enables practitioners to *weigh the trade-offs* of incorporating encryption in real-world multi-institutional collaborations.

Overall, these contributions *push the state of the art* by reinforcing privacy (especially under partial data overlaps), providing a more rigorous theoretical underpinning, incorporating floating-point arithmetic out of the box, and demonstrating empirical feasibility across diverse datasets and sizable federations.

## 2 Related Work

Federated survival analysis seeks methods for implementing survival models across distributed datasets while respecting local data ownership. Several studies have explored federated survival analysis under both horizontal and vertical data partitioning [6–9]. These works addressed challenges such as decentralized modeling and aggregation of survival estimates, yet they largely did not consider data security or potential data leakage threats. Consequently, although these approaches facilitate collaboration among healthcare institutions, they do not incorporate advanced cryptographic safeguards to prevent misuse or unauthorized inference of sensitive patient-level information. Furthermore, most existing solutions lack a rigorous theoretical foundation for modeling the impact of distributed computations on survival estimates, such as noise growth or utility-loss analyses under partial federated aggregations.

In contrast, research on *confidential* federated analytics has mostly focused on federated learning (FL), where homomorphic encryption is employed to protect gradients during model updates. Pan *et al.* [10] introduced FedSHE, an FL scheme using Adaptive Segmented CKKS Homomorphic Encryption to guard against gradient leakage at the aggregation server. However, they assume that all clients (and the Key Management Center) are trusted and do not collude, thus limiting its scope in adversarial scenarios. Similar assumptions arise in [11–13], where a *single* public-private key pair is shared by multiple clients, again presuming that the clients and/or a trusted key generator will not abuse or collude to reveal private data. While [11] combines homomorphic encryption with authenticated encryption to verify aggregation integrity, and [13] selectively encrypts privacy-sensitive parameters, both rely on a single-key setup. A more general multiparty approach is briefly mentioned in [13], yet the evaluation is confined to communication cost comparisons. Likewise, [14] discusses a single-key homomorphic solution for genomic analyses, noting only in passing how a multiparty extension might be realized. Other studies, such as [15] and [16], also employ single-key or somewhat-homomorphic encryption in FL but assume minimal adversarial behavior.

The framework introduced by Froelicher *et al.* [1] leverages a multiparty encryption library (Lattigo) for genomic data, enabling collaborative survival curve computation, yet it does not fully explore advanced threshold key-management schemes or nuanced reconstruction-attack scenarios. Additionally, while this line of work partly addresses the theoretical underpinnings of homomorphic encryption in distributed settings, deeper analyses of noise propagation, convergence guarantees, and privacy thresholds in survival models remain relatively limited or empirical in nature.

Overall, although prior studies demonstrate the feasibility of federated survival models and basic integration of homomorphic encryption for FL, the challenge of *strong adversarial models*—particularly those involving partial collusion, reconstruction attacks, and formal theoretical guarantees—remains underexplored in existing approaches.

## 3 Background

### 3.1 Homomorphic Encryption

Homomorphic encryption, as introduced by Rivest et al. (1978) [17], is a form of encryption that enables computations to be performed directly on ciphertexts. The resulting encrypted output, when decrypted, corresponds to the outcome of operations applied to the plaintext. This unique property of homomorphic encryption makes it highly valuable in the fields of data privacy and federated analytics, where sensitive data are processed.

The first practical Fully Homomorphic Encryption (FHE) scheme was proposed by Craig Gentry in 2009 [18]. Since then, various homomorphic encryption schemes have been introduced, each aiming to enhance computational efficiency [19–22]. These schemes were initially proposed as single-key homomorphic encryption methods. Although useful in several scenarios, single-key systems are not suitable for federated analytics. In federated analytics, different clients need their own unique secret keys to ensure that their data remains inaccessible to other parties.

The problem can be addressed by utilizing threshold homomorphic encryption [23, 24] that combines the principles of homomorphic encryption with threshold cryptography [25–27]. Threshold homomorphic encryption enables multiple parties to jointly perform computations on encrypted data without revealing the underlying plaintext to any individual party. The decryption of the resulting ciphertext requires a minimum number of parties, known as the threshold, to collaborate. Each party holds a share of the decryption key, and the data can be decrypted only when a sufficient number of key shares are combined. This mechanism enhances security by preventing any single party from accessing the complete decryption key.

Several threshold homomorphic encryption schemes have been introduced in the literature and are available as open-source libraries [28, 29]. These schemes support arithmetic operations on both complex numbers and integers using the single-instruction, multiple-data (SIMD) approach. As a result, multiple data points can be encrypted within a single ciphertext. This enables homomorphic operations to be executed simultaneously in a component-wise manner across multiple datasets.

The Brakerski-Gentry-Vaikuntanathan (BGV) [19] and Brakerski/Fan-Vercauteren (BFV) [20, 30] schemes are based on the hardness of the Ring Learning with Errors (Ring-LWE) problem, which is a variant of the Learning with Errors problem tailored for rings. The BGV scheme was introduced to address some of the computational inefficiencies found in earlier homomorphic encryption systems. The innovations in managing noise and enabling modulus switching allow it to perform operations faster and with lower computational overhead compared to the first-generation FHE schemes. While the BGV scheme is not inherently equipped to handle unlimited operations due to noise growth, it supports levelled homomorphic encryption. This means that it can handle any number of operations up to a predefined depth. The depth, related to the number of layers in arithmetic circuits that can be evaluated, must be defined in advance during the setup of the encryption parameters. The BFV scheme, a scale-invariant construction, exhibits the same noise growth characteristics as the BGV scheme. Both schemes are specifically designed to perform complex arithmetic over integer arithmetic circuits.

The Ducas-Micciancio (FHEW) [31] and the Chillotti-Gama-Georgieva-Izabachene (CGGI) [21] schemes are specifically designed to support the encryption of small bit-width integers and are optimized for Boolean circuit evaluation. In the FHEW scheme, the authors introduced an efficient bootstrapping technique that significantly reduces the noise level. In the CGGI scheme, the bootstrapping speed is improved by implementing programmable bootstrapping. This computational operation, performed on a ciphertext during the bootstrapping phase, also reduces noise in ciphertext processing.

The Cheon-Kim-Kim-Song (CKKS) scheme [32], is another threshold homomorphic encryption scheme that facilitates approximate homomorphic computations. One of the novel features of the CKKS scheme is its use of a rescaling operation. This operation is crucial for managing the growth of noise in ciphertexts during multiplicative operations. In homomorphic encryption, noise is introduced with each operation performed, and if it grows too large, it can prevent accurate decryption. The rescaling operation in CKKS helps mitigate this issue by scaling down the ciphertext and the noise, thereby enabling deeper computation circuits. Like the BGV scheme, this scheme supports levelled homomorphic encryption. Unlike previously mentioned homomorphic encryption schemes that were limited to integer arithmetic, CKKS allows for approximate calculations on encrypted real and complex numbers. This capability is particularly important for applications involving scientific computations, statistics, and machine learning algorithms that require handling of non-integer data.

It is also worth mentioning that these schemes fall under the umbrella of lattice-based cryptography. Currently, there are no known efficient quantum algorithms capable of solving hard lattice problems, which makes lattice-based methods robust against both classical and quantum attacks.

## 3.2 Survival Analysis: Kaplan Meier Estimation

Kaplan-Meier estimation is a widely used non-parametric method for analyzing time-to-event data, particularly in survival analysis. Originally introduced by Edward L. Kaplan and Paul Meier in 1958, the Kaplan-Meier estimator provides a way to estimate the survival function, which represents the probability that a given event (e.g., death, disease relapse, or equipment failure) occurs after a specified time. The estimator is especially useful in fields like medical research, engineering, and reliability analysis, where understanding the duration until an event occurs is critical.

### 3.2.1 Survival Function and Time-to-Event Analysis

The primary object of interest in Kaplan-Meier estimation is the *survival function*, denoted by  $S(t)$ . The survival function  $S(t)$  gives the probability that an individual or object will survive beyond a certain time  $t$ :

$$S(t) = P(T > t), \tag{1}$$

where  $T$  is the random variable representing the time-to-event. The survival function  $S(t)$  typically decreases over time as the probability of an event occurring increases. In many cases, the survival curve, which plots  $S(t)$  against  $t$ , is used to visualize the probability of surviving over time.

### 3.2.2 Key Concepts in Kaplan-Meier Estimation

Kaplan-Meier estimation is based on three main concepts:

- **Event Times:** The specific times at which events (e.g., deaths) occur in the dataset. These times define the points at which the survival probability  $S(t)$  is updated.
- **At-Risk Population:** The number of individuals or units still at risk of experiencing the event just before each event time. This count decreases over time as events occur or individuals are censored.
- **Censored Data:** Censoring occurs when the exact event time is unknown for certain individuals, either because they left the study before experiencing the event or the study ended before the event occurred. Censoring is handled naturally in Kaplan-Meier estimation by adjusting the at-risk population accordingly, ensuring that individuals who are censored contribute to survival estimates only for the time they were observed.

### 3.2.3 Methodology of the Kaplan-Meier Estimator

The Kaplan-Meier estimator calculates the survival function  $\hat{S}(t)$  by estimating the probability of survival at each observed event time. Let  $t_i$  denote the ordered event times in the dataset,  $d_i$  the number of events occurring at  $t_i$ , and  $n_i$  the number of individuals at risk just before  $t_i$ . The probability of surviving beyond  $t_i$  is given by  $1 - \frac{d_i}{n_i}$ , representing the proportion of at-risk individuals who survive past  $t_i$ . The Kaplan-Meier estimator is defined as:

$$\hat{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right), \quad (2)$$

where the product is taken over all event times  $t_i$  up to  $t$ .

### 3.2.4 Interpreting the Kaplan-Meier Curve

The Kaplan-Meier curve is a step function that decreases at each event time  $t_i$ , representing the cumulative survival probability. Between event times, the survival probability remains constant, resulting in a series of horizontal steps. This curve provides valuable insights into the probability of survival over time, making it possible to compare survival rates across different groups, treatments, or conditions. Additionally, the Kaplan-Meier estimator can be used to compute median survival times and generate confidence intervals around survival estimates.

## 3.3 Applications of Kaplan-Meier Estimation

The Kaplan-Meier estimator is extensively applied in medical research to evaluate the effectiveness of treatments, analyze patient survival, and compare different treatment groups. In engineering, Kaplan-Meier estimation is used in reliability testing to assess the durability of components or systems over time. In these applications, handling censored data is essential, as individuals or units may not experience the event by the end of the observation period, and Kaplan-Meier estimation effectively incorporates this aspect.

Despite its simplicity and non-parametric nature, Kaplan-Meier estimation has proven to be a powerful tool for survival analysis. However, as data privacy regulations have become more stringent, especially in healthcare, federated implementations of Kaplan-Meier estimation are increasingly needed. Such implementations enable

institutions to collaborate on survival analysis without sharing sensitive patient data directly, motivating the development of secure computation techniques to support privacy-preserving Kaplan–Meier estimation.

## 4 Problem Formulation

In a federated environment, multiple institutions  $\{I_1, I_2, \dots, I_K\}$  aim to collaboratively estimate a global Kaplan–Meier (KM) survival function  $\hat{S}(t)$  without directly sharing their sensitive data (e.g., individual time-to-event records). Each institution  $I_k$  (for  $k = 1, \dots, K$ ) maintains a local dataset  $D_k$  containing:

- **Event times:**  $\{t_i^{(k)}\}$ , the distinct times at which an event (e.g., death or failure) occurs in  $I_k$ 's dataset.
- **Event counts:**  $d_i^{(k)}$ , the number of events happening at each local event time  $t_i^{(k)}$ .
- **At-risk counts:**  $n_i^{(k)}$ , the number of individuals at risk just before each event time  $t_i^{(k)}$ .

