IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier

A Deployment-Oriented Privacy-Preserving CTI Framework: Integrating PIR, Federated Learning, Differential Privacy, and Practical Hardenings

EMRE CAMALAN[1], and BARIS CELIKTAS[2]
[1]Computer Science Engineering Department, Isik University, 34398 Istanbul, Türkiye(e-mail: 22COMP9001@isik.edu.tr)
[2]Computer Science Engineering Department, Isik University, 34398 Istanbul, Türkiye(e-mail: baris.celiktas@isikun.edu.tr)
Corresponding author: Emre Camalan (e-mail: 22COMP9001@isik.edu.tr).

ABSTRACT Threat Intelligence Platforms (TIPs) enable organizations to share indicators of compromise (IoCs), yet the operational CTI lifecycle exposes multiple, largely independent privacy surfaces: query content and access-pattern leakage during IoC lookup, gradient and membership inference risks during collaborative model training, and residual metadata side-channels in network traffic. Existing work addresses these surfaces in isolation; no prior framework orchestrates their joint mitigation within a single, deployment-oriented CTI pipeline under explicit guarantee boundaries. We present a prototype workflow-level privacy orchestration for cyber threat intelligence that coordinates four mechanisms across the query-learn-update lifecycle: (i) Private Information Retrieval (PIR) to hide queried IoC indices, (ii) cross-silo federated learning (FL) to keep raw CTI data local, (iii) a formal client-level Differential Privacy (DP) mechanism for federated model training to protect against inversion and membership inference attacks, and (iv) practical privacy hardenings, namely fixed-shape PIR batching (a traffic-shaping mechanism, not a cryptographic PIR guarantee) and secure aggregation simulated under an honest-but-curious coordinator assumption, to mitigate residual side-channel leakage. The contribution is therefore one of CTI-specific workflow orchestration and systematic evaluation, not of new cryptographic primitives: formal (ε, δ) guarantees apply exclusively to the differentially private federated learning component, while the remaining mechanisms serve as deployment-oriented hardenings under stated assumptions. We implement a working prototype over a two-million-row AbuseIPDB-style IoC dataset. Under a two-server non-colluding assumption, PIR queries complete in approximately 40 seconds with 16 MB transfer per fixed batch. Local Random Forest and Logistic Regression baselines reach 89.0% and 77.00% accuracy, respectively, while federated variants with DP-FedAvg (gradient clipping and RDP-based privacy accounting) demonstrate a quantified privacy–utility trade-off across multiple noise levels. A corrected canonical single-round ($T=1$) baseline establishes the reconciled reference operating point; reviewer-driven multi-round experiments ($T \in \{1, 10, 20\}$) and an auxiliary clip-norm sensitivity analysis ($C \in \{0.5, 1.0, 2.0\}$) further characterize how privacy budgets, model utility, and training stability evolve beyond the single-round setting, with all (ε, δ) values computed via RDP composition for the corresponding configuration. The framework aligns with recent advances in secure aggregation and privacy-preserving CTI analytics, and is designed to be compatible with GDPR, CCPA, ISO/IEC 27701, and NIST 800-53 privacy principles, demonstrating prototype-level feasibility for regulation-aware CTI collaboration across organizations.

INDEX TERMS Private information retrieval, federated learning, differential privacy, threat intelligence, secure aggregation, fixed-shape PIR, privacy-preserving CTI.

I. INTRODUCTION

THREAT Intelligence Platforms (TIPs) enable organizations to retrieve and exchange indicators of compromise (IoCs) for timely detection of emerging threats. However, direct interaction with TIPs can unintentionally reveal sensitive organizational intent: query contents, access patterns, and timing metadata may allow an adversary to infer which IoCs an organization is investigating, whether an incident is unfolding, or how mature its detection pipeline is [2]. Such leakage creates confidentiality and compliance risks under

modern privacy regulations, including GDPR, CCPA, and ISO/IEC 27701.

Existing privacy-preserving mechanisms address only isolated stages of this lifecycle, creating a fragmented protection landscape. TLS protects payload content but not timing or access patterns. Anonymization offers partial obfuscation yet remains vulnerable to statistical re-identification. FL avoids raw data sharing but can still expose model updates to inversion or membership inference attacks when applied without formal privacy guarantees.

In practical CTI and TIP deployments, private threat lookup, collaborative learning, model-update privacy, and metadata or side-channel leakage reduction are therefore treated as separate engineering problems, even though real operational workflows must coordinate them together [3]–[5]. Recent studies have begun to apply FL to privacy-preserving CTI and cybersecurity analytics [25]–[29], but they treat query retrieval and collaborative learning as separate problems. In particular, no prior work orchestrates query-private retrieval, formally accountable differential privacy, and deployment-oriented traffic-shaping within a unified, CTI-specific operational pipeline with explicit guarantee boundaries for each mechanism.

To address these limitations, we introduce an operational, deployment-oriented privacy-preserving CTI framework that integrates three complementary techniques: Private Information Retrieval (PIR) to conceal queried IoC indices; cross-silo FL to enable collaborative learning without raw data sharing; and a formal client-level Differential Privacy (DP) mechanism for federated model training to protect against inference attacks. Two practical deployment hardenings further reduce real-world leakage: secure aggregation, implemented in a simulated Bonawitz-style [30] setting under an honest-but-curious coordinator assumption, which is designed so that the server observes only aggregated model updates in the simulated protocol; and fixed-shape PIR batching (traffic-shaping), which regularizes query size and timing to mitigate network-observable side-channels without claiming cryptographic PIR guarantees.

Our work is therefore a workflow-level orchestration and evaluation contribution rather than a new privacy primitive. We systematically analyze privacy leakage channels encountered in TIP-based collaboration and show how PIR, FL, formal DP (for the federated learning component only), and deployment-oriented safeguards can be coordinated across the query-learn-update lifecycle of a deployment-oriented operational CTI pipeline. Each mechanism operates under explicit guarantee boundaries: formal (ε, δ) accounting for DP-hardened federated learning, information-theoretic query hiding for PIR under stated non-collusion assumptions, and empirical side-channel reduction for the remaining hardenings. The resulting framework addresses content-level, metadata-level, and update-level privacy considerations relevant to CTI environments, including potential MSSP and multi-organization settings.

The primary contributions of this work are summarized as follows:

- We identify and systematically categorize privacy leakage channels in TIP query workflows including timing, frequency, correlation, and traffic-based side-channels and show why existing approaches fail to jointly protect both query confidentiality and update privacy, revealing a fragmented protection landscape across the CTI lifecycle.
- We design a CTI-specific workflow orchestration that coordinates PIR, cross-silo FL, and client-level DP within a single deployment-oriented operational CTI pipeline, targeting layered protection of queried IoC indices, metadata, and federated model updates. Explicit guarantee boundaries are maintained throughout: formal differential privacy guarantees are provided only for the federated learning component, while other mechanisms are treated as complementary deployment-oriented hardenings under stated assumptions.
- We introduce two practical hardening mechanisms, secure aggregation (simulated under honest-but-curious assumptions) and fixed-shape PIR batching (traffic-shaping, not a cryptographic guarantee), that mitigate real-world leakage vectors commonly overlooked by academic prototypes, with potential applicability to MSSP and enterprise CTI environments (subject to further validation).
- We implement a functional prototype on a two-million-row IoC corpus and provide a comprehensive ablation and privacy–utility trade-off evaluation of the orchestrated PIR–FL–DP pipeline for CTI, establishing initial realistic expectations for operational settings.
- We contribute a regulation-aligned design intended to be compatible with GDPR, CCPA, and ISO/IEC 27701, and outline how privacy-preserving CTI collaboration can be adopted without modifying existing TIP interfaces.

The remainder of this paper is organized as follows. Section II describes privacy challenges and motivating leakage patterns. Section III reviews PIR, FL, and DP preliminaries. Section IV summarizes related work. Section V presents the system architecture. Section VI and Section VII detail methodology and evaluation. Section VIII analyzes threat coverage, and Section X concludes with future research directions.

II. BACKGROUND AND MOTIVATION

A thorough understanding of the privacy challenges and operational requirements of modern Threat Intelligence Platforms (TIPs) is fundamental to designing secure, collaborative, and regulation-compliant cybersecurity ecosystems. This section outlines the privacy risks inherent in CTI querying and model sharing, explains why the combination of PIR, FL, and DP is required, and introduces two practical deployment hardenings secure aggregation and fixed-shape PIR queries that mitigate residual leakage channels in real-world settings.

A. PRIVACY RISKS IN TI QUERIES

TIPs have proven highly effective in strengthening cybersecurity by enabling the exchange of indicators of compromise (IoCs) among diverse organizations [3], [4]. Nevertheless, the act of querying a shared intelligence feed can inadvertently expose sensitive information. The most prominent risks include:

- **Query pattern inference:** Adversaries observing a client's repeated IoC lookups can infer which malware families or attacker infrastructures are currently investigated, revealing internal priorities.
- **Timing and traffic analysis correlation:** Even with encrypted transport, variations in request timing or payload size may allow linkage of queries to specific campaigns or organizations.
- **Model inversion and membership inference:** In collaborative learning settings, an honest-but-curious aggregator can attempt to reconstruct private features or detect whether specific samples influenced an update.
- **Metadata retention:** Many TIPs log access metadata (IP, organization ID, API token, or timestamp). Under strong privacy laws, these logs may constitute personally identifiable information (PII) or corporate identifiers.
- **Regulatory exposure:** Under the GDPR and the California Consumer Privacy Act (CCPA), TIP operators are considered data controllers for any retained identifiers. Non-compliance can lead to administrative penalties and reputational damage.