To form the global KM estimator, we first collect the *union* of all local event times into a set of unique times  $\{t_i\}_{i=1}^M$ , sorted in ascending order. For each  $t_i$ , let

$$d_i = \sum_{k=1}^K d_i^{(k)}, \quad n_i = \sum_{k=1}^K n_i^{(k)},$$

where  $d_i^{(k)}$  and  $n_i^{(k)}$  default to 0 if  $t_i$  does not appear in institution  $I_k$ 's dataset. The *global* Kaplan–Meier survival function then follows the standard definition:

$$\hat{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right).$$

This yields the estimated probability that a randomly chosen subject from the *combined* population remains event-free up to time  $t$ .

The central challenge in this federated setting is computing the aggregated counts  $d_i$  and  $n_i$  *securely* across all institutions without revealing any institution's underlying records, thus satisfying privacy regulations and preserving local data autonomy. In the subsequent sections, we describe how threshold homomorphic encryption enables the necessary secure aggregation of event and at-risk counts, culminating in a joint survival estimate that closely matches what would be obtained in a standard, centralized KM analysis.

### 4.1 Challenges in Federated Kaplan–Meier Estimation

Although federated Kaplan–Meier analysis avoids centralizing raw patient-level data, several key obstacles arise when securely computing the global at-risk and event counts:

1. **Data Privacy Across Institutions:** Simply sharing raw at-risk counts  $n_i^{(k)}$  and event counts  $d_i^{(k)}$  can lead to privacy breaches, particularly in institutions with small populations. Regulations such as GDPR and HIPAA mandate stringent data confidentiality practices, prohibiting patient-level detail exchange.



2. **Secure Aggregation Without Decryption:** Traditional systems typically decrypt data before computation, reintroducing privacy risks. In federated KM estimation, we must aggregate encrypted counts so that neither the server nor other participants learn each institution’s local values.
3. **Decentralized Key Management:** Many federated environments require each institution to retain control over its own data and cryptographic keys, rather than relying on a single trusted authority. This motivates threshold or multiparty encryption schemes, wherein partial decryption shares are combined to recover the aggregated result without granting any one party full decryption power.
4. **Reconstruction Attacks with Overlapping Data:** Even when institutions share only aggregated statistics, an adversarial party can attempt a *reconstruction attack* by subtracting its own local counts from the federated totals. Such inferences become especially potent when patient records overlap across multiple sites. In Section 7.4, we further examine how data overlap and federation size impact reconstruction accuracy and present empirical evidence of this vulnerability.

To address these challenges, we propose a threshold homomorphic encryption approach that preserves data privacy during aggregation and mitigates the risks posed by reconstruction attacks, as detailed in the subsequent sections.

## 5 Proposed Solution

In this section, we detail a threshold homomorphic encryption (HE) framework designed to enable privacy-preserving, federated Kaplan-Meier estimation across multiple institutions. We begin by outlining the core assumptions, notations, and threat model, followed by a description of the distributed key generation process and, finally, the federated Kaplan Meier computation steps under the threshold HE scheme.

### 5.1 Assumptions, Definitions, and Notations

#### 5.1.1 Threat Model Assumptions

The proposed framework is specifically designed to facilitate collaborative analysis and encrypted sensitive data sharing among health care institutions, with only authorized personnel allowed to add and process data, and initiate and carry out the federated procedures. Given these circumstances, we anticipate that the inherent boundaries of the Honest-but-Curious threat model will apply. We consider the following assumptions.

1. **Semi-Honest (Honest-but-Curious) Participants:** Institutions and the central server follow the protocol faithfully (no active data tampering or insertion of malicious operations), yet they may attempt to glean additional information from the data or partial decryptions they are legitimately allowed to see.
2. **Limited (Sub-Threshold) Collusion:** We assume no coalition of institutions large enough to meet or exceed the threshold  $T$  colludes to combine their secret key shares and decrypt data without authorization. If fewer than  $T$  participants collude, they cannot decrypt the aggregated ciphertexts. Those holding enough shares to surpass the threshold are presumed to adhere to legitimate protocol steps rather than colluding maliciously.
3. **No Malicious Tampering:** While participants may be curious, we assume they do not intentionally alter data, generate invalid partial decryptions, or otherwise

disrupt protocol integrity. The system functions in a semi-honest environment, not a fully malicious one.

4. **Stable Federation Membership:** The set of institutions taking part in the analysis remains fixed throughout the key generation and partial decryption phases. Institutions do not join or leave mid-protocol, avoiding complexities with dynamic key sharing and membership changes.

Under these assumptions, threshold homomorphic encryption and secure multi-party protocols prevent any single entity (including the central server) from decrypting sensitive data. This design thereby safeguards individual-level information, particularly in scenarios where curious participants might attempt to infer hidden details from aggregated counts or partial decryption results.

### 5.1.2 Notations and Key definitions

Let:

- $K$  denote the number of participating institutions.
- $I_k$  be the  $k$ -th institution (for  $k = 1, \dots, K$ ).
- Each institution  $I_k$  (also called a *local party*, *client*, or *data provider*) holds its own local dataset  $D_k$ , consisting of:
  - $\{t_i^{(k)}\}$ : the event times for each observed event. We treat these as *non-sensitive*, meaning they do not require encryption. A global (federated) set of time points is then formed by aggregating all  $\{t_i^{(k)}\}$  in the clear (i.e., unencrypted) across participating institutions, consistent with the assumptions of the FAMHE framework [1].
  - $d_i^{(k)}$ : the number of events at each time  $t_i^{(k)}$ ,
  - $n_i^{(k)}$ : the number of individuals at risk just prior to each event time  $t_i^{(k)}$ .
- $T$  denotes the threshold specifying the minimum number of institutions required to perform decryption.
- All institutions share a common public key  $pk$  for the threshold homomorphic encryption scheme, while each institution  $I_k$  holds a unique decryption key share  $sk_k$ .
- Only a subset of at least  $T$  institutions can collectively decrypt data; fewer than  $T$  key shares are insufficient for decryption.

### 5.2 Distributed Key Generation Process

To enable threshold-based encryption and decryption, we employ a distributed key generation process. While there are different possible network topologies (e.g., star-based vs. ring-based), in this work, we focus on a **star-based** approach.

- **Star-Based Topology:**
  - A central server (coordinator) manages the key generation steps, instructing each institution in sequence to update a shared public key.
  - After all designated institutions have contributed, the final public key is distributed to every participant.
- **Ring-Based Topology (Not Implemented Here):**
  - Each institution would be connected to exactly two others in a ring structure, passing updated keys in a loop.

- Further discussion of the pros and cons of these topologies is outside the scope of this paper.

---

**Algorithm 1** Central Server Algorithm for Distributed Key Generation
 

---

- 1: Selects the threshold group of participating institutions
  - 2: Generates a cryptographic context  $cc$  containing homomorphic encryption parameters
  - 3: Sets public key  $pk$  equals None
  - 4: **for** each institution  $I_k$  in the threshold group **do**
  - 5:     Send  $pk$  to institution  $I_k$  and  $cc$
  - 6:     Receive updated public key,
  - 7:      $pk =$  updated public key
  - 8: **end for**
  - 9: Final public key equals  $pk$
  - 10: Send final public key to all participating institutions
- 

---

**Algorithm 2** Local Party Algorithm for Distributed Key Generation
 

---

- 1: Receive current public key from Central Server and cryptographic context  $cc$
  - 2: Generate new public key and secret key share
  - 3: Store secret key share locally
  - 4: Send new public key to Central Server
  - 5: Receive final public key from Central Server and stores it locally
- 

The central server initiates the key generation process by establishing homomorphic encryption parameters and specifying the sequence in which participating institutions should generate the common public key and their individual secret key shares. Upon receiving the homomorphic encryption parameters and the updated public key, each participating institution employs these elements to generate a new version of the public key, which is then returned to the server. The final public key obtained through this process becomes the common public key distributed among all participants and is used for encryption. This process requires serialization/deserialization of the cryptographic context objects and keys before exchanging them among participating institutions.

In this work, the scope of the homomorphic operations is limited to additive operations. However, in some implementations, particularly if multiplicative or inverse operations are required for weighted aggregations, additional **evaluation keys** may be generated.

### 5.3 Algorithm for Federated Kaplan-Meier Estimation

Once the distributed key generation is complete, each institution (local party/client/data provider) holds a local secret key share, and a shared public key is available to all parties.

### 5.3.1 Algorithm for Each Local Party (Institution $I_k$ )

---

**Algorithm 3** Local Party Algorithm for Threshold Homomorphic Encryption

---

- 1: **Public Key Setup:** A shared public key  $pk$  is generated for the threshold homomorphic encryption scheme.
- 2: **for** each event time  $t_i^{(k)}$  in local dataset  $D_k$  **do**
- 3:     Encrypt the local event count  $d_i^{(k)}$  using  $pk$ :

$$\text{Encrypted}(d_i^{(k)}) = \text{Encrypt}(d_i^{(k)}, pk)$$

- 4:     Encrypt the local at-risk count  $n_i^{(k)}$  using  $pk$ :

$$\text{Encrypted}(n_i^{(k)}) = \text{Encrypt}(n_i^{(k)}, pk)$$

- 5: **end for**
  - 6: Send  $\text{Encrypted}(d_i^{(k)})$  and  $\text{Encrypted}(n_i^{(k)})$  for each event time  $t_i^{(k)}$  to the central server.
- 

Each participating institution encrypts its own event and at-risk counts before sending them to the central server. By doing so, the raw data never leave the institution in plaintext form. Using the shared public key  $pk$ , each institution transforms its local counts  $\{d_i^{(k)}, n_i^{(k)}\}$  into ciphertexts. This ensures that even if messages are intercepted or the central server is compromised, individual patient-level information remains protected. The local parties thus retain control over their data while still contributing to the federated Kaplan–Meier computation.

### 5.3.2 Algorithm for the Central Server

---

**Algorithm 4** Central Server Algorithm for Threshold Homomorphic Encryption

---

- 1: **Receive Encrypted Data:** For each event time  $t_i$ , collect encrypted event counts  $\{\text{Encrypted}(d_i^{(k)})\}$  and at-risk counts  $\{\text{Encrypted}(n_i^{(k)})\}$  from all institutions.
- 2: **for** each event time  $t_i$  **do**
- 3:     Perform homomorphic addition to aggregate encrypted event counts:

$$\text{AggregatedEncrypted}(d_i) = \sum_{k=1}^K \text{Encrypted}(d_i^{(k)})$$

- 4:     Perform homomorphic addition to aggregate encrypted at-risk counts:

$$\text{AggregatedEncrypted}(n_i) = \sum_{k=1}^K \text{Encrypted}(n_i^{(k)})$$

- 5: **end for**
  - 6: Distribute aggregated encrypted values  $\text{AggregatedEncrypted}(d_i)$  and  $\text{AggregatedEncrypted}(n_i)$  for each  $t_i$  to the decryption group (a subset of institutions satisfying the threshold  $T$ ).
- 

The central server acts primarily as a secure aggregator in this threshold homomorphic encryption setup. It receives encrypted event and at-risk counts from each institution and performs the necessary homomorphic additions to obtain aggregate ciphertexts—without ever learning the underlying plaintext values. Once these aggregated ciphertexts are computed, the server distributes them to the threshold group for partial decryption, thus coordinating the overall process without having access to raw patient data at any point. Throughout this aggregation, no weighting is applied, ensuring that all institutions' data is treated equally.

### 5.3.3 Collaborative Decryption and Kaplan-Meier Computation

---

**Algorithm 5** Collaborative Decryption and Kaplan-Meier Calculation
 

---

- 1: **for** each event time  $t_i$  **do**
- 2:     **for** each institution  $I_k$  in the threshold group **do**
- 3:         Compute decryption shares for event and at-risk counts:
 
$$\text{Share}_{d_i}^{(k)} = \text{DecryptShare}(\text{AggregatedEncrypted}(d_i), sk_k)$$

$$\text{Share}_{n_i}^{(k)} = \text{DecryptShare}(\text{AggregatedEncrypted}(n_i), sk_k)$$
- 4:         Send decryption shares  $\text{Share}_{d_i}^{(k)}$  and  $\text{Share}_{n_i}^{(k)}$  to the decryption coordinator.
- 5:     **end for**
- 6:     Combine decryption shares from at least  $T$  institutions to obtain decrypted values  $d_i$  and  $n_i$ :
 
$$d_i = \text{CombineShares}(\{\text{Share}_{d_i}^{(k)}\}_{k \in T})$$

$$n_i = \text{CombineShares}(\{\text{Share}_{n_i}^{(k)}\}_{k \in T})$$
- 7: **end for**
- 8: Compute the Kaplan-Meier survival probability:

$$S(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$


---

Once decrypted, each aggregated count  $d_i$  and  $n_i$  can be used to compute the classical KM estimator. Because only the threshold group can recover these plaintext values, confidentiality is preserved under the assumption that no unauthorized coalition meets or exceeds  $T$  key shares.

### 5.4 Theoretical analysis

**Theorem 1** (Utility Loss Bound in Federated Homomorphic Encrypted Kaplan-Meier Estimators for Worst Case Scenario). *Let  $\hat{S}_{centralized}(t)$  and  $\hat{S}_{federated}(t)$  denote the Kaplan-Meier survival probabilities estimated at time  $t$  in centralized and federated homomorphic encrypted (HE) settings, respectively. Assume:*

1. The event counts  $d_i^{(k)}$  and at-risk counts  $n_i^{(k)}$  from client  $k$  are encrypted and aggregated using CKKS homomorphic encryption.
2. Noise is introduced during two distinct phases:
  - (a) **Aggregation Noise**  $\varepsilon_{aggregation}$ : Noise from ciphertext operations, such as addition and multiplication, which depends on the total number of ciphertexts contributed by all clients.
  - (b) **Decryption Noise**  $\varepsilon_{decryption}$ : Noise from partial decryption and multiparty decryption fusion, involving a subset of  $T \leq K$  clients.
3. Each client contributes a dataset of size  $D_k$ , resulting in  $M_k$  ciphertexts proportional to  $D_k$ .

4. A common ciphertext modulus  $q$  is used across all clients.
5. At-risk counts  $n_i > 0$  for all  $t_i$ , as zero would render survival probabilities undefined.

Under these assumptions, the utility loss  $\Delta S(t)$  and error growth can be analyzed as follows:

**Utility Loss Bound:**

$$\Delta S(t) = |\hat{S}_{\text{centralized}}(t) - \hat{S}_{\text{federated}}(t)| \leq \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \cdot \sum_{t_i \leq t} \frac{O(\log(q) \cdot \sum_{k=1}^K M_k) + O(\sigma \cdot \sqrt{T})}{n_i + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}, \quad (3)$$

where  $M_k \propto D_k$  reflects the relationship between dataset size and the number of ciphertexts.

For large  $n_i$ , the bound simplifies to:

$$\Delta S(t) \leq \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \cdot \sum_{t_i \leq t} \frac{O(\log(q) \cdot \sum_{k=1}^K M_k) + O(\sigma \cdot \sqrt{T})}{n_i}. \quad (4)$$

*Proof.* 1. **Centralized Kaplan-Meier Survival Probability:**

- The centralized Kaplan-Meier survival probability is calculated as:

$$\hat{S}_{\text{centralized}}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right), \quad (5)$$

where  $d_i$  is the event count, and  $n_i$  is the at-risk count at time  $t_i$ .

2. **Federated Kaplan-Meier Survival Probability:**

- In the federated setup, the survival probability is computed using encrypted event counts  $d_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)$  and at-risk counts  $n_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)$ .
- The federated survival probability is:

$$\hat{S}_{\text{federated}}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}{n_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}\right). \quad (6)$$

3. **Error in Survival Probability:**

- The utility loss  $\Delta S(t)$  is defined as the absolute difference:

$$\Delta S(t) = |\hat{S}_{\text{centralized}}(t) - \hat{S}_{\text{federated}}(t)|. \quad (7)$$

- Expanding both terms:

$$\Delta S(t) = \left| \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) - \prod_{t_i \leq t} \left(1 - \frac{d_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}{n_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}\right) \right|. \quad (8)$$

#### 4. Bounding Noise Growth:

- Aggregation noise reflects the contribution from all clients:

$$\varepsilon_{\text{aggregation}}(t_i) = O\left(\log(q) \cdot \sum_{k=1}^K M_k\right), \quad (9)$$

where  $M_k \propto D_k$ .

- Decryption noise depends on the subset  $T$  of clients involved in decryption:

$$\varepsilon_{\text{decryption}}(t_i) = O(\sigma \cdot \sqrt{T}), \quad (10)$$

where  $\sigma$  is the standard deviation of noise during decryption.

- Total noise combines aggregation and decryption noise:

$$\varepsilon(t_i) = O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T}). \quad (11)$$

#### 5. Final Utility Loss Bound:

- Substituting the noise terms into the cumulative error:

$$\Delta S(t) \leq \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \cdot \sum_{t_i \leq t} \frac{O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T})}{n_i + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}. \quad (12)$$

- For large  $n_i$ :

$$\Delta S(t) \leq \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \cdot \sum_{t_i \leq t} \frac{O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T})}{n_i}. \quad (13)$$

□

The utility loss bound explicitly accounts for both aggregation and decryption noise. Proper parameter tuning, including scaling factors, batch sizes, and noise distributions, is essential to minimizing this loss.

**Theorem 2** (Utility Loss Bound for Bounded At-Risk Counts with Noise Terms). *If the at-risk counts  $n_i$  are bounded below by a constant  $c > 0$ , and the noise terms from aggregation and decryption are included in the denominator, the utility loss bound can*



be expressed as:

$$\Delta S(t) \leq \frac{1}{c} \cdot \sum_{t_i \leq t} \frac{O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T})}{1 + \frac{O(\log(q) \cdot \sum_{k=1}^K M_k) + O(\sigma \cdot \sqrt{T})}{n_i}}. \quad (14)$$

For sufficiently large  $n_i$ , the bound simplifies to:

$$\Delta S(t) \leq \frac{1}{c} \cdot \sum_{t_i \leq t} \left( O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T}) \right). \quad (15)$$

*Proof.* 1. **Utility Loss Bound with At-Risk Counts and Noise Terms:**

- From the utility loss theorem, the bound is:

$$\Delta S(t) \leq \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \cdot \sum_{t_i \leq t} \frac{\varepsilon(t_i)}{n_i + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}. \quad (16)$$

- Substituting the expressions for  $\varepsilon_{\text{aggregation}}(t_i)$  and  $\varepsilon_{\text{decryption}}(t_i)$ :

$$\varepsilon_{\text{aggregation}}(t_i) = O\left(\log(q) \cdot \sum_{k=1}^K M_k\right), \quad \varepsilon_{\text{decryption}}(t_i) = O(\sigma \cdot \sqrt{T}). \quad (17)$$

- The denominator becomes:

$$n_i + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i) = n_i + O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T}). \quad (18)$$

2. **Rearranging for Simplicity:**

- The denominator inside the summation can be factored as:

$$1 + \frac{O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T})}{n_i}. \quad (19)$$

- Substituting this back into the utility loss bound:

$$\Delta S(t) \leq \frac{1}{c} \cdot \sum_{t_i \leq t} \frac{O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T})}{1 + \frac{O(\log(q) \cdot \sum_{k=1}^K M_k) + O(\sigma \cdot \sqrt{T})}{n_i}}. \quad (20)$$

3. **Simplification for Large  $n_i$ :**

- When  $n_i$  is large compared to the noise terms, the dominant contribution in the denominator is the constant 1:

$$1 + \frac{O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T})}{n_i} \approx 1. \quad (21)$$

- The summation simplifies to:

$$\Delta S(t) \leq \frac{1}{c} \cdot \sum_{t_i \leq t} \left( O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T}) \right). \quad (22)$$

#### 4. Final Simplified Form:

- Substituting the simplified summation, the final utility loss bound becomes:

$$\Delta S(t) \leq \frac{1}{c} \cdot \sum_{t_i \leq t} \left( O\left(\log(q) \cdot \sum_{k=1}^K M_k\right) + O(\sigma \cdot \sqrt{T}) \right). \quad (23)$$

□

**Theorem 3** (Convergence of Federated Kaplan-Meier Estimators). *Let  $\hat{S}_{\text{centralized}}(t)$  and  $\hat{S}_{\text{federated}}(t)$  denote the Kaplan-Meier survival probabilities at time  $t$  in centralized and federated homomorphic encrypted (HE) settings, respectively. Assume the noise introduced by aggregation and decryption,  $\varepsilon_{\text{aggregation}}$  and  $\varepsilon_{\text{decryption}}$ , approaches zero. Then:*

$$\lim_{\varepsilon_{\text{aggregation}}, \varepsilon_{\text{decryption}} \rightarrow 0} \hat{S}_{\text{federated}}(t) = \hat{S}_{\text{centralized}}(t). \quad (24)$$

*Proof.* 1. **Centralized Survival Probability:**

- The centralized Kaplan-Meier survival probability is given by:

$$\hat{S}_{\text{centralized}}(t) = \prod_{t_i \leq t} \left( 1 - \frac{d_i}{n_i} \right), \quad (25)$$

where  $d_i$  and  $n_i$  are the counts of events and at-risk counts at each event time  $t_i$ , respectively.

#### 2. Federated Survival Probability with Noise:

- The federated survival probability includes noise terms:

$$\hat{S}_{\text{federated}}(t) = \prod_{t_i \leq t} \left( 1 - \frac{d_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}{n_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)} \right), \quad (26)$$

where  $k$  denotes the client index.