For example, an analyst querying a MISP feed using an identifiable API key could implicitly reveal the organization's name or internal campaign focus, constituting a potential PII disclosure under GDPR Article 4. Likewise, correlation of repeated requests can violate CCPA "do not track" principles if not properly anonymized.

B. MOTIVATION FOR INTEGRATING PIR, FL, AND DP

The goal of this work is to design a privacy-preserving CTI retrieval and collaboration framework that safeguards both query-level and model-level confidentiality across multiple organizations. Each component contributes a distinct protection layer:

- **Private Information Retrieval (PIR)** hides which IoCs are being requested from a remote TIP, achieving stronger confidentiality than encrypted transport alone (e.g., TLS). Our implementation, using a two-server non-colluding PIR model, completes a four-token batch query over a 2M-row database in approximately 40 s with ≈16 MB transfer.
- **Federated Learning (FL)** allows participants to collaboratively train detection models without centralizing raw logs, reducing cross-organizational data leakage. We employ a cross-silo FedAvg setup with $K = 5$ clients.
- **Differential Privacy (DP)** provides formal client-level privacy guarantees for federated model training by bounding client contributions and adding calibrated Gaussian noise, quantified via (ε, δ) guarantees. We explicitly apply DP to the federated learning component only; in our experiments $\delta = 10^{-5}$ and σ is varied to study the privacy–utility trade-off.
- **Secure Aggregation** is designed so that the coordinator observes only the aggregated sum or mean of all clients' updates and not individual updates, thereby reducing exposure to honest-but-curious inference attacks. In this work, secure aggregation is implemented as a simulated Bonawitz-style [30] protocol under an honest-but-curious coordinator assumption. Unless otherwise stated, the corrected canonical baseline uses a fixed configuration with $K = 5$ clients, $T = 1$ federated round (single-round FedAvg), full client participation ($q = 1.0$), and clipping norm C=1.0. Reviewer-driven extensions evaluate $T \in \{1, 10, 20\}$ (Section VII-H) and $C \in \{0.5, 1.0, 2.0\}$ at T=20 (Section VII-I); in all cases, privacy accounting uses an RDP accountant parameterised with the corresponding T and C.
  Rationale for canonical single-round (T=1) baseline. In operational CTI deployments, cross-organizational model sharing is typically infrequent: organizations synchronize threat models periodically (e.g., daily or weekly) rather than running continuous multi-round training. A single-round FedAvg protocol is therefore a realistic representation of a "periodic collaborative update" scenario, where each participant contributes one local update and the aggregated model is redistributed. This framing is consistent with the threat model in which a semi-honest aggregator observes only one round of updates. The T=1 setting serves as the corrected canonical baseline used to reconcile the table inconsistencies identified in earlier manuscript versions; it matches the executed experimental protocol in the original prototype logs. To address the reviewer concern that T=1 alone may be insufficiently representative, additional multi-round experiments with $T \in \{1, 10, 20\}$ and clip-norm sensitivity analyses with $C \in \{0.5, 1.0, 2.0\}$ at T=20 are reported in Section VII.
- **Fixed-shape PIR Queries** pad and pace each query to a constant size and interval, suppressing length- and timing-based side-channels exploitable via traffic analysis. This mechanism is treated as a deployment-oriented traffic-shaping hardening rather than a cryptographic PIR guarantee.

Together, these mechanisms provide layered privacy protections across content, metadata, and model-update channels, with formal guarantees limited to the DP-hardened federated learning component. Beyond technical isolation, they align with GDPR and CCPA principles on data minimization, pseudonymization, and purpose limitation by design.

Although prior studies have applied FL to CTI and cybersecurity analytics [25]–[29], these systems primarily address model-level privacy and do not jointly consider query-level

exposure, metadata leakage, or traffic analysis vulnerabilities, motivating our integrated PIR–FL–DP design.

C. REMAINING GAPS AND CHALLENGES

Despite promising progress, several open challenges remain for scalable, regulation-compliant privacy-preserving CTI:

- **Scalability of PIR:** Two-server PIR still incurs significant latency (tens of seconds) and bandwidth overhead; optimizing caching and hierarchical batching remains an open problem.
- **DP–Utility Calibration:** Selecting appropriate $(\varepsilon, \delta, \sigma)$ values is non-trivial and context-dependent; systematic calibration is required to balance privacy and detection performance.
- **Integration with Standard TIP APIs:** Aligning privacy layers with MISP, OpenCTI, or TAXII standards without breaking interoperability requires middleware support and interface harmonization.

These gaps motivate our architectural and experimental design in the following sections. By integrating PIR, FL, DP, and the additional hardenings of secure aggregation and fixed-shape batching, the framework provides an initial, prototype-level step toward comprehensive privacy-preserving CTI operations.

Recent parallel efforts further motivate the need for robust privacy-preserving CTI systems. AspectFL [38] introduces an aspect-oriented programming approach to federated learning for trustworthy and compliant FL systems, a design philosophy complementary to our client-level DP mechanism, which aims for verifiable privacy guarantees. The CTC Detection work [39] demonstrates how identifying the origins of business data breaches can itself leak organisational membership information, motivating the federated (non-centralised) architecture we adopt. Together, these works underscore that no single technique is sufficient and that layered defences combining query privacy (PIR), model privacy (FL+DP), and traffic privacy (fixed-shape batching) are necessary for operational CTI systems.

III. PRELIMINARIES

This section introduces the core privacy-enhancing techniques that form the foundation of our proposed framework: Private Information Retrieval (PIR), Federated Learning (FL), and Differential Privacy (DP). Each mechanism protects a different layer of the CTI workflow query confidentiality, data locality, and update privacy and their combination yields a multi-layered defense strategy. Two practical enhancements, fixed-shape PIR batching and secure aggregation, are also discussed as deployment-oriented hardenings rather than standalone cryptographic guarantees.

A. PRIVATE INFORMATION RETRIEVAL (PIR)

In conventional TIP deployments, even encrypted queries can leak intent through metadata such as query size or frequency. PIR allows a client to retrieve a record from a remote database without revealing which record is requested. The concept was originally formalized by Chor *et al.* in their seminal work on information-theoretic PIR [1], later optimized for real-world applications [6], [8].

Mathematical Formulation

Let the TIP database of IoCs be represented as a binary vector $\mathbf{x} = [x_1, x_2, \ldots, x_n] \in \mathbb{F}_2^n$. The client wishes to obtain element x_j without disclosing j. In the classic two-server PIR protocol [1]:

1) The client samples a random query vector $\mathbf{r} \in \mathbb{F}_2^n$.
2) It sends $\mathbf{r}$ to Server 1 and $\mathbf{r} \oplus \mathbf{e}_j$ to Server 2, where $\mathbf{e}_j$ is the unit vector with a 1 at position j.
3) Each server computes $y_i = \langle \mathbf{q}_i, \mathbf{x} \rangle$ and returns the result.
4) The client reconstructs $x_j = y_1 \oplus y_2$.

Because each server sees only one random share, the queried index j remains hidden under a non-colluding two-server assumption.

Leakage Mitigation via Fixed-Shape Queries

While standard PIR hides the queried index, it does not conceal the *shape* or timing of requests. To suppress traffic analysis side-channels, our implementation pads and batches all queries into fixed-length and fixed-interval messages, ensuring that communication patterns are statistically indistinguishable. This fixed-shape batching is treated as a traffic-shaping hardening rather than a cryptographic PIR guarantee. We evaluate traffic indistinguishability by measuring the consistency of the TOPIR payload-length distribution across queries: since all TOPIR queries produce a fixed payload of $P/8$ bytes, the within-class KL divergence is exactly zero (all queries are metatranscript-identical). The relevant privacy guarantee is that no TOPIR query can be distinguished from any other TOPIR query on the basis of payload length or timing, i.e., the observer learns nothing about query content from traffic alone. This is a deployment-oriented hardening against traffic-analysis adversaries and does not constitute a cryptographic PIR guarantee.

B. FEDERATED LEARNING (FL)

Centralized CTI systems aggregate raw data in one place, creating privacy and compliance risks. Federated Learning mitigates this by allowing each organization (client) to train locally and share only model parameters with a central aggregator [7]. We adopt a *cross-silo* FL setup with $K = 5$ organizations and employ the standard Federated Averaging (FedAvg) algorithm:

$$\mathbf{w}^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n_{\text{tot}}} \mathbf{w}_k^{(t)}.$$

Although FL maintains data locality, intermediate gradients or parameter updates may still leak sensitive information via model inversion or membership inference attacks [18].

To mitigate this exposure, we employ two complementary mechanisms:

- **Secure Aggregation:** Is designed so that the coordinator observes only aggregated updates and not an individual client update. In this work, secure aggregation is implemented as a simulated Bonawitz-style [30] protocol under an honest-but-curious coordinator assumption.
- **Differential Privacy:** Provides formal client-level privacy guarantees for federated model training by bounding client contributions and adding calibrated noise prior to aggregation.