#### 3. Convergence as Noise Approaches Zero:

- As  $\varepsilon_{\text{aggregation}}(t_i), \varepsilon_{\text{decryption}}(t_i) \rightarrow 0$ :

$$\frac{d_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)}{n_i^{(k)} + \varepsilon_{\text{aggregation}}(t_i) + \varepsilon_{\text{decryption}}(t_i)} \rightarrow \frac{d_i}{n_i}. \quad (27)$$

- Hence:

$$\lim_{\varepsilon_{\text{aggregation}}, \varepsilon_{\text{decryption}} \rightarrow 0} \hat{S}_{\text{federated}}(t) = \hat{S}_{\text{centralized}}(t). \quad (28)$$

□

**Theorem 4** (Scalability of Noise with Client Count). *Let  $\varepsilon_{\text{aggregation}}^{(k)}(t_i)$  and  $\varepsilon_{\text{decryption}}^{(k)}(t_i)$  represent the noise introduced during aggregation and decryption for client  $k$  at time  $t_i$  in a federated Kaplan-Meier setup with  $K$  clients and  $M_k$  ciphertexts per client. The total noise growth is given by:*

$$\varepsilon_{\text{total}} = \sum_{k=1}^K O(M_k \cdot \log(q_k)) + O(\sigma \cdot \sqrt{T}),$$

where  $q_k$  is the ciphertext modulus for client  $k$ ,  $\sigma$  is the standard deviation of decryption noise, and  $T \leq N$  is the subset of clients participating in decryption.

*Proof.* 1. **Aggregation Noise Growth:**

- Each client contributes  $M_k$  ciphertexts, and the server aggregates these over  $K$  clients.
- Noise from ciphertext addition grows linearly with the number of ciphertexts per client  $M_k$ :

$$\varepsilon_{\text{aggregation}}^{(k)}(t_i) = O(M_k \cdot \log(q_k)), \quad (29)$$

where  $\log(q_k)$  accounts for the scaling modulus in CKKS encryption for client  $k$ .

- The total aggregation noise from all clients is:

$$\varepsilon_{\text{aggregation}} = \sum_{k=1}^K O(M_k \cdot \log(q_k)). \quad (30)$$

2. **Decryption Noise Growth:**

- Each client contributes independently to the partial decryption process.
- Noise from combining partial decryptions grows with the square root of the number of clients involved in decryption ( $T$ ):

$$\varepsilon_{\text{decryption}} = O(\sigma \cdot \sqrt{T}), \quad (31)$$

where  $\sigma$  is the noise standard deviation per client and  $T \leq K$  is the number of decryption participants.

3. **Total Noise Growth:**

- Combining aggregation and decryption noise gives:

$$\varepsilon_{\text{total}} = \sum_{k=1}^K O(M_k \cdot \log(q_k)) + O(\sigma \cdot \sqrt{T}). \quad (32)$$

□

## 6 Experiments

### 6.1 Dataset Description

We used two datasets for our experiments: a) the NCCTG Lung Cancer dataset [3] [4], and b) the Synthetic Breast Cancer dataset [5].

The **NCCTG Lung Cancer dataset** comprises 228 observations across 10 variables. For our experiments, we focus primarily on two variables:

- **time**: A numeric variable representing survival time in days, defined as the number of days from the start of the study to either death or censoring.
- **status**: An integer variable indicating the censoring status: ‘1’ for censored observations and ‘2’ for death.

The **Synthetic Breast Cancer dataset**, released by the Netherlands Comprehensive Cancer Organisation (IKNL), contains 60,000 synthetically generated observations across 46 variables. For our analysis, we emphasize two key variables:

- **vit\_stat\_int**: A numeric variable representing survival time in days.
- **vit\_stat**: An integer variable indicating the status, where ‘0’ denotes censored observations and ‘1’ denotes death.

These datasets provide diverse structures and characteristics for evaluating Kaplan-Meier survival analysis methodologies under different conditions. Both these datasets are right censored.

### 6.2 Objectives of the Experiments

The main objectives of our experiments conducted in this study are to evaluate the performance, accuracy, and practicality of federated Kaplan-Meier survival analysis using multikey CKKS homomorphic encryption. These objectives are designed to comprehensively assess the proposed method’s capabilities while ensuring privacy preservation. The key objectives are as follows:

- **Visualization of Survival Curves**: Generate Kaplan-Meier survival curves using the federated setup with homomorphic encryption and compare them visually against those produced by a centralized or non-encrypted federated method.
- **Statistical Comparison of Survival Curves**: Conduct **log-rank tests** to evaluate whether there are statistically significant differences between the survival curves generated by the encrypted federated approach and those obtained using non-encrypted federated method.
- **Numerical Accuracy Assessment**: Measure the alignment between survival probabilities from the encrypted federated method and non-encrypted federated method through numerical metrics such as: a) Mean Absolute Error (MAE) which

- measures average deviation across all time points b) Maximum Absolute Error (MaxAE) which measures the largest deviation at any time point.
- **Reconstruction Attack Analysis:** Assess the ability of an adversary to infer sensitive information, such as the number of individuals at risk and the number of events (e.g., deaths). This analysis quantifies privacy risks by evaluating reconstruction accuracy in federated non-encrypted Kaplan-Meier survival curves across varying data distributions, numbers of providers, and degrees of data overlap between providers.
  - **Performance and Scalability Analysis:** Assess the computational efficiency of the encrypted federated survival analysis under various configurations. Test the scalability by varying the number of data providers and dataset sizes.

### 6.3 Experimental Setup

We repeated each experiment 10 times to facilitate clearer visualization of the resulting survival curves, to assess statistical and numerical accuracy between the federated encrypted and non-encrypted approaches, and to measure the associated computational overhead.

#### 6.3.1 Single machine setup

All experiments were performed on Macbook Pro with Apple M3 Pro, 36 GB of RAM, running Sonoma 14.5. All the code were implemented in Python 3.12.2.

#### 6.3.2 Homomorphic encryption parameters

To implemented our solution, we utilized the OpenFHE library [28]. The library is implemented in C++ and includes Python bindings, simplifying its integration into data analytics applications. Additionally, the library implements the thresholdization for BGV, BFV, and CKKS schemes. For these experiments, we use the thresholdization of the CKKS scheme, which supports computations over real numbers.

**Table 1:** Common Key Parameters (identical across all clients)

Parameter	Value	Description
<i>Scheme</i>	CKKS	The homomorphic encryption scheme.
<i>Security Level</i>	HEStd.128.classic	Targets 128-bit classical security.
<i>Standard Deviation</i>	3.2	Noise parameter $\sigma$ for key generation.
<i>Secret Key Distribution</i>	UNIFORM.TERNARY	Secret keys sampled from $\{-1, 0, +1\}$ uniformly.
<i>Multiplicative Depth</i>	3	Maximum number of homomorphic multiplications without bootstrapping.
<i>Key Switching Technique</i>	HYBRID	Key switching/relinearization method.
<i>Batch Size</i>	16	Number of slots for packing in CKKS.
<i>Scaling Mod Size</i>	50 bits	Bit-width of each ciphertext modulus "chunk."
<i>First Mod Size</i>	60 bits	Bit-width of the first prime in the modulus chain.
<i>Compression Level</i>	COMPRESSION_LEVEL.COMPACT	Affects how evaluation keys are compressed.
<i>Public Key (final)</i>	Shared among all clients	Result of multiparty key generation; distributed to each client.

Table 1 captures the homomorphic encryption parameters that are shared among all clients. These include the type of encryption scheme (CKKS), security level, noise

distribution parameters, the chosen key-switching technique, and so on. Critically, after the multiparty key generation process, there is a single, final public key that each client uses for encryption.

**Table 2:** Unique Key Parameters (per client)

Parameter	Description
<i>Secret Key Share</i>	Each client locally generates and retains its own secret key share. The shares never leave the client; they combine only logically for multiparty decryption.

Table 2 highlights the parameters that differ on a per-client basis. In particular, each client retains its own secret key share. These shares never leave the clients; they are only combined “logically” during the multiparty decryption phase (each client provides a partial decryption), thereby preserving security while enabling a joint decryption result. Table highlights the parameters that differ on a per-client basis. In particular, each client retains its own secret key share. These shares never leave the clients; they are only combined “logically” during the multiparty decryption phase (each client provides a partial decryption), thereby preserving security while enabling a joint decryption result.

### 6.3.3 Data Partitioning

In our experimental setup, each dataset is horizontally partitioned among multiple clients. We begin by randomly shuffling the entire dataset (using a fixed random seed for reproducibility) and then splitting the rows into a specified number of subsets—one subset per client. This approach ensures that each client receives a roughly equal number of records, but includes all of the original features (i.e., columns) for those records. Because the shuffle-and-split procedure draws each subset from the same overall distribution, it approximates an i.i.d. (independent and identically distributed) partition of the data.

### 6.3.4 Reconstruction Attack Analysis

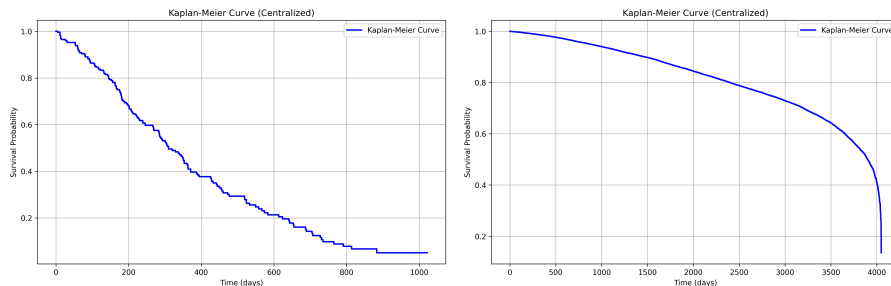
In this study, we simulate a federated scenario where multiple healthcare providers each hold a portion of patient-level time-to-event data. To capture different real-world possibilities, we split the dataset into subsets with varying levels of overlap:

- **No Overlap:** Providers have completely distinct patient records, modeling scenarios where each site covers a fully separate population.
- **Small Overlap:** A modest fraction of rows (e.g., 10%) is duplicated across each provider, reflecting occasional patient mobility or partial data sharing between sites.
- **Large Overlap:** A substantial fraction of rows (e.g., 50%) is shared among providers, approximating a highly integrated healthcare network in which multiple sites see the same patients. By examining these three overlap conditions (none, small, large), we effectively cover the spectrum of how patient data might be distributed across federated providers, allowing us to evaluate the impact of overlap on both survival analysis accuracy and privacy risks (i.e., reconstruction attacks).

## 7 Results

### 7.1 Visualization of Survival Curves

#### 7.1.1 Centralized Survival Curves



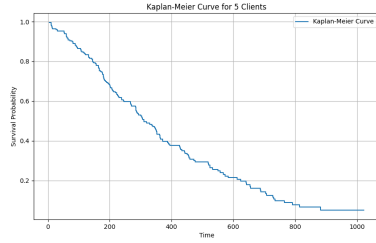
(a) Survival Curve for the lung cancer dataset without any federation and encryption (b) Survival Curve for the breast cancer dataset without any federation and encryption

**Fig. 1:** Centralized, Survival Curves for the Lung and Breast Cancer datasets

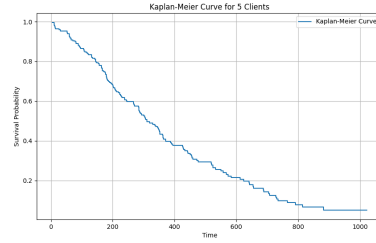
Figures 1a and 1b present the Kaplan–Meier survival curves for the lung and breast cancer datasets, respectively, under a centralized (non-federated, non-encrypted) setting. These plots serve as a baseline visualization, illustrating the underlying survival functions without any data partitioning or encryption processes involved. The survival curve for the breast cancer data is plotted using synthetic data; therefore, its shape might be a result of the synthetic data generation process not accurately reflecting the real data distribution.

#### 7.1.2 Federated and Federated-Encrypted Survival Curves

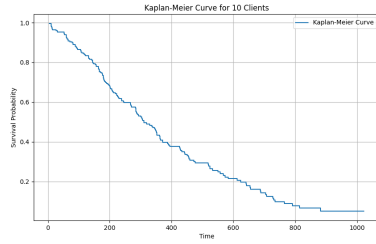
Figures 2 and 3 present the Kaplan–Meier survival curves for the lung cancer and breast cancer datasets, respectively, under both federated and federated-encrypted scenarios at various client counts. Each subfigure compares the survival estimation obtained when the data is split among multiple clients (federation) against the estimation obtained when the same federated process is combined with encryption. These visualizations offer a direct comparison, allowing us to observe whether encryption and distribution of data across multiple clients substantially alter the estimated survival curves relative to one another. In practice, if the curves remain visually similar across the federated and federated-encrypted approaches, it suggests that the federated computation and associated encryption methodologies do not materially change the underlying survival estimates. Specifically, for both datasets, as the number of clients increases (5, 10, 20, 50), the shape and position of the survival curves remain consistent between the federated and federated-encrypted implementations. This indicates that adding more clients to the federated environment does not introduce additional discrepancies or distortions attributable to encryption. In essence, these figures and the corresponding observations provide strong evidence that federated survival analysis, whether encrypted or not, yields results comparable to the centralized scenario.



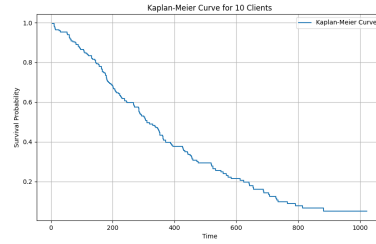
(a) Federated, 5 Clients



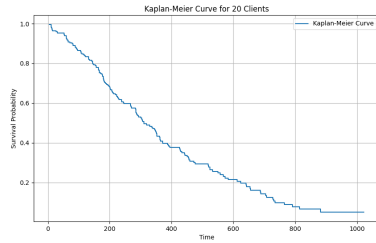
(b) Federated Encrypted, 5 Clients



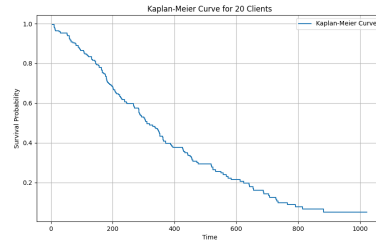
(c) Federated, 10 Clients



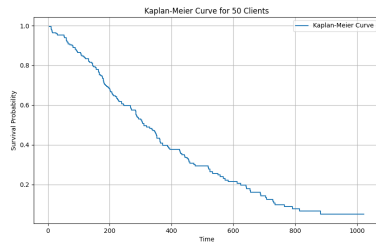
(d) Federated Encrypted, 10 Clients



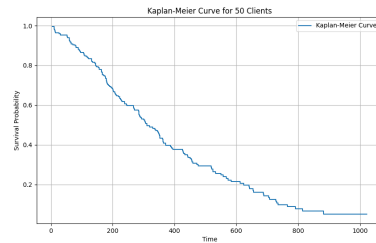
(e) Federated, 20 Clients



(f) Federated Encrypted, 20 Clients



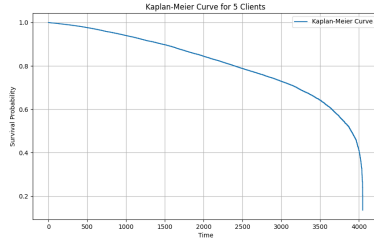
(g) Federated, 50 Clients



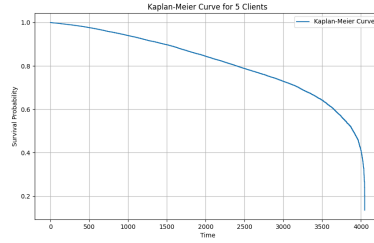
(h) Federated Encrypted, 50 Clients

**Fig. 2:** Federated and Federated Encrypted Survival Curves for the Lung Cancer dataset at Varying Client Counts

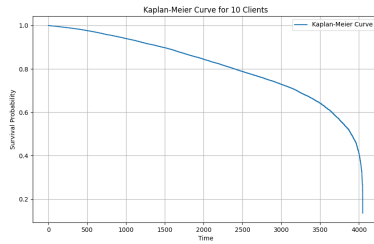




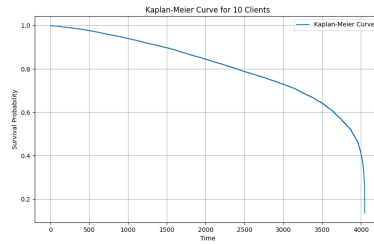
(a) Federated, 5 Clients



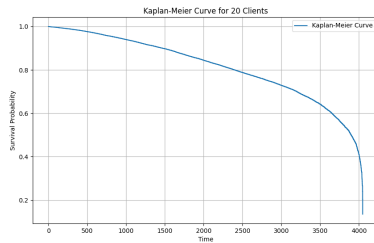
(b) Federated Encrypted, 5 Clients



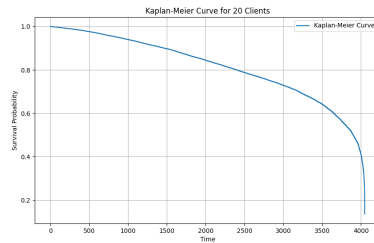
(c) Federated, 10 Clients



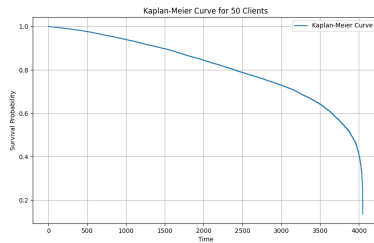
(d) Federated Encrypted, 10 Clients



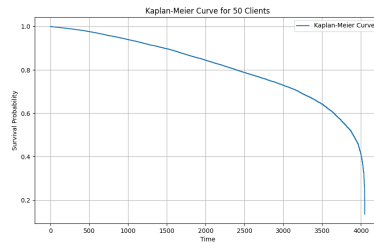
(e) Federated, 20 Clients



(f) Federated Encrypted, 20 Clients



(g) Federated, 50 Clients



(h) Federated Encrypted, 50 Clients

**Fig. 3:** Federated and Federated Encrypted Survival Curves for the Breast Cancer dataset at Varying Client Counts

## 7.2 Statistical Comparison of Survival Curves

We have conducted log-rank tests to do statistical comparisons between the federated encrypted and federated non-encrypted survival curves. The log-rank test is a non-parametric statistical test commonly used to compare the survival functions of two or more groups. It assesses whether the observed differences in survival times between these groups could be due to random variation or if they are statistically significant.

### Null Hypothesis

For this analysis, the null hypothesis ( $H_0$ ) is:

$H_0$  : **There is no difference in the survival distributions between the federated-encrypted and the federated non-encrypted groups.**

If the test provides sufficient evidence against  $H_0$ , we would conclude that there is a statistically significant difference in the survival distributions. If not, we fail to reject  $H_0$  and conclude that any observed differences are likely due to chance.

### P-Value

The P-value represents the probability of obtaining the observed results, or more extreme, assuming the null hypothesis is true. A low P-value (typically less than 0.05) suggests that the observed differences are unlikely to be due to random chance, thus providing evidence against  $H_0$ . Conversely, a high P-value suggests that the observed differences could easily arise from random variation, providing little reason to reject  $H_0$ .

### 7.2.1 Log Rank Test Results and P-Values for the Lung Cancer Dataset

Clients	Test Statistic Mean	Test Statistic CI	P-Value Mean	P-Value CI
2	0.0057	[0.0005, 0.0109]	9.5406e-01	[9.3017e-01, 9.7795e-01]
5	0.0138	[0.0057, 0.0220]	9.2111e-01	[8.9006e-01, 9.5217e-01]
10	0.0164	[0.0096, 0.0232]	9.0389e-01	[8.8303e-01, 9.2474e-01]
20	0.0183	[0.0077, 0.0289]	9.0897e-01	[8.7375e-01, 9.4420e-01]
30	0.0160	[0.0050, 0.0271]	9.1882e-01	[8.8212e-01, 9.5553e-01]
40	0.0169	[0.0057, 0.0280]	9.1589e-01	[8.7905e-01, 9.5274e-01]
50	0.0056	[-0.0013, 0.0126]	9.6287e-01	[9.3407e-01, 9.9168e-01]

**Table 3:** Summary of Test Statistics and P-Values for comparing federated encrypted and federated non-encrypted Kaplan-Meier estimators for the lung cancer dataset.

Based on the results available in the Table 3, the following can be interpreted:

- **Test Statistics:** The test statistics across various client counts (2, 5, 10, 20, 30, 40, 50) are small (ranging roughly from 0.0056 to 0.0183). Such small test statistics indicate minimal difference between the two survival curves.
- **Confidence Intervals for Test Statistics:** The confidence intervals (e.g., [0.0005, 0.0109] for 2 clients or [0.0077, 0.0289] for 20 clients) are tight and centered around small values. This narrow range indicates little uncertainty, suggesting that it is unlikely there are large, unobserved differences in the underlying survival distributions. The negative lower confidence limit (-0.0013) for 50 clients

is due to the reflection of uncertainty in the estimate rather than an indication of something inherently incorrect. Confidence intervals for a statistical measure represent a range of plausible values based on the sample data. If the true effect size (difference in survival distributions) is very close to zero, the confidence interval may span both slightly positive and slightly negative values.

- **P-Values:** The P-values are consistently high (close to or above 0.9), which strongly suggests that any observed differences in survival are plausible under the null hypothesis. In statistical terms, these high P-values provide no compelling evidence to reject  $H_0$ . Instead, they indicate that the observed variations in survival could easily be explained by random chance.
- **Confidence Intervals for P-Values:** The P-value confidence intervals are also tight and remain consistently high. This reinforces the stability of the estimation, further solidifying the conclusion that there is no statistically significant difference between the two groups.

### 7.2.2 Log Rank Test Results and P-Values for the Breast Cancer Dataset

Clients	Test Statistic Mean	Test Statistic CI	P Value Mean	P Value CI
2	0.0004	[0.0001, 0.0006]	9.8768e-01	[9.8198e-01, 9.9338e-01]
5	0.0004	[0.0002, 0.0005]	9.8674e-01	[9.8233e-01, 9.9116e-01]
10	0.0005	[0.0002, 0.0007]	9.8435e-01	[9.8030e-01, 9.8840e-01]
20	0.0005	[0.0002, 0.0007]	9.8444e-01	[9.7950e-01, 9.8937e-01]
30	0.0003	[0.0001, 0.0006]	9.8808e-01	[9.8266e-01, 9.9350e-01]
40	0.0004	[0.0002, 0.0007]	9.8607e-01	[9.8044e-01, 9.9170e-01]
50	0.0004	[0.0002, 0.0007]	9.8599e-01	[9.8101e-01, 9.9096e-01]

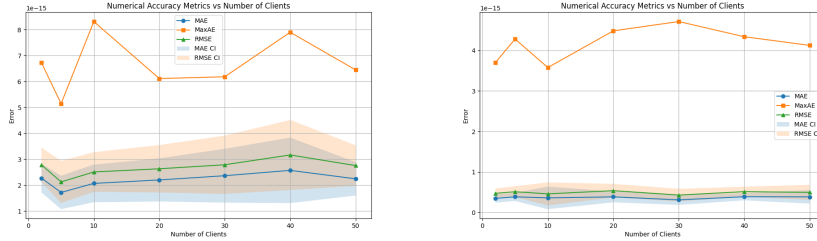
**Table 4:** Summary of Test Statistics and P-Values for comparing federated encrypted and federated non-encrypted Kaplan–Meier estimators for the breast cancer dataset.