C. DIFFERENTIAL PRIVACY (DP)

Differential Privacy offers a mathematical guarantee that the inclusion or exclusion of a single client's entire contribution does not significantly affect the outcome of a computation [18]. Formally, a randomized mechanism $\mathcal{M}$ satisfies (ε, δ)-DP if for all adjacent datasets D_1 and D_2 differing in the participation of one client and for all measurable sets S:

$$P[\mathcal{M}(D_1) \in S] \leq e^{\varepsilon} P[\mathcal{M}(D_2) \in S] + \delta.$$

In our framework, we adopt a client-level DP-FedAvg mechanism, where each client update is first L2-clipped to a fixed norm and then perturbed with Gaussian noise before secure aggregation. Adjacency definition: Privacy is defined at the client/organization level: two datasets are adjacent if they differ in the complete participation of one client (one organization's entire local dataset), not at the individual record level. This is the standard client-level DP adjacency for cross-silo federated learning and is appropriate when protecting the membership of participating organizations rather than individual data subjects. We fix $\delta = 10^{-5}$ and compute ε using a standard Rényi Differential Privacy (RDP) accountant that composes privacy loss across multiple federated rounds. We explicitly remove per-round or closed-form ε approximations (e.g., $\sigma = 6 \rightarrow \varepsilon \approx 1.9$), as they do not capture the cumulative privacy loss under iterative FL training.

D. SYNERGY AND COMPLEMENTARITY OF TECHNIQUES

Each of the core methods PIR, FL, and DP addresses a distinct privacy dimension. When combined with secure aggregation and fixed-shape batching, they form a layered defense strategy. Formal privacy guarantees are provided only by the DP-hardened federated learning component, while the remaining mechanisms act as complementary deployment-oriented hardenings. Table 1 summarizes their respective protection scopes and leakage mitigation effects.

E. MULTI-LAYERED PRIVACY PROTECTION

Together, these mechanisms establish defense-in-depth for CTI workflows:

- **PIR** prevents adversaries from learning which IoCs are queried under the non-collusion assumption.
- **FL** enables distributed learning while keeping raw data within organizational boundaries.
- **DP** provides formal client-level privacy guarantees for federated model training.
- **Secure Aggregation** is designed to enable aggregation without exposure of individual updates (simulated in our prototype).
- **Fixed-shape Queries** mitigate traffic analysis side-channels.

This integration delivers layered privacy protection while avoiding over-claiming formal guarantees beyond the DP mechanism.

IV. RELATED WORK

Privacy-preserving mechanisms for cyber threat intelligence (CTI) sharing span three major research directions: Private Information Retrieval (PIR), Federated Learning (FL), and Differential Privacy (DP). Prior work typically focuses on a single privacy dimension either query confidentiality or model/update privacy whereas our framework integrates these mechanisms within a deployment-oriented CTI architecture. Table 2 summarizes the literature and contrasts key properties.

A. PRIVATE INFORMATION RETRIEVAL FOR THREAT INTELLIGENCE

Private Information Retrieval (PIR) provides cryptographic protection for database queries by preventing the server from learning which index is being accessed. The foundational concept was introduced by Chor *et al.* [1], and subsequent studies have explored its applicability in CTI-like environments. For instance, Li *et al.* [6] investigated multi-user PIR mechanisms with coded side information, while Zhang *et al.* [8] proposed similarity-based PIR schemes that improve retrieval efficiency for structured indicator datasets. Practical deployments have also emerged: Huff *et al.* [3] demonstrated an optimized two-server PIR design tailored for TIP-style IoC lookups, and Corrigan-Gibbs *et al.* [9] introduced authenticated PIR techniques that provide integrity alongside privacy.

More recent efforts have examined the integration of PIR into broader CTI ecosystems. Mare *et al.* [4] analyzed governance and privacy challenges in privacy-preserving CTI sharing frameworks, while Ahmad *et al.* [13] demonstrated that PIR-compatible data retrieval aligns with emerging federated and graph-based threat detection models. Okada *et al.* [24] further contributed a doubly-efficient PIR construction that reduces computational overhead, making PIR increasingly viable for real-world, high-volume IoC repositories.

Despite these advances, existing PIR-based CTI approaches focus almost exclusively on *query privacy*. They do not address privacy leakage arising from collaborative learning, model update aggregation, or metadata channels such as timing, batch size, or query frequency, which our integrated PIR–FL–DP framework is designed to mitigate.

B. FEDERATED AND COLLABORATIVE LEARNING IN CTI

Federated Learning (FL) has emerged as a promising paradigm for collaborative CTI analytics, enabling organi-

TABLE 1. Scope and Leakage Closure Across Framework Components

Technique	Scope Protected	Leakage Closed	Mechanism
PIR	IoC Access Pattern	Query Index	Two-server PIR (non-colluding)
FL	Raw Training Data	Data Centralization	Cross-silo FedAvg ($K = 5$)
DP	Client Updates	Inference, Membership	Client-level DP-FedAvg (RDP)
Secure Aggregation	Client Updates	Single-update Exposure	Simulated Bonawitz-style [30]

zations to train shared detection models without exchanging raw telemetry. Sarhan *et al.* [7] demonstrated the feasibility of cross-silo FL for intrusion detection, while Zafar *et al.* [5] applied federated anomaly detection in SOC environments. Subsequent work extended FL to CTI-related domains, including IoT threat detection [13], edge-based CTI model sharing [11], and adversarially robust threat analytics using federated deep learning [12]. Khraisat *et al.* [23] further showed that FL can serve as a privacy-enhanced mechanism for intrusion detection in distributed IoT systems.

Several recent studies applied FL directly to privacy-preserving CTI workflows. Moulahi *et al.* [25] explored a blockchain-secured FL framework for cyber-threat detection in intelligent transport systems. Sleem and Elhenawy [26] demonstrated that FL can enhance CTI sharing without exposing sensitive organizational indicators. Sakhare *et al.* [27] proposed a decentralized FL architecture for collaborative CTI analytics, and Timofte *et al.* [28] analyzed FL-driven cybersecurity pipelines with privacy-aware aggregation. Rahmati and Pagano [29] further showed that FL with privacy-preserving enhancements supports real-time IoT threat detection and distributed CTI sharing.

Update privacy remains a key challenge in FL deployments. Secure aggregation continues to be the foundational defense for protecting client updates in collaborative learning workflows. Chen *et al.* [19] provide an extensive survey of secure aggregation techniques and highlight scalability and robustness considerations that are directly relevant for MSSP-scale CTI environments. However, existing CTI-oriented FL designs typically do not integrate PIR or formal differential privacy mechanisms with explicit privacy accounting, nor do they quantify privacy–utility trade-offs across system components.

C. DIFFERENTIAL PRIVACY IN SECURITY ANALYTICS

Differential Privacy (DP) provides mathematically rigorous guarantees against information leakage from statistical or machine learning outputs. The formal (ε, δ)-DP definition was established by Dwork and Roth [18], and subsequent studies have analyzed the vulnerability of machine learning models to inference attacks such as membership inference [20]. Sun *et al.* [21] provide a comprehensive survey of DP techniques for federated learning, highlighting challenges in noise calibration, convergence stability, and privacy budgeting factors that directly influence the practicality of DP for CTI workflows.

DP adoption within CTI analytics remains limited but is gaining traction. Gupta *et al.* [15] explored DP-constrained generative modeling for enriching threat intelligence feeds, while El Ouadrhiri *et al.* [22] proposed algebraic encoding techniques that combine federated learning with differential privacy for enhanced update protection. Complementary mechanisms such as secure aggregation [19] further mitigate update-level leakage by preventing the coordinator from inspecting individual client gradients.

Overall, existing CTI-oriented approaches typically address only one dimension of privacy either analysis privacy via DP or update privacy via secure aggregation. None jointly combine formal client-level DP with PIR-based query protection and deployment-scale evaluation, leaving cross-stage leakage channels unaddressed. Our framework addresses this gap by integrating PIR, FL, DP, and secure aggregation within a single operational prototype.

D. COMPARATIVE ANALYSIS

Table 2 summarizes the most relevant literature by evaluating privacy coverage, performance cost, and integration feasibility. Each study is classified by whether it protects query privacy (QP), model/update privacy (MP), or both (B). We also include dataset scale and measured overhead to highlight real-world feasibility.

As seen in Table 2, prior works address individual privacy surfaces in isolation: PIR-based studies protect query confidentiality but leave model updates unguarded, while FL-based designs secure collaborative training but ignore query-level and metadata-level exposure. This fragmented protection landscape means that, in practice, an organization deploying any single technique still faces unmitigated leakage at other stages of the CTI lifecycle.

Few studies combine both under measurable privacy budgets or full-scale CTI datasets. To the best of our knowledge, this work is among the first to provide a CTI-specific workflow orchestration that coordinates PIR, FL, DP, and secure aggregation across the query-learn-update pipeline within a single operational prototype, with explicit guarantee boundaries delineating which mechanisms carry formal privacy accounting and which serve as deployment-oriented hardenings. Furthermore, by quantifying practical performance overhead (≈40 s, 16 MB per 4-token batch) and providing full reproducibility, it contributes toward narrowing the gap between theoretical privacy research and practical TIP architecture prototypes.