Based on the results available in the Table 4, the following is interpreted:

- **Test Statistic:** The test statistic values are consistently very small (on the order of 0.0003 to 0.0005). Such small values indicate that there are effectively no discernible differences in the estimated survival distributions between the federated-encrypted and federated non-encrypted scenarios.
- **Confidence Intervals for the Test Statistic:** The confidence intervals for the test statistic are extremely narrow, and they do not deviate significantly from zero. This suggests a high degree of certainty that the true difference in the survival curves is close to zero.
- **P-Values:** The P-values are consistently close to 1. High P-values imply that the observed differences, if any, are easily attributable to random variation rather than a meaningful effect of encryption on the survival analysis. In other words, there is no statistical evidence to reject the null hypothesis of no difference.
- **P-Value Confidence Intervals:** The P-value confidence intervals remain tight and centered near values just under 1, reinforcing the conclusion that no statistically significant difference exists.

In summary, no statistically significant differences were observed between the federated non-encrypted and federated-encrypted survival curves for either dataset.

### 7.3 Numerical Accuracy Assessments



(a) Numerical accuracy between the fed-encrypted and federated non-encrypted survival curves for the lung cancer dataset. (b) Numerical accuracy between the fed-encrypted and federated non-encrypted survival curves for the breast cancer dataset.

**Fig. 4:** Numerical Accuracy Results for the Lung and Breast Cancer datasets

The interpretations of the numerical accuracy results obtained for both the lung cancer and breast cancer datasets under federated and federated-encrypted scenarios are the following:

- **Consistency Across Datasets:** For both the lung and breast cancer datasets, the federated-encrypted approach yields survival estimates that remain closely aligned with those of the federated non-encrypted baseline. Regardless of whether the dataset consists of real-world (lung) or synthetic data (breast), the core finding is that encryption does not introduce meaningful distortions into the estimated survival curves.
- **Stability of MAE and RMSE:** The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) remain relatively low and stable as the number of clients increases. While there may be minor fluctuations or incremental increases as the federation expands to more clients, these changes are modest. This suggests that the accuracy of the survival estimates is robust to scaling, with no significant degradation when more institutions contribute data.
- **MaxAE Variability:** Although the Maximum Absolute Error (MaxAE) occasionally exhibits peaks at certain client counts, these represent isolated instances rather than a pervasive issue. Across both datasets, the vast majority of time points maintain small deviations between encrypted and non-encrypted federated scenarios, and large outliers are not frequent enough to undermine the overall accuracy.
- **Confidence Intervals and Reliability:** The confidence intervals for MAE and RMSE remain relatively tight for both datasets, reinforcing the reliability of

the estimates. As the number of clients grows, slight broadening of these intervals occurs, but not to a degree that would indicate substantial uncertainty or instability.

Across both the lung cancer and breast cancer datasets, the federated-encrypted survival analysis demonstrates a high degree of numerical accuracy and stability, closely matching the federated non-encrypted approach. Increases in the number of clients do not substantially compromise accuracy, and while occasional larger errors occur, they do not impact the overall integrity of the results. This indicates that employing encryption within a federated setting is feasible and does not significantly affect the fidelity of the resulting survival estimates.

## 7.4 Reconstruction Attack Analysis

### 7.4.1 Reconstruction Risk Scenario

In a federated survival analysis setting, multiple healthcare providers (e.g., hospitals or clinics) each hold a portion of patient-level time-to-event data. Rather than pooling their data directly, the providers collaborate by sending aggregated statistics to a central server, which computes the overall survival function via the Kaplan–Meier estimator. Under a non-encrypted approach, each provider transmits their counts of patients at risk ( $n_i$ ) and the number of observed events ( $d_i$ ) at each time point  $t_i$ .

#### *Federated Computation.*

Each data provider computes local Kaplan–Meier statistics for its subset of the data, yielding time-indexed counts of how many patients remain at risk and how many events occur at each time point.

#### *Aggregation at the Server.*

The server receives these unencrypted local counts from all participating providers. By summing the individual counts, the server forms federated aggregated statistics, representing the entire federated dataset.

#### *Attacker’s Knowledge.*

Assume one of the providers (the “attacker”) aims to infer sensitive details about patients at other institutions. This attacker knows:

- Their own local counts,  $(n_i^{(A)}, d_i^{(A)})$ .
- The aggregated counts across all providers,  $(n_i^{(\text{Fed})}, d_i^{(\text{Fed})})$ .

#### *Reconstruction Attack.*

By subtracting their own known local counts from the federated totals, the attacker infers the combined contributions of all other providers:

$$n_i^{(\text{Others})} = n_i^{(\text{Fed})} - n_i^{(A)}, \quad d_i^{(\text{Others})} = d_i^{(\text{Fed})} - d_i^{(A)}.$$

In scenarios where some patient data overlaps between providers (e.g., multiple sites containing records on the same individuals), this subtraction-based approach can

become even more revealing, as the attacker may indirectly learn overlapping patient information if the aggregated totals reflect shared data points.

***Evaluating Reconstruction Success.***

To quantify how accurately the attacker recovers the other providers’ data, we compare the inferred counts,  $\hat{x}_i$ , with the true counts,  $x_i$ , across each relevant time point  $i$ . Various metrics can be computed for both at-risk counts ( $n_i$ ) and event counts ( $d_i$ ), including:

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{x}_i - x_i|.$$

A smaller MAE implies higher risk, as it indicates the attacker’s estimates closely match real data. Conversely, a large MAE suggests lower risk, reflecting poorer reconstruction accuracy.

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2}.$$

RMSE penalizes large deviations more than MAE, thus indicating whether outliers in the attacker’s estimates significantly degrade the reconstruction.

- **Coefficient of Determination ( $R^2$ ):**

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{x}_i - x_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad \bar{x} = \text{mean of } x_i.$$

Values close to 1 signify that the attacker’s inferences capture most of the variation in the real data, indicating a strong reconstruction.

- **0–1 Accuracy Metric:** A normalized metric is used, such as

$$\text{Accuracy} = 1 - \frac{\text{MAE}}{\text{Scale}},$$

where Scale can be the maximum or range of the true values. Higher accuracy values (closer to 1) represent more precise reconstruction and therefore greater privacy risk.

In the following figures (Figures 5 and 6), we present six reconstruction metrics for two datasets: **NCCTG Lung Cancer** and **Synthetic Breast Cancer**. We compare three overlap scenarios (*None Overlap*, *Small Overlap*, and *Large Overlap*) across increasing numbers of providers (2 to 50). We consider two key Kaplan–Meier quantities:

- $n_{\text{at.risk}}$ : The number of patients at risk.

- $d_t$ : The number of events (e.g., deaths).

For each of these, three metrics are assessed:

- **Reconstruction Accuracy** (0–1), derived from normalized MAE.
- $R^2$  (Coefficient of Determination), indicating how closely inferred values match the ground truth.
- **RMSE** (Root Mean Squared Error), capturing the average magnitude of the attacker’s errors.

### 1. *Reconstruction Accuracy (0–1)*

- **Few Providers (2–3)**. Large overlap yields near-perfect accuracy in both datasets when only one other site is present. The attacker’s knowledge heavily overlaps with that single remaining provider, making subtraction-based inference almost exact.
- **More Providers (5–50)**. Accuracy rapidly falls toward zero for all overlap settings. As additional sites join, the attacker’s ability to isolate each provider’s data diminishes, especially under large overlap (where many providers share the same records, further confusing the attack).

### 2. *Coefficient of Determination ( $R^2$ )*

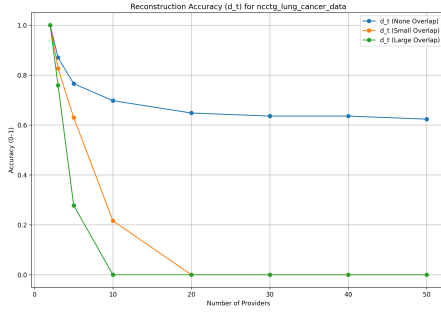
- In both datasets,  $R^2$  frequently becomes negative for provider counts above 5 or 10, indicating reconstruction is worse than a naive “predict the mean” baseline.
- Large overlap can start near zero or slightly positive for very few sites but plunges into large negative values once the federation grows, reflecting how multiple shared records confound a simple subtraction approach.

### 3. *Root Mean Squared Error (RMSE)*

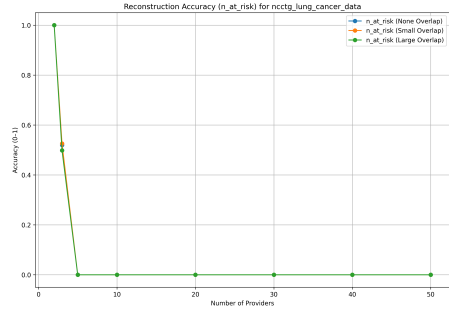
- **Lung Cancer**. Under no overlap, RMSE remains comparatively low, implying the attacker’s estimates are (ironically) more accurate than in large overlap when many sites are present. Large overlap grows steeply with more providers, showing the attack’s failure on multi-site shared data.
- **Breast Cancer**. A similar pattern occurs: large overlap has minimal RMSE for 2 providers but escalates quickly to high values by 50 providers, while no overlap retains the lowest RMSE across many providers.

+

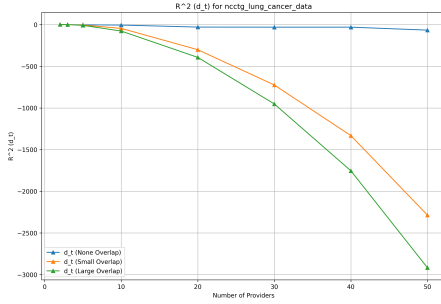
## 7.4.2 NCCTG Lung Cancer Dataset Results



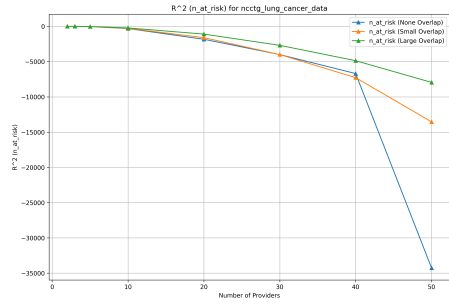
(a) Reconstruction Accuracy ( $d_t$ )



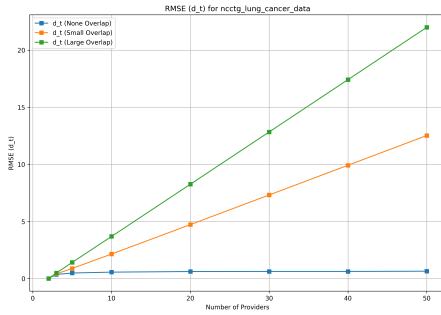
(b) Reconstruction Accuracy ( $n_{at\_risk}$ )



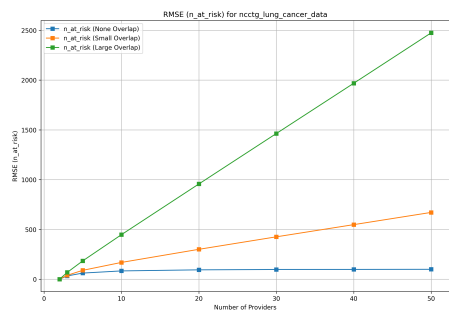
(c)  $R^2$  ( $d_t$ )



(d)  $R^2$  ( $n_{at\_risk}$ )



(e) RMSE ( $d_t$ )

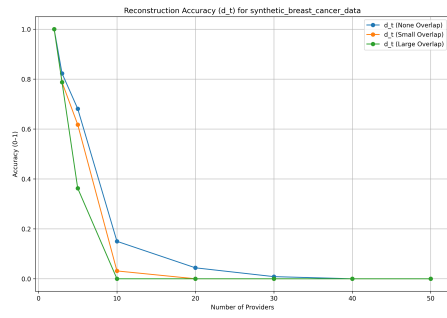


(f) RMSE ( $n_{at\_risk}$ )

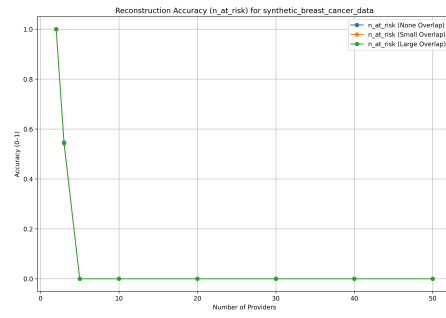
**Fig. 5:** NCCTG Lung Cancer Dataset: Reconstruction metrics comparing *None*, *Small*, and *Large* Overlap for  $d_t$  and  $n_{at\_risk}$ . The x-axis is the number of providers, and each curve shows how an attacker's reconstruction quality changes with the federation size.



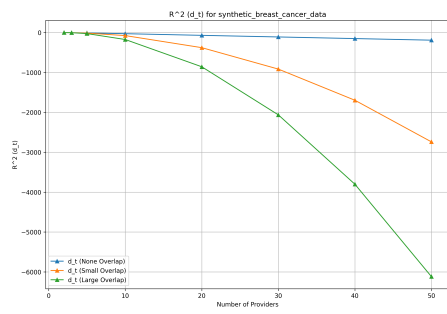
### 7.4.3 Synthetic Breast Cancer Dataset Results



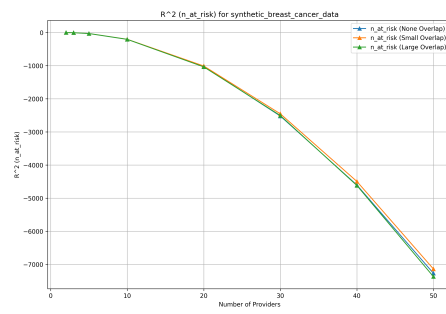
(a) Reconstruction Accuracy ( $d_t$ )



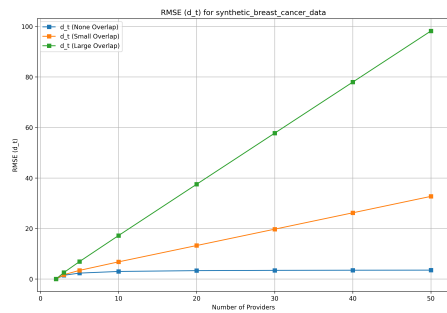
(b) Reconstruction Accuracy ( $n_{at\_risk}$ )



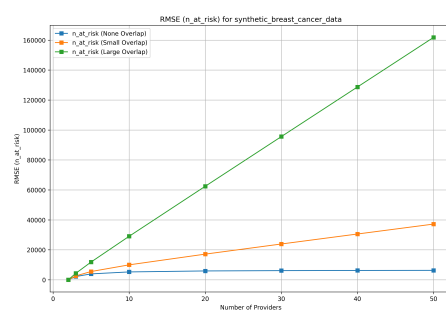
(c)  $R^2(d_t)$



(d)  $R^2(n_{at\_risk})$



(e) RMSE ( $d_t$ )



(f) RMSE ( $n_{at\_risk}$ )

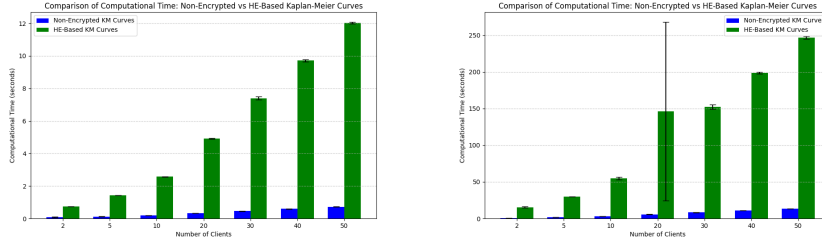
**Fig. 6:** Synthetic Breast Cancer Dataset: Reconstruction metrics for *None*, *Small*, and *Large* Overlap using  $d_t$  and  $n_{at\_risk}$ .

### Overall Findings

1. **Small Federations (2–3 Providers).** Large overlap plus only one or two other sites leads to near-complete reconstruction, representing a severe privacy risk.
2. **Larger Federations (5–10+).** Accuracy converges near zero, while RMSE climbs;  $R^2$  goes highly negative. The attacker’s simple subtraction strategy breaks down as more providers contribute data.
3. **Overlap Impact.** Large overlap is devastating for privacy only when the federation is small. With many providers, large overlap ironically confuses the attacker more, driving RMSE upward and reconstruction quality downward.
4. **Both Datasets.** Despite differences in scale (e.g., the breast cancer dataset’s larger RMSE), the overall pattern holds: high risk in small, high-overlap settings; diminished attacker performance in larger federations.

In essence, unencrypted federated survival analysis is most vulnerable when few providers participate and they share a substantial fraction of the same patients. As the federation grows and/or overlap decreases, the attack’s efficacy rapidly collapses, pushing both  $R^2$  and accuracy toward negligible values while inflating RMSE.

### 7.5 Performance and Scalability Analysis



(a) federated non-encrypted vs federated encrypted, lung cancer dataset (b) federated non-encrypted vs federated encrypted, breast cancer dataset

**Fig. 7:** Comparison of computational times between federated non-encrypted and federated encrypted Kaplan-Meier estimators across varying numbers of clients.

Figure 7 compares the computational times required for non-encrypted and homomorphically encrypted (HE) federated Kaplan–Meier (KM) estimations across varying numbers of clients (from 2 to 50) for both the **NCCTG Lung Cancer** and **Synthetic Breast Cancer** datasets.

In case of NCCTG Lung Cancer Dataset, for smaller client counts, non-encrypted computation is extremely fast, taking only about 0.09s at 2 clients and rising to around 0.73s by 50 clients. In contrast, homomorphically encrypted (HE) times range from roughly 0.74s at 2 clients to nearly 12.0s at 50 clients, lower in absolute terms than the breast cancer dataset, yet still reflecting a notable overhead when compared to the non-encrypted baseline. Specifically, the ratio of HE to non-encrypted computation is about  $8.4\times$  at 2 clients and increases steadily, reaching around  $16.6\times$  by 50 clients.

Although this ratio is somewhat less extreme than the 18–19 $\times$  range observed for the breast cancer dataset, it clearly remains a substantial cost attributable to encryption.

When it comes to the synthetic breast cancer dataset, in the non-encrypted scenario, computation time increases steadily from approximately 0.8s at 2 clients to around 13.4s at 50 clients. By contrast, the homomorphically encrypted (HE) version starts at about 15.5s for 2 clients and reaches nearly 247s (over 4 minutes) by 50 clients. Consequently, the ratio of HE to non-encrypted times typically hovers around 18–19 $\times$ , although at 20 clients there is an outlier spike to approximately 25.7 $\times$ , potentially due to resource or key-switching overhead triggered at this scale.

Overall, these results confirm a clear trend: homomorphic encryption adds a significant overhead to federated KM analysis compared to non-encrypted computation. The exact slowdown factor depends on both: a) the dataset’s baseline (non-encrypted) cost (computational time) and b) the number of clients (scale of the federation).

In these experiments, latency estimates caused by the federated approach were excluded from the evaluation. Including latency would not alter the comparative results between the encrypted and non-encrypted solutions. However, it would introduce higher delays when compared to a centralized (non-federated) solution.

## 8 Discussion

Our proposed framework substantially extends existing methods for privacy-preserving federated Kaplan–Meier analysis [1, 2]. By leveraging CKKS encryption, we enable precise floating-point operations—facilitating realistic time-to-event modeling—while limiting the privacy risks that arise in multi-institutional collaborations.

One of the notable contributions is a detailed analysis of *reconstruction attacks*, wherein a malicious institution can subtract its local counts from the global aggregated counts to infer other institutions’ data. Prior efforts have often acknowledged the feasibility of federated approaches but provided only partial insight into how data overlaps among institutions can amplify privacy threats. In contrast, our work systematically examines how different federation sizes and degrees of patient overlap affect an attacker’s reconstruction accuracy. We find that, in *small federations* (e.g., 2–3 providers) with large data overlap, an adversary can nearly replicate at-risk and event counts for other sites, posing a serious confidentiality risk. In particular, when only two providers are involved, we demonstrate that an attacker can completely reconstruct the opposing site’s data. Homomorphic encryption proves especially critical under these conditions, as it effectively neutralizes the subtraction-based inference vector.

When the federation grows beyond 5–10 providers, we observe that the attacker’s reconstruction accuracy falls sharply, with negative  $R^2$  values and increasing RMSE. In larger federations, overlapping patient records across numerous sites dilute the effectiveness of simple subtraction attacks, rendering near-complete reconstruction far less likely. From a practical standpoint, these findings suggest that *homomorphic encryption is indispensable* in *small-scale federations*, particularly if high overlap exists among participants. As the number of institutions increases, the inherent difficulty of reconstruction diminishes; however, encryption remains advisable when stringent privacy requirements or regulations necessitate maximum data protection, regardless of natural obfuscation due to federation size.

Although homomorphic encryption incurs additional computational overhead, commonly up to an order of magnitude beyond non-encrypted baselines. Our empirical assessments on both the NCCTG Lung Cancer and Synthetic Breast Cancer datasets confirm its feasibility up to 50 clients. The capability for exact floating-point arithmetic also ensures that homomorphically computed Kaplan–Meier curves and log-rank statistics are virtually indistinguishable from centralized or non-encrypted federated results. This balance of privacy protection and analytic accuracy marks a significant step forward over previous frameworks constrained by integer-only homomorphic schemes [1].

Beyond the empirical evidence, we established a comprehensive theoretical model capturing both *aggregation noise* (introduced when ciphertexts from multiple clients are summed or multiplied) and *decryption noise* (arising during the threshold decryption process). Formally, our utility-loss bounds show that the difference between the encrypted federated estimates and the centralized (unencrypted) estimates remains small, proportional to the total number of ciphertext operations and the associated noise parameters. Moreover, we prove that as these noise parameters trend to negligible levels (e.g., through appropriate choice of ciphertext modulus or batching strategies), the federated Kaplan–Meier estimator *converges* to the centralized estimator. These findings provide strong theoretical reassurance that neither the addition of encryption layers nor the threshold-based decryption process sacrifices statistical validity in the long run.

However, despite these advancements, several challenges warrant further research. First, our experimental settings emphasize i.i.d. partitioning; real-world federations often involve diverse populations and non-i.i.d. data distributions, necessitating more sophisticated partition strategies or adaptive encryption parameters. Second, while homomorphic encryption substantially mitigates privacy risks, it amplifies computational overhead, which may become prohibitive in extremely large consortia. Future work could explore more efficient key-switching and partial plaintext tricks or integrate other privacy mechanisms (e.g., differential privacy [33, 34]) alongside encryption.