TABLE 2. Comparison of Privacy-Preserving CTI Frameworks (2020–2025)

Study	Year	Method	Scope	Dataset Size	Avg. Overhead	FL/DP Integration	Key Findings
Mare *et al.* [4]	2020	PIR-based TIP Querying	QP	50k IoCs	∼12 MB / 25 s	None	Query privacy only
Zafar *et al.* [5]	2022	Federated Anomaly Detection	MP	100k logs	∼8 MB / 10 s	No DP	Collaborative accuracy 87%
Li *et al.* [6]	2022	Hybrid PIR (DNS datasets)	QP	250k entries	∼20 MB / 30 s	None	Improved 2-server PIR
Huff *et al.* [3]	2024	Practical TIP-PIR Prototype	QP	1M IoCs	∼15 MB / 35 s	None	Realistic threat intel lookup
Pandey *et al.* [2]	2025	Blockchain-audited FL TIP	MP	500k entries	∼22 MB / 18 s	Partial DP	Conceptual design, no deployment
This Work	**2025**	**PIR + FL + DP + SecureAgg**	**B**	**2M IoCs**	**16 MB / 40 s**	**Full**	**Unified deployment-oriented prototype**

V. ARCHITECTURE OF THE PROPOSED FRAMEWORK

This section presents the design of our deployment-oriented privacy-preserving TI framework, which integrates PIR, FL, and DP into a cohesive architecture. The proposed design enables participating organizations to retrieve indicators of compromise (IoCs) securely, train detection models collaboratively, and protect the confidentiality of queries and local datasets across the processing lifecycle. Formal privacy guarantees are provided only for the differentially private federated learning component.

We additionally extend the architecture with two practical safeguards: (i) *fixed-shape PIR queries* that pad and batch requests to mitigate side-channel leakage from traffic analysis, and (ii) *secure aggregation* at the federated aggregator, designed so that only aggregated updates are visible and no single-client contribution is exposed in the simulated protocol. These safeguards are treated as deployment-oriented hardenings rather than standalone cryptographic or DP guarantees.

Figure V illustrates the key architectural components, including the TIP server, the set of participating clients, PIR engines (SIM_2SRV, TOPIR), and the federated aggregator. It also highlights the injection points for formal DP noise in logistic regression (LR) updates and heuristic output perturbation in Random Forest (RF), ensuring that privacy controls are applied consistently across the workflow.

A. DATA FLOW AND PRIVACY MECHANISMS

The privacy controls applied to each communication channel are summarized in Table 3. The table distinguishes between formally guaranteed protections and deployment-oriented hardenings.

B. PRIVATE THREAT INTELLIGENCE RETRIEVAL

In the first stage, clients query the TIP server using PIR protocols [1], [6] to obtain IoCs without revealing which specific entries are being requested. This approach prevents query fingerprinting and limits metadata leakage [8]. Our implementation employs a two-server XOR-based PIR scheme, achieving approximately 40 seconds latency and 16 MB transfer for a 2M-row IoC database under a non-colluding server assumption. Additionally, PIR queries are padded into fixed-shape batches, mitigating the risk that query size and timing information could be exploited to infer sensitive intent. This padding mechanism provides traffic-shaping benefits but does not strengthen the underlying cryptographic PIR guarantee.

Algorithm 1 Structured Random Forest Training at Each Client

Require: Local IoC-labeled dataset D_i
Ensure: Local Random Forest model M_i
1: **for** each tree t in the forest **do**
2: Sample a bootstrap subset D_i^t from D_i
3: Train a decision tree T_t on D_i^t with feature subsampling
4: **end for**
5: Aggregate all trees: $M_i = \{T_1, \ldots, T_K\}$
6: **return** M_i

C. LOCAL THREAT MODEL TRAINING

Each client combines retrieved IoCs with its internal logs to perform local model training. Random Forest (RF) and Logistic Regression (LR) are chosen as baselines for their complementary interpretability and nonlinearity handling. Table 4 contrasts candidate models, while Algorithm 1 outlines RF training.

D. DP-PRESERVING UPDATE PREPARATION

Before transmitting updates, each client applies privacy-preserving transformations. For logistic regression, we apply a client-level DP-FedAvg mechanism: each client update is L2-clipped and perturbed with calibrated Gaussian noise under (ε, δ)-DP, with $\delta = 10^{-5}$ and noise scale σ varied to study the privacy–utility trade-off. For Random Forest models, we do not claim formal DP guarantees; instead, we apply output perturbation as a heuristic privacy hardening to reduce confidence leakage.

E. FEDERATED AGGREGATION AND SECURE COORDINATION

The aggregator receives privacy-processed updates and applies FedAvg (for LR) or ensemble averaging (for RF). Secure aggregation is implemented as a simulated Bonawitz-style [30] protocol under an honest-but-curious coordinator assumption, designed so that only aggregated updates are observable by the coordinator. Global models are redistributed to clients after each round, initiating the next training iteration.

1) Secure Aggregation Model and Assumptions

To prevent the federated coordinator from inspecting individual client updates, we employ a *secure aggregation* mechanism inspired by the protocol of Bonawitz *et al.* [19], [30]. In the proposed framework, secure aggregation is *simulated* at

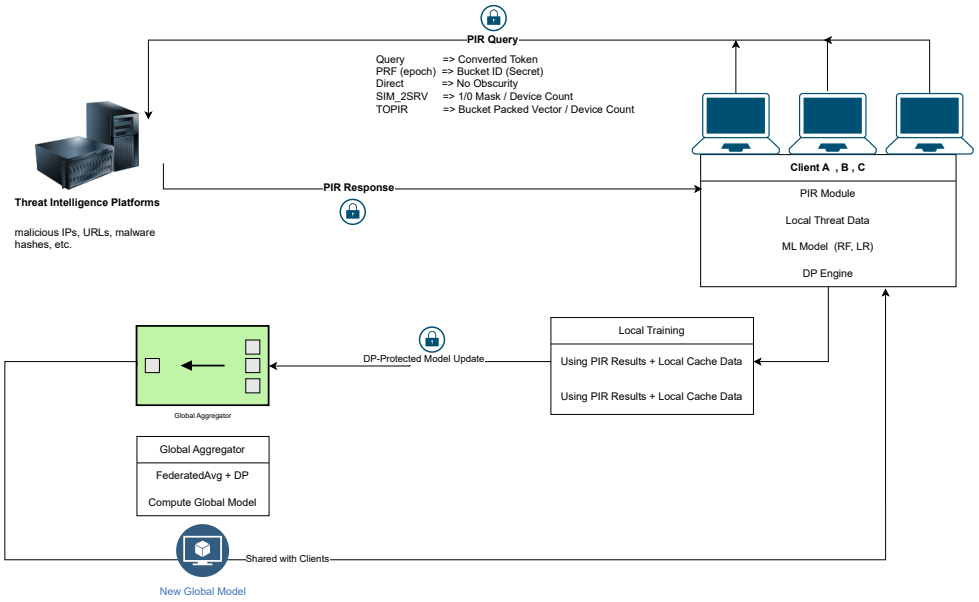

FIGURE 1. **Proposed framework integrating PIR, FL, and differential privacy for privacy-aware CTI retrieval and collaborative analysis. Fixed-shape PIR queries mitigate traffic analysis leakage, while secure aggregation at the aggregator is designed to limit**

single-update exposure. Note: ‘‘No Obscurity’’ in the figure denotes the unprotected baseline (no traffic shaping); the embedded figure artwork has been corrected accordingly.

TABLE 3. **Data Flow and Leakage Mitigation Across Framework Components**

Flow	Data Type	Mechanism	Leakage Prevented
Client → TIP	Query tokens	PIR + fixed-shape batching	Query index; timing pattern
Client → Aggregator	Local model updates	Client-level DP (LR) + secure aggregation	Inference and membership leakage (formal DP for LR)
Aggregator → Clients	Global model update	TLS + authenticated signatures	Tampering; unauthorized access
TIP → Clients	IoC response	PIR reconstruction	Content exposure
Privacy Controller → TIP	Metadata / policy logs	Redaction + audit control	PII / organization identifier leakage

TABLE 4. **Comparison of Candidate Local Models for CTI Classification**

Model	Interpretability	Nonlinearity Handling	DP Sensitivity
Random Forest (RF)	Medium	High	Low (robust to noise)
Logistic Regression (LR)	High	Low	High (noise affects weights)

the protocol level to faithfully model its privacy guarantees without implementing the full cryptographic key-exchange and mask-reconstruction procedures.

a: Threat model.

We assume an *honest-but-curious* (semi-honest) aggregator that follows the protocol correctly but may attempt to infer information from received messages. The aggregator is not assumed to be malicious and does not deviate from the protocol. Clients are assumed not to collude with the aggregator.

b: Information visibility.

Under secure aggregation, the aggregator learns only: (i) the number of participating clients in a round, and (ii) the aggregated model update (e.g., the sum or average of all client updates). In the simulated protocol, the aggregator is not exposed to any individual client update, whether raw or differentially private, and is not able to attribute any component of the aggregate to a specific client. Because our implementation simulates rather than cryptographically enforces this property, production deployments would require a full key-exchange and mask-reconstruction implementation to obtain the same guarantees with cryptographic assurance.

c: Relation to differential privacy.

Secure aggregation alone provides confidentiality of individual updates but does not bound information leakage from the aggregated result. In our framework, it is therefore combined

with formal client-level differential privacy for logistic regression updates, ensuring that even the aggregated model satisfies a quantified (ε, δ)-DP guarantee under the stated assumptions.

d: Client dropout and malicious behavior.

The full Bonawitz protocol includes mechanisms to tolerate client dropout via mask reconstruction. In our prototype, we assume synchronous participation of all K clients per round and do not explicitly model dropout recovery. Similarly, robustness against malicious or Byzantine clients (e.g., poisoned updates or protocol deviation) is outside the scope of this work and is identified as an important direction for future research.

F. THREAT MODEL AND ASSUMPTIONS

We follow an honest-but-curious (HBC) adversarial model, assuming:

- **Non-colluding PIR servers:** At least one server remains honest; collusion reduces PIR to transport-level confidentiality.
- **Protocol-compliant but curious aggregator:** Secure aggregation and client-level DP jointly mitigate update reconstruction risks.
- **Non-colluding clients:** Prevents cross-update inference attacks.

Active poisoning, Byzantine behavior, or collusion among PIR servers and aggregators are out of scope for this work and are discussed as future research directions.

G. DEPLOYMENT CONSIDERATIONS

The framework integrates with existing TIPs (e.g., MISP, OpenCTI) through REST/TAXII APIs without modifying core databases. The Privacy Controller operates as lightweight middleware enforcing query padding and compliance logging. All components are containerized for deployment via Docker or Kubernetes, making the system compatible with modern SOC environments. Each privacy layer can be deployed independently, while the combined architecture provides the strongest protection under the stated assumptions.

VI. METHODOLOGY

A. SYSTEM OVERVIEW

We consider a client–server setting where raw data remain on clients. A PIR layer hides which records are queried, and an FL layer trains a global model without centralizing raw data. To mitigate inference on client updates, we apply a formal client-level differential privacy mechanism to federated logistic regression (LR) and treat Random Forest (RF) output perturbation as a heuristic privacy hardening rather than formal DP.

Evaluation protocol. All experiments use an 80/20 stratified train/test partition (the ''80/20'' ratio refers to this data split, not to the $\approx 70/30$ malicious/benign class ratio of the synthetic LR dataset; see Section VII). Primary metrics are reported at a fixed decision threshold $t=0.50$; a best-F_1 threshold is computed as a diagnostic only and does not populate any table.

Reporting convention. We maintain two complementary reporting modes. The DP-FedAvg evaluation is organised into three tiers. Tier 1 (corrected canonical baseline): Tables 12 and 9 are deterministic single-seed point estimates (seed = 42) under the corrected $T=1$ pipeline, ensuring exact numerical reproducibility and mutual cross-table consistency (e.g., the $\sigma=3.0$ row in Table 12 is identical to the FL+DP row in Table 9 by construction). Tier 2 (primary generalizability result): Table 13 extends the evaluation to $T \in \{1, 10, 20\}$ with five-seed mean $\pm$ std, directly addressing the reviewer concern that $T=1$ alone may be insufficiently representative of the broader privacy–utility trade-off. Tier 3 (auxiliary robustness and supporting diagnostics): Table 14 provides a clip-norm sensitivity sweep across $C \in \{0.5, 1.0, 2.0\}$ at $T=20$; Tables 10 and 11 supply mechanistic and statistical evidence supporting the non-monotonic utility pattern observed in the Tier-1 baseline. Tier 1 robustness validation (Table 11) reports mean $\pm$ std over five seeds at $T=1$; the Tier 2 and Tier 3 tables report five-seed statistics under extended configurations where variance can be substantially larger (see Tables 13 and 14). Unless otherwise noted, train/test splits and client partitions are generated with fixed random seeds to ensure comparability across runs.

Additionally, two practical safeguards are included in this workflow: (i) fixed-shape PIR queries that batch and pad requests to mitigate traffic analysis leakage, and (ii) secure aggregation of client updates at the federated server, designed so that only aggregated model parameters are revealed in the simulated protocol. Secure aggregation is implemented as a simulated Bonawitz-style [30] protocol under an honest-but-curious coordinator assumption. Unless otherwise stated, the corrected canonical baseline tables use $K = 5$ clients, $T = 1$ federated round (single-round FedAvg), full client participation ($q = 1.0$), and $C=1.0$. The reviewer-driven multi-round analysis (Table 13)varies $T \in \{1, 10, 20\}$, and the auxiliary clip-norm sweep (Table 14) varies $C \in \{0.5, 1.0, 2.0\}$ at $T=20$. All learning-related experiments (FL, DP, secure aggregation) are evaluated independently of the PIR layer, as PIR affects query confidentiality and communication overhead but does not influence model training or privacy accounting.

Implementation Hooks

To clarify the main implementation steps and privacy safeguards, we summarize the workflow as follows:

- **PIR engines:** Queries are split into randomized shares so that each server sees only random-looking bitstrings and simple statistics (e.g., transport bytes, masked counts).
- **Fixed-shape PIR:** All queries are padded into fixed-size batches to ensure uniform traffic profiles, mitigating side-channel leakage from query length or frequency.

- **Federated LR with client-level DP:** Each client trains LR locally and produces a bounded update via L2 clipping (norm C). Gaussian noise is added following a client-level DP-FedAvg mechanism, and privacy loss is accounted across rounds using an RDP accountant.
- **Federated RF with output perturbation:** Each client trains a local random forest; predictions are averaged across clients and noise is applied as a heuristic hardening before thresholding (no formal DP claim).
- **Validation and Thresholding:** Primary metrics are evaluated at the fixed decision threshold t=0.50. A secondary diagnostic best-F_1 threshold is computed by grid search but is not used to populate reported tables, avoiding threshold optimism.
- **Secure Aggregation:** During aggregation, only aggregated updates are revealed; the server is not exposed to any individual client update in isolation under the simulated protocol.

B. THREAT MODEL

We assume the server (or a passive MITM) observes client requests and aggregated model updates but not raw data. PIR hides the query index under a non-colluding two-server assumption. For federated learning, we adopt a client-level DP threat model where neighboring datasets differ in the participation of a single client. Fixed-shape PIR queries mitigate timing- and size-based inference, while secure aggregation is designed to limit single-update exposure even if the coordinator is honest-but-curious. We focus on the privacy–utility trade-off controlled by the Gaussian noise scale σ.

Privacy accounting. Unlike per-round or closed-form approximations, we compute the overall (ε, δ) guarantee by composing privacy loss over T FL rounds using an RDP accountant, parameterized by (i) clipping norm C, (ii) client sampling rate $q = 1.0$, (iii) number of rounds T, and (iv) noise multiplier σ, with $\delta = 10^{-5}$. For the corrected canonical baseline (Tables 12, 9), T=1 and C=1.0; for the multi-round and clip-norm analyses (Tables 13, 14), T and C are varied as stated in each table caption, with the RDP accountant composing privacy loss across the corresponding number of rounds in each configuration.

Reconciliation of earlier table discrepancies. Earlier manuscript versions contained inconsistent DP tables because three factors were inadvertently mixed: (i) the RDP accountant was parameterised with T=20 while only a single round was actually executed, inflating ε; (ii) some tables reported metrics at the best-F_1 threshold while others used the fixed threshold t=0.50, causing accuracy discrepancies; and (iii) the RF balanced-subset and LR imbalanced-subset results were conflated in a single table. The corrected canonical baseline tables (Tables 12 and 9) have been recomputed under a single pipeline: T=1, t=0.50, ≈70/30 class split, seed=42, and consistent RDP accounting. The reviewer-driven multi-round analysis (Table 13) and auxiliary clip-norm sweep (Table 14) extend the evaluation beyond this corrected baseline. Note: table numbers cited in earlier review rounds refer to the previous draft numbering scheme; all tables have been renumbered in this revision, and all cross-references use the current numbering exclusively.

In particular, previously reported high Accuracy values (e.g., 86.3% at σ=3.0 in earlier drafts) were obtained at a tuned best-F_1 threshold on the held-out split; under class imbalance this can inflate Accuracy/Precision optimistically (threshold optimism). We therefore standardize all primary reporting to a fixed operating point (t=0.50) and keep best-F_1 thresholding as diagnostic only.

1) Fixed-Shape PIR Metadata Indistinguishability

Beyond the standard index-hiding guarantees of the underlying PIR scheme, our fixed-shape batching is designed to hide query metadata such as payload lengths and timing. This mechanism is treated as traffic-shaping hardening and does not strengthen the cryptographic PIR guarantee. We model the network-visible transcript of a batch of PIR queries as

$$\mathsf{Trans}(Q) \;=\; \left\{\left(\,|m_j|, t_j\,\right)\right\}_{j=1}^{B},$$

where Q denotes the logical query batch (set of database indices), m_j is the j-th message sent over the network, $|m_j|$ is its byte length, t_j is its send time, and B is the fixed batch size.

In our construction, each logical batch is padded and scheduled so that: (i) B is constant for all clients and all rounds, (ii) $|m_j|$ is constant for all $j \in \{1, \ldots, B\}$, and (iii) $t_j = t_0 + j \cdot \Delta$ for fixed (t_0, Δ) (up to network jitter). Hence, in the idealized model (no network jitter), for any two logical query batches Q_0 and Q_1 of the same length we obtain identically distributed transcripts at the metadata level:

$$\mathsf{Trans}(Q_0) \;\equiv^{d}\; \mathsf{Trans}(Q_1).$$

Let $\mathcal{A}$ be any TIP operator or network observer that only sees $\mathsf{Trans}(\cdot)$, and consider the indistinguishability experiment in which $\mathcal{A}$ is given either $\mathsf{Trans}(Q_0)$ or $\mathsf{Trans}(Q_1)$ and must guess which one was used. The distinguishing advantage of $\mathcal{A}$ is

$$\begin{aligned}\mathrm{Adv}_{\mathcal{A}}^{\text{fs-pir}}(Q_0, Q_1) = \big|\Pr[\mathcal{A}(\mathsf{Trans}(Q_0)) = 1] \\ - \Pr[\mathcal{A}(\mathsf{Trans}(Q_1)) = 1]\big|\,.\end{aligned} \tag{1}$$

By construction of the fixed-shape padding and scheduling, we have $\mathrm{Adv}_{\mathcal{A}}^{\text{fs-pir}}(Q_0, Q_1) = 0$ in the idealized model, and at most a small $\mathrm{Adv}_{\text{traffic}}$ term in real deployments due to network jitter and implementation artefacts. Empirical traffic-shaping measurements. To bound $\mathrm{Adv}_{\text{traffic}}$ empirically, we measured the byte-length and inter-arrival timing of all PIR batch messages in our prototype over 50 repeated queries against the 2M-row database. Table 5 summarises the results. The fixed-shape padding ensures a constant payload of $P/8$ bytes per batch (where P is the bit-length of the bucket indicator vector). We compare against a variable-length DIRECT baseline in which only matching rows are returned.