Further, there exists a small potential risk of key leakage and that private key shares could be reconstructed by curious parties after a large number of decryption rounds. It is therefore crucial to implement robust measures that prevent key leakage and maintain the security lifecycle of secret-key shares. One of the most effective strategies for achieving these objectives is the implementation of key rotation / revocation routines. Next, our current results focus on the standard Kaplan–Meier test; expanding this framework to left and interval censored datasets and more advanced survival models (e.g., Cox proportional hazards) is essential for broader clinical applicability. Moreover, real-world deployment requires tackling practical hurdles such as network latency, institutional governance, and secure key distribution in production environments. Additionally, beyond the reconstruction attacks analyzed herein, new variants of inference or side-channel attacks can also be tested to further validate and harden the proposed framework against evolving threat landscapes.

In the evaluation presented in this work, the parties operate on datasets of equal size for each iteration of the federated Kaplan–Meier estimation. In practical settings, this requires either an underlying negotiation protocol, which enables parties to agree

on the size of the data subset before each iteration, or an implementation of a weighted aggregation. However, negotiation rounds introduce additional communication overhead, thereby increasing latency. Furthermore, institutions with more data will need to retain extra data for subsequent rounds, while those with less data will have to wait until the required amount is acquired, thereby delaying the overall process. Finally, they expose more information about the data sources, consequently increasing the risks of reconstruction attacks. On the other hand, implementing weighted aggregation requires the use of homomorphic computations, which include multiplicative operations and the application of inverse functions. This approach negatively affects both the precision of the outcomes as well as the computation time. Further research is required to define the conditions under which each of these approaches might be preferable.

## 9 Conclusion

By combining threshold-based CKKS encryption, explicit reconstruction-attack analysis, and rigorous theoretical bounds on noise and convergence, this work advances the state-of-the-art work privacy and scalability of federated survival analysis. The empirical findings highlights how homomorphic encryption is especially vital in small, high-overlap federations, while remaining beneficial for larger-scale collaborations seeking robust privacy guarantees. Through continued investigation into more complex survival models, non-i.i.d. data distributions, and operational integration, our approach can further advance multi-institutional research without compromising patient confidentiality.

## References

- [1] Froelicher, D., Troncoso-Pastoriza, J.R., Raisaro, J.L., Cuendet, M.A., Sousa, J.S., Cho, H., Berger, B., Fellay, J., Hubaux, J.-P.: Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature Communications* **12**(1), 5910 (2021) <https://doi.org/10.1038/s41467-021-25972-y>
- [2] Geva, R., Gusev, A., Polyakov, Y., Liram, L., Rosolio, O., Alexandru, A., Genise, N., Blatt, M., Duchin, Z., Waissengrin, B., *et al.*: Collaborative privacy-preserving analysis of oncological data using multiparty homomorphic encryption. *Proceedings of the National Academy of Sciences* **120**(33), 2304415120 (2023)
- [3] Therneau, T., Atkinson, E., Crowson, C.: Lung Cancer Data in the Survival Package. (2024). Accessed: 2024-12-02. <https://rdrr.io/cran/survival/man/lung.html>
- [4] Loprinzi, C.L., Laurie, J.A., Wieand, H.S., Krook, J.E., Novotny, P.J., Kugler, J.W., Bartel, J., Law, M., Bateman, M., Klatt, N.E.: Prospective evaluation of prognostic variables from patient-completed questionnaires. north central cancer treatment group. *Journal of Clinical Oncology* **12**(3), 601–607 (1994)
- [5] (IKNL), N.C.C.O.: Netherlands Cancer Registry (NCR). <https://iknl.nl/en/ncr>. Accessed: 2024-12-12 (2024)
- [6] Masciocchi, C., Gottardelli, B., Savino, M., Boldrini, L., Martino, A., Mazzarella, C., Massaccesi, M., Valentini, V., Damiani, A.: Federated cox proportional hazards model with multicentric privacy-preserving lasso feature selection for survival

- analysis from the perspective of personalized medicine. In: 2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS), pp. 25–31 (2022). <https://doi.org/10.1109/CBMS55023.2022.00012>
- [7] Archetti, A., Ieva, F., Matteucci, M.: Scaling survival analysis in healthcare with federated survival forests: A comparative study on heart failure and breast cancer genomics. *Future Generation Computer Systems* **149**, 343–358 (2023) <https://doi.org/10.1016/j.future.2023.07.036>
- [8] Andreux, M., Manoel, A., Menuet, R., Saillard, C., Simpson, C.: Federated Survival Analysis with Discrete-Time Cox Models (2020). <https://arxiv.org/abs/2006.08997>
- [9] Imakura, A., Tsunoda, R., Kagawa, R., Yamagata, K., Sakurai, T.: Dc-cox: Data collaboration cox proportional hazards model for privacy-preserving survival analysis on multiple parties. *Journal of Biomedical Informatics* **137**, 104264 (2023) <https://doi.org/10.1016/j.jbi.2022.104264>
- [10] Pan, Y., Chao, Z., He, W., Jing, Y., Hongjia, L., Liming, W.: Fedshe: privacy preserving and efficient federated learning with adaptive segmented ckks homomorphic encryption. *Cybersecurity* **7**(1), 40 (2024) <https://doi.org/10.1186/s42400-024-00232-w>
- [11] Madi, A., Stan, O., Mayoue, A., Grivet-Sébert, A., Gouy-Pailler, C., Sirdey, R.: A secure federated learning framework using homomorphic encryption and verifiable computing. In: 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS), pp. 1–8 (2021). <https://doi.org/10.1109/RDAAPS48126.2021.9452005>
- [12] Fang, H., Qian, Q.: Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet* **13**(4) (2021) <https://doi.org/10.3390/fi13040094>
- [13] Jin, W., Yao, Y., Han, S., Joe-Wong, C., Ravi, S., Avestimehr, A.S., He, C.: Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system. *ArXiv abs/2303.10837* (2023)
- [14] Blatt, M., Gusev, A., Polyakov, Y., Goldwasser, S.: Secure large-scale genome-wide association studies using homomorphic encryption. *Cryptology ePrint Archive*, Paper 2020/563 (2020). <https://doi.org/10.1073/pnas.1918257117> . <https://eprint.iacr.org/2020/563>
- [15] Sarkar, E., Chielle, E., Gursoy, G., Chen, L., Gerstein, M., Maniatakos, M.: Privacy-preserving cancer type prediction with homomorphic encryption. *Scientific Reports* **13**(1), 1661 (2023) <https://doi.org/10.1038/s41598-023-28481-8>
- [16] Truhn, D., Tayebi Arasteh, S., Saldanha, O.L., Müller-Franzes, G., Khader, F., Quirke, P., West, N.P., Gray, R., Hutchins, G.G.A., James, J.A., Loughrey, M.B., Salto-Tellez, M., Brenner, H., Brobeil, A., Yuan, T., Chang-Claude, J., Hoffmeister, M., Foersch, S., Han, T., Keil, S., Schulze-Hagen, M., Isfort, P., Bruners, P., Kaissis, G., Kuhl, C., Nebelung, S., Kather, J.N.: Encrypted federated learning for secure decentralized collaboration in cancer image analysis. *Medical Image Analysis* **92**, 103059 (2024) <https://doi.org/10.1016/j.media.2023.103059>
- [17] Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. *Foundations of Secure Computation*, Academia Press, 169–179

- (1978)
- [18] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. STOC '09, pp. 169–178. Association for Computing Machinery, New York, NY, USA (2009). <https://doi.org/10.1145/1536414.1536440> . <https://doi.org/10.1145/1536414.1536440>
  - [19] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. ITCS '12, pp. 309–325. Association for Computing Machinery, New York, NY, USA (2012). <https://doi.org/10.1145/2090236.2090262> . <https://doi.org/10.1145/2090236.2090262>
  - [20] Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical gapsvp. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012, pp. 868–886. Springer, Berlin, Heidelberg (2012)
  - [21] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016, pp. 3–33. Springer, Berlin, Heidelberg (2016)
  - [22] Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013, pp. 75–92. Springer, Berlin, Heidelberg (2013)
  - [23] Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold fhe. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012, pp. 483–501. Springer, Berlin, Heidelberg (2012)
  - [24] Boneh, D., Gennaro, R., Goldfeder, S., Jain, A., Kim, S., Rasmussen, P.M.R., Sahai, A.: Threshold cryptosystems from threshold fully homomorphic encryption. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, pp. 565–596. Springer, Cham (2018)
  - [25] Desmedt, Y.: In: Tilborg, H.C.A., Jajodia, S. (eds.) Threshold Cryptography, pp. 1288–1293. Springer, Boston, MA (2011). [https://doi.org/10.1007/978-1-4419-5906-5\\_330](https://doi.org/10.1007/978-1-4419-5906-5_330) . [https://doi.org/10.1007/978-1-4419-5906-5\\_330](https://doi.org/10.1007/978-1-4419-5906-5_330)
  - [26] Schoenmakers, B.: In: Tilborg, H.C.A., Jajodia, S. (eds.) Threshold Homomorphic Cryptosystems, pp. 1293–1294. Springer, Boston, MA (2011). [https://doi.org/10.1007/978-1-4419-5906-5\\_13](https://doi.org/10.1007/978-1-4419-5906-5_13) . [https://doi.org/10.1007/978-1-4419-5906-5\\_13](https://doi.org/10.1007/978-1-4419-5906-5_13)
  - [27] Bendlin, R., Damgård, I.: Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In: Micciancio, D. (ed.) Theory of Cryptography, pp. 201–218. Springer, Berlin, Heidelberg (2010)
  - [28] Badawi, A.A., Alexandru, A., Bates, J., Bergamaschi, F., Cousins, D.B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., Liu, Z., Micciancio, D., Pascoe, C., Polyakov, Y., Quah, I., R.V., S., Rohloff, K., Saylor, J., Saponitsky, D., Triplett, M., Vaikuntanathan, V., Zucca, V.: OpenFHE: Open-Source Fully Homomorphic Encryption Library. Cryptology ePrint Archive, Paper 2022/915.

- accessed: 2024-10-01 (2022). <https://eprint.iacr.org/2022/915>
- [29] Lattigo v5. Online: <https://github.com/tuneinsight/lattigo>. accessed: 2024-10-01 (2023)
  - [30] Fan, J., Vercauteren, F.: Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive, Paper 2012/144. accessed: 2024-10-01 (2012). <https://eprint.iacr.org/2012/144>
  - [31] Ducas, L., Micciancio, D.: FHEw: Bootstrapping homomorphic encryption in less than a second. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology – EUROCRYPT 2015, pp. 617–640. Springer, Berlin, Heidelberg (2015)
  - [32] Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017, pp. 409–437. Springer, Cham (2017)
  - [33] Raghavan Veeraragavan, N., Praneeth Karimireddy, S., Nygård, J.F.: A differentially private kaplan-meier estimator for privacy-preserving survival analysis. arXiv e-prints, 2412 (2024)
  - [34] Rahimian, S., Kerkouche, R., Kurth, I., Fritz, M.: Private and collaborative kaplan-meier estimators. In: Proceedings of the 23rd Workshop on Privacy in the Electronic Society, pp. 212–241 (2024)