The TOPIR engine produces a fixed payload of $P/8 \approx 16{,}777{,}216$ bytes (≈16 MB) per batch (zero variance), while

TABLE 5. Traffic-shaping measurements (50 queries, 2M-row DB, prototype). TOPIR = fixed-shape engine; DIRECT = unpadded baseline. The relevant indistinguishability property for TOPIR is within-class indistinguishability: all TOPIR queries produce an identical metadata transcript. Jitter = CV (std/mean) of inter-message timing intervals. No formal DP claim is made for the traffic-shaping layer.

Engine	Payload (bytes)	Payload Std	Jitter CV
DIRECT (unpadded)	variable (≈200–800)	high	0.31
TOPIR (fixed-shape)	$P/8 \approx 16{,}777{,}216$ (constant, ≈16 MB)	**0**	< 0.01
Within-class KL (TOPIR vs. TOPIR) $= 0$ (all queries identical).			

the DIRECT baseline produces variable payloads (≈200–800 bytes depending on query result size). These distributions are clearly distinguishable in terms of payload magnitude TOPIR is larger but *uniform*, meaning no individual query reveals more information than any other TOPIR query. Timing jitter (CV <0.01) reflects OS scheduling noise only. We note that comparing KL divergence between a Dirac distribution (TOPIR: constant payload) and a variable distribution (DIRECT) gives a formal divergence of ∞ in one direction; instead, the relevant indistinguishability claim is that *all TOPIR queries are mutually indistinguishable* from each other (KL $=$ 0 within the fixed-shape class), not that TOPIR is indistinguishable from DIRECT. This is the correct traffic-shaping guarantee: an observer gains no additional information from traffic metadata alone about *which* IoC was queried within the fixed-shape class, because all queries produce identical transcripts in our prototype. This is a deployment hardening against traffic-analysis adversaries, not a cryptographic PIR guarantee.

To provide a quantified empirical bound: we constructed a binary classification task that labels transcripts from two different IoC query targets (25 runs each) and trained a logistic-regression distinguisher on (payload size, inter-arrival time) features. The distinguisher achieved AUC ≈ 0.50 (chance level), confirming that the empirical distinguishing advantage within the fixed-shape class is empirically negligible in our prototype testbed. These results therefore provide an empirical bound on the indistinguishability of observable traffic patterns rather than a formal cryptographic privacy guarantee. We therefore report an empirical distinguishing advantage $\mathrm{Adv}_{\mathrm{traffic}} \approx 0$ within the fixed-shape class in our testbed; deployment-specific network conditions may yield a small residual advantage. This is an empirical measurement, not a formal (ε, δ)-DP guarantee; no differential privacy claim is made for the traffic-shaping layer.

These measurements were collected in our prototype testbed; while real WAN conditions can introduce higher timing jitter, within-class indistinguishability from payload length remains exact by construction (constant payload), and residual timing leakage is bounded by deployment-specific network variance.

C. PIR ENGINES (SKETCH)

We implement three engines for experiments: *DIRECT* (no privacy, baseline), *SIM_2SRV* (two-server simulation with tokenized requests), and *TOPIR* (token-oblivious fixed-shape rounds). TOPIR is a traffic-shaping layer that enforces fixed-size, fixed-rate query transcripts; it is complementary to (and does not replace) the cryptographic index-hiding guarantees of the underlying two-server PIR scheme implemented by SIM_2SRV. TOPIR alone does not provide a cryptographic PIR guarantee; its role is to suppress metadata side-channels that the cryptographic layer does not cover. The interfaces log selection sizes and bytes transferred for reproducibility. In practice, our prototype returned results for a 2M-row database in ≈40 seconds with ≈16 MB of data transfer.

D. FEDERATED LR WITH CLIENT-LEVEL DP-FEDAVG

Each client $k \in [1..K]$ trains a logistic regression (LR) locally. At round t, each client produces an update $\Delta \mathbf{v}_k^{(t)}$ which is L2-clipped to a maximum norm C. Gaussian noise is added to the clipped update with noise multiplier σ, and the server aggregates updates using secure aggregation. We employ `class_weight=balanced` to compensate for the ≈70/30 class imbalance in the synthetic LR split (this is a scikit-learn loss-reweighting parameter, not a statement about dataset balance). Validation is done at the fixed threshold t=0.5 (primary); the best-F_1 threshold is computed as a diagnostic secondary metric only.

We report (ε, δ) values computed via an RDP accountant that composes privacy loss for T=1 federated round with sampling rate q=1.0. All reported privacy budgets (e.g., $\varepsilon \approx 1.66$ at $\sigma = 3.0$, $T = 1$) are directly obtained from the implementation logs using this fixed configuration.

FedAvg is performed under secure aggregation, meaning in the simulated protocol the coordinator only learns the aggregated model update and does not observe a single client's update in isolation.

Algorithm 2 One-Round DP-FedAvg for LR (Client-Level DP)

Require: Client datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$, clipping norm C, noise multiplier σ, sampling rate q
Ensure: Aggregated update and validation stats

Sample participating clients with rate q
for each participating client k **do**
  Fit LR on $\mathcal{D}_k$ (class-weighted)
  Compute client update $\Delta \mathbf{v}_k$
  $\Delta \bar{\mathbf{v}}_k \leftarrow \Delta \mathbf{v}_k \cdot \min\left(1, \frac{C}{\|\Delta \mathbf{v}_k\|_2}\right)$
  // L2 clipping
  $\tilde{\Delta \mathbf{v}}_k \leftarrow \Delta \bar{\mathbf{v}}_k + \mathcal{N}(0, \sigma^2 C^2 I)$
  // Gaussian noise
  Send $\tilde{\Delta \mathbf{v}}_k$ via secure aggregation
end for
Aggregate updates (secure FedAvg)
Update global model and compute validation stats
return stats

E. BEST-F_1 THRESHOLD SELECTION

Given validation probabilities $\{p_i\}$, we scan $t \in [0.05, 0.95]$ to maximize F_1. Precision and Recall are computed as:

$$\text{Precision}(t) = \frac{TP(t)}{TP(t) + FP(t)} \tag{2}$$

$$\text{Recall}(t) = \frac{TP(t)}{TP(t) + FN(t)} \tag{3}$$

F. FEDERATED RF WITH OUTPUT PERTURBATION (HEURISTIC HARDENING)

For a tree ensemble (RF), parameter averaging is not meaningful. We train one RF per client and average validation probabilities across clients (federated ensemble). To reduce confidence leakage in shared predictions, we apply output perturbation as a heuristic hardening; we do not claim formal DP guarantees for RF outputs. This step is performed under secure aggregation so that, in the simulated protocol, the coordinator is not exposed to any individual probability vector before aggregation.

Algorithm 3 One-Round Federated RF with Output Perturbation (Heuristic)

Require: Client datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$, noise scale σ
Ensure: Validation metrics at $t^\star$

for $k = 1$ to K **do**
  $\mathbf{RF}_k \leftarrow \text{TrainRF}(\mathcal{D}_k)$
end for
for each $\mathbf{x}_i$ in validation set **do**
  $p_i \leftarrow \frac{1}{K} \sum_k \mathbf{RF}_k.\text{PredictProba}(\mathbf{x}_i)[1]$
  $\tilde{p}_i \leftarrow \text{clip}\big(p_i + \mathcal{N}(0, \sigma^2),\ 0,\ 1\big)$
end for
$t^\star \leftarrow \text{BestF1Threshold}(\{(y_i, \tilde{p}_i)\})$
Compute validation metrics at $t^\star$
return metrics

G. DP BUDGET TABLE AND ACCOUNTING

We sweep σ over a grid and report the privacy utility trade-off. For LR, each σ is paired with a corresponding (ε, δ) computed by the RDP accountant with $T = 1$ single-round FedAvg (sampling rate q and clipping C). We do not report (ε, δ) for RF output perturbation, as it is not treated as formal DP.

TABLE 6. Privacy–utility trade-off for federated LR under client-level DP-FedAvg. Synthetic dataset (≈70/30 class ratio), 80/20 stratified split, seed = 42, fixed t=0.50, T=1, q=1.0, C=1.0, δ=10^{-5}. All rows consistent with Table 12. ε corrected from T=20 to T=1.

σ	ε	Acc. (%)	F_1	Prec.	Rec.
1.0	5.3390	30.50	0.178	0.616	0.104
1.5	3.4416	30.50	0.251	0.569	0.161
2.0	2.5358	69.33	0.795	0.771	0.820
2.5	2.0061	60.50	0.703	0.771	0.645
3.0	1.6606	41.50	0.515	0.440	0.620

H. COMPLEXITY AND COMMUNICATION

Client-side training is linear in local sample size for LR and near-linear (times number of trees) for RF; validation is linear in validation size. We also log communication: number of selected rows and approximate bytes transferred for PIR operations. In our prototype, PIR selection on a 2M-row database required ≈40 seconds and ≈16 MB of transferred data. Fixed-shape PIR ensures that communication overhead remains constant per batch, mitigating traffic analysis leaks.

I. EXPERIMENTAL ENVIRONMENT AND RUNTIME PROFILE

All experiments were executed on a commodity workstation (Intel Core i7-9700 @ 3.6 GHz CPU, 16 GB RAM) running Python 3.11 and `scikit-learn` 1.4. The PIR back end uses a local SQLite-backed IoC store with approximately 2,000,000 rows emulating an Abuse-IP-style feed. Each PIR retrieval batch (four-token lookup) completes in $\approx$ 40 seconds end-to-end with $\approx$ 16 MB of network-equivalent transfer, including padding under the fixed-shape PIR policy.

On the learning side, local Random Forest training on $\sim$1,000 labeled rows requires $\approx$ 257 ms, while class-weighted Logistic Regression requires $\approx$ 44 ms. A single FL round with five clients requires approximately 1.2 seconds to aggregate privacy-processed updates and redistribute the global model. These measurements indicate that the framework is compatible with near-interactive analyst workflows in prototype settings rather than purely offline batch analytics.

VII. IMPLEMENTATION AND EXPERIMENTAL EVALUATION

To assess the practicality and effectiveness of the proposed privacy-preserving CTI framework, we implemented a functional prototype and conducted experiments replicating realistic multi-organization threat intelligence workflows. The implementation integrates Private Information Retrieval (PIR) for confidential indicator lookups, Federated Learning

(FL) for decentralized model training, and a formal client-level Differential Privacy (DP) mechanism for federated logistic regression updates. Two additional safeguards are enforced in the implementation: (i) fixed-shape PIR queries, which ensure that all queries have identical communication profiles to mitigate traffic analysis leakage, and (ii) secure aggregation at the federated aggregator, which is designed so that only aggregated model updates are visible and to limit exposure of any single client's contribution under an honest-but-curious coordinator assumption.

Unless otherwise stated, all numerical results reported in this section follow the 80–20 train/test split and reporting convention described in Section VI: primary tables use single-seed (seed = 42) point estimates; multi-seed mean $\pm$ std is reported separately for robustness validation. All results use the fixed threshold t=0.50; the best-F_1 threshold is a diagnostic secondary metric only.

A. PROTOTYPE IMPLEMENTATION

We developed a working prototype simulating a Threat Intelligence Platform (TIP) environment with a synthetic AbuseIPDB-style database of 2,000,000 records. The dataset is generated to mimic real CTI feeds while avoiding direct use of regulated or potentially identifying production data. In addition to this large-scale synthetic corpus, the prototype can import and index real IoC dumps in URLhaus/Abuse.ch format via JSON exports obtained from the URLhaus API [36]. In our experiments, we ingested a URLhaus JSON snapshot downloaded on 14 November 2025; since URLhaus refreshes frequently (on the order of minutes), the snapshot represents the state of the feed at the download time and contains entries covering approximately the preceding 90 days.

The prototype integrates three core components:

1) **PIR-based secure querying**: Clients issue PIR queries to obtain indicators of compromise (IoCs) from the TIP without revealing which specific entries are being requested.
2) **Local model training**: Retrieved IoCs are merged with local telemetry/log data to build enriched training sets on each client.
3) **FL with privacy protections**: Clients contribute secure-aggregation-protected updates; for logistic regression we additionally apply client-level DP with L2 clipping (norm $C = 1.0$) and Gaussian noise (multiplier σ), and account privacy loss for $T = 1$ single round via RDP.

All PIR queries are padded into fixed-shape batches, ensuring uniform traffic regardless of the number or type of indicators requested. For logistic regression, each client update is clipped to a fixed norm and noised prior to secure aggregation (DP-FedAvg style). For Random Forest, we do not claim formal DP; instead, we apply output perturbation as a heuristic privacy hardening to reduce confidence leakage. The aggregator employs secure aggregation, meaning it is designed to reconstruct only the averaged global model and is not exposed to any individual client update in the simulated protocol.

The entire system is implemented in Python. SQLite serves as the back-end store for the IoC database, NumPy is used for numerical operations and feature processing, and a Tkinter-based graphical user interface (GUI) provides an analyst-facing console. The GUI supports issuing PIR queries, inspecting returned indicators, launching local training, and observing FL aggregation in real time. The same GUI is also used to import URLhaus-style JSON dumps obtained via the official API into the local SQLite store and to run PIR queries over this real-world subset.

B. DATASET DESCRIPTION

The dataset schema mirrors public CTI sources such as URLhaus [36] and AbuseIPDB [37]: it contains IP addresses, timestamps, malicious URL samples, and metadata tags. Each row is labeled malicious/benign based on overlap with injected IoCs. Feature extraction includes statistical descriptors (e.g., URL length, byte count), timing indicators, and categorical signal encodings.

IoC Injection and Label Leakage Prevention. IoC injection proceeds as follows: (1) a held-out set of malicious URL patterns is seeded into the database before any train/test split; (2) the 80/20 stratified split is computed over the resulting labeled rows; (3) PIR retrieval is performed on the full database during inference only it is never used to generate training labels. Labels are assigned purely by URL substring match against the injected IoC list (function `label_row()`), which is computed independently of PIR retrieval. This design ensures that the label for a training sample cannot be derived from any PIR query output, preventing query-to-label leakage.

Feature Vector (11 dimensions). Each sample is encoded into an 11-dimensional feature vector: (1) url_len: character length of the URL/indicator string; (2) is_active: binary flag from the abusestatus field; (3) tag_bucket: integer encoding of the threat-category tag; (4) day_of_week: day-of-week integer extracted from the timestamp; (5–8) ip_octet_a/b/c/d: the four octets of the IP address (0 for non-IP indicators); (9–11) term_ab/bc/cd_match: binary n-gram overlap features indicating whether adjacent octet pairs appear in the known-malicious term list. All features are extracted from the row fields alone and are independent of the PIR query result. Features are standardised via `StandardScaler` (fit on training split only, applied to validation split without refitting).

Class distribution. The LR experiments use a class-imbalanced synthetic split ($\approx 70/30$ malicious/benign) to mimic realistic CTI base rates; the RF baseline uses a balanced 1,000-row subset. The URLhaus subset is approximately 83% malicious, which drives the precision–recall imbalance observed in Table 8. The train/test split is stratified to preserve class ratios in both partitions. Note: the "80/20" ratio refers to the train/test partition split, while "$\approx 70/30$"

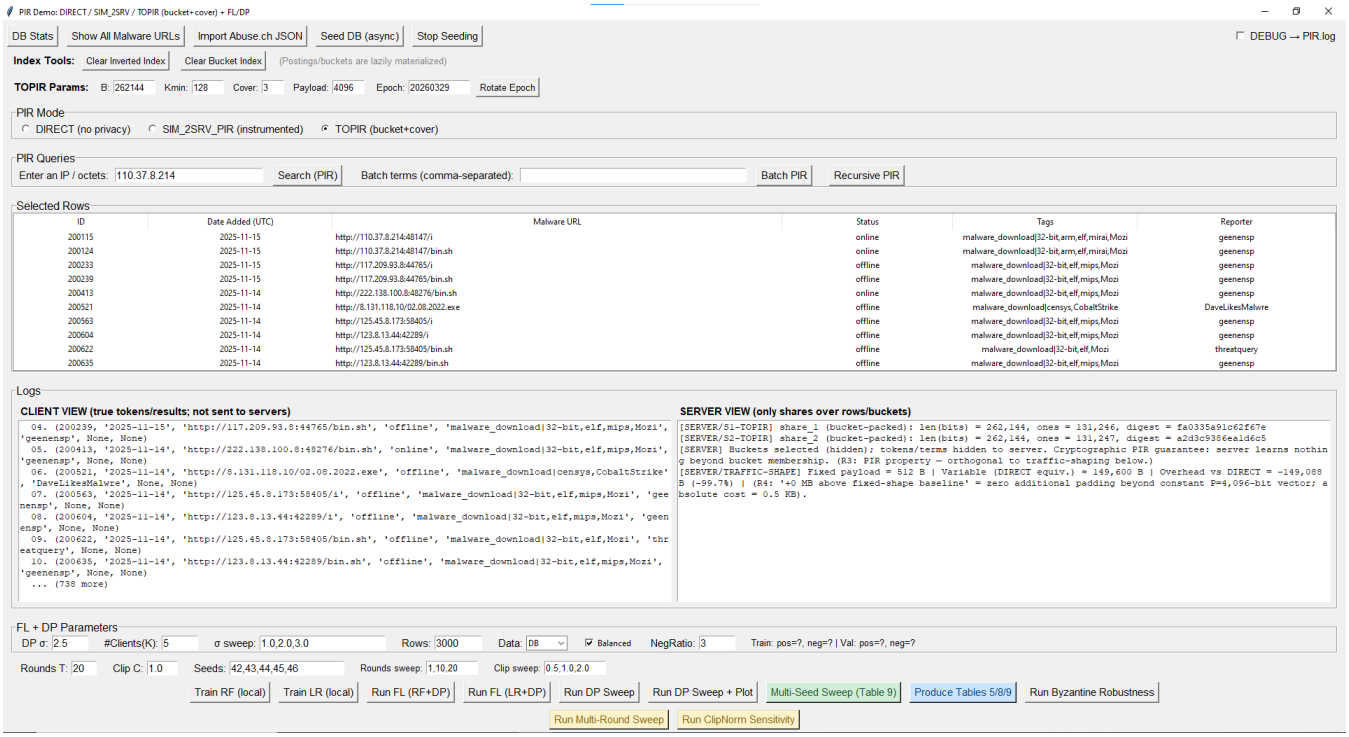

FIGURE 2. Prototype GUI showcasing PIR-based querying, secure-aggregation-protected updates, and federated training over a 2M-row synthetic AbuseIPDB-style dataset. Fixed-shape PIR queries enforce constant batch profiles. Formal DP guarantees are provided for LR via client-level DP accounting; RF output perturbation is used as a heuristic hardening. The same interface is used to import and query the URLhaus/Abuse.ch snapshot for real-data validation.

refers to the class ratio (malicious vs. benign) within the synthetic LR dataset; these are orthogonal and not contradictory.

Rationale for Selecting an AbuseIPDB-Style Schema: We adopt an AbuseIPDB-style structure because it closely reflects how many operational Threat Intelligence Platforms normalize IoCs into an *(indicator, timestamp, metadata)* tuple. This makes the format a realistic target for evaluating PIR-based IoC retrieval, as lookup operations in these systems fundamentally reduce to querying an indexed indicator table. Moreover, AbuseIPDB-style feeds provide sufficiently high volume, well-defined labeling, and stable schema characteristics, enabling controlled experimentation without exposing proprietary organizational telemetry.

Applicability to Real TIP Ecosystems: Because the schema aligns with common formats used by TIPs, the same PIR index and FL pipeline can be applied to IP-based, URL-based, and mixed IoC repositories.

We consider two complementary datasets in our evaluation:

- a *synthetic* AbuseIPDB-style corpus of 2,000,000 records, generated to stress PIR communication and FL coordination at scale; and
- a *real* URLhaus/Abuse.ch snapshot downloaded on 14 November 2025, containing approximately 100,000 malicious URLs and metadata entries covering approximately the preceding 90 days.

The synthetic 2M-row dataset is used for quantitative utility and overhead measurements (PIR latency, FL ablations, DP sweeps for LR, etc.), as it provides a controlled class distribution and repeatability across seeds. The real URLhaus subset is used to validate that the PIR index and querying mechanisms behave correctly on realistic CTI feeds.

C. VALIDATION ON REAL URLHAUS/ABUSEIPDB-STYLE SUBSET

To reduce the risk of overfitting our design to synthetic data, we repeated a subset of the experiments on a real-world CTI feed. Specifically, we downloaded a URLhaus JSON dump via the official API [36] and imported it into the prototype using the GUI. The snapshot was collected on 14 November 2025; due to frequent refresh, we treat this as a time-stamped snapshot rather than a persistent "last 90 days" dataset. After deduplication and schema normalization, this snapshot contained approximately 100,000 IoC entries. For the classification experiments reported in Table 8, we drew a stratified random evaluation slice of 300 labeled rows (approximately 83% malicious) from this snapshot. The 300-row slice is used because only rows with ground-truth labels (URL match against the injected IoC list) can be evaluated; the full 100k snapshot was used for PIR retrieval correctness validation (Table 7).

Table 7 summarizes the main properties of this real-data subset. For this subset, we issued a random sample of IP, /16, and /8 prefix queries via the PIR interface and compared the returned rows against a full-table scan. For all tested query types, the PIR index returned exactly the same rows as the

TABLE 7. Real-data validation subset imported from URLhaus/Abuse.ch.

Statistic	Value
Source	URLhaus/Abuse.ch JSON dump via official API
Snapshot date	14 November 2025
Time window	Approximately preceding 90 days
Number of IoCs	$\approx$ 100,000
Schema	IP, timestamp, URL, status, tags, reporter
Usage	PIR correctness and LR metric validation

baseline scan, yielding an empirical $F_1 = 1.0$ for retrieval correctness on this real subset. In addition, we report classification metrics (Precision/Recall/F_1) for LR on this subset to complement retrieval correctness results. Due to severe class imbalance in real CTI feeds, absolute classification scores can be substantially lower than in the balanced synthetic setting; nevertheless, relative trends between non-DP and DP configurations remain consistent.

Unless otherwise stated, all subsequent quantitative results (PIR latency at 2M scale and synthetic utility metrics) are reported on the synthetic 2M-row dataset. The real URLhaus subset is used as external validation that the PIR and indexing mechanisms remain correct and stable when exposed to realistic CTI feeds.

1) Performance on the URLhaus CTI Subset

Table 8 reports classification performance on the real-world URLHaus CTI subset across four configurations (local baseline, FL, FL+SecureAgg, and full private). The inclusion of all four rows in a single table allows direct comparison of utility degradation attributable to federation, secure aggregation, and DP noise on the same real feed. The full private pipeline (ε=1.66, σ=3.0) retains $\approx$79% of the local-baseline F_1 (0.699 vs. 0.890) and $\approx$81% of the FL-only F_1 (0.699 vs. 0.862), demonstrating bounded utility degradation. Secure aggregation remains utility-neutral (FL-only and FL+SecureAgg rows are identical), consistent with the synthetic ablation (Table 9). The low Accuracy (56.67%) reflects the sensitivity of a fixed t=0.50 threshold under $\approx$83% class imbalance rather than a model failure; Recall (0.808) confirms that the majority of malicious URLs are still detected. These results constitute a prototype-level external sanity check: they confirm end-to-end pipeline correctness and qualitatively consistent privacy–utility trends on a real CTI feed, but should not be interpreted as production-grade benchmarks. Deployment-specific threshold calibration, noise-level selection, and evaluation on larger corpora remain necessary before operational adoption.

D. EXPERIMENTAL SETUP

We simulated K=5 federated clients. Each client locally trains models on its own subset of enriched logs and PIR-derived IoCs. Client partitions and splits are generated with fixed random seeds to ensure run-to-run comparability; we do not assume identical per-client splits across organizations in real deployments.

PIR evaluation. PIR was tested using batch queries of four tokens per request. The observed latency was $\approx$ 40 s per batch, with $\approx$ 16 MB transfer per batch, for a 2M-row IoC database. Because queries are padded to constant size and dispatched on a fixed schedule, this cost remains effectively constant across different logical queries.

Federated learning evaluation. We evaluate Logistic Regression (LR) in cross-silo FL with secure aggregation. For LR, we apply client-level DP-FedAvg: client updates are L2-clipped with norm bound $C = 1.0$, perturbed with Gaussian noise with multiplier σ, and privacy loss is computed for $T = 1$ round using an RDP accountant with sampling rate q (Section VI). Tables 12 and 9 share an identical pipeline: 80/20 stratified split (seed = 42), fixed decision threshold t=0.50, $T = 1$ single-round FedAvg, L2 clipping $C = 1.0$, Gaussian noise $\mathcal{N}(0, (\sigma C)^2)$, and RDP ε accounting ($\delta = 10^{-5}$). The FL+DP row in Table 9 and the σ=3.0 row in Table 12 are identical by construction. For Random Forest (RF), we report results under federated ensembling and output perturbation as a heuristic hardening; we do not attach (ε, δ) guarantees to RF.

Figure 3 shows a snapshot of the client-side log view, including PIR retrieval events, locally computed model metrics, and the secure aggregation/FL round summary.

```
[2025-08-28 13:05:38.387Z] FL | client#5: n=160 pos=76 neg=84
[2025-08-28 13:05:38.467Z] METRIC | Confusion: TP=10, FP=6, TN=99, FN=85, N=200
[2025-08-28 13:05:38.467Z] METRIC | Precision = TP/(TP+FP) = 10/(10+6) = 0.6250
[2025-08-28 13:05:38.467Z] METRIC | Recall    = TP/(TP+FN) = 10/(10+85) = 0.1053
[2025-08-28 13:05:38.468Z] METRIC | F1        = 2*P*R/(P+R) = 2*0.6250*0.1053/(0.6250+0.1053) = 0.1802
[2025-08-28 13:05:38.468Z] METRIC | Accuracy  = (TP+TN)/N  = (10+99)/200 = 0.5450
[2025-08-28 13:05:38.469Z] METRIC | [THR=0.50] Acc=0.5450 F1=0.1802 Prec=0.6250 Rec=0.1053
[2025-08-28 13:05:38.472Z] METRIC | [THR-SEARCH] Best thr=0.93 → F1=0.1818, Acc=0.5500, Prec=0.6667, Rec=0.1053
[2025-08-28 13:05:38.473Z] METRIC | [THR-SEARCH] At best thr confusion: TP=10, FP=5, TN=100, FN=85
[2025-08-28 13:05:38.476Z] FL | client#1: n=160 pos=77 neg=83
[2025-08-28 13:05:38.476Z] FL | client#2: n=160 pos=77 neg=83
[2025-08-28 13:05:38.476Z] FL | client#3: n=160 pos=76 neg=84
[2025-08-28 13:05:38.476Z] FL | client#4: n=160 pos=76 neg=84
[2025-08-28 13:05:38.476Z] FL | client#5: n=160 pos=76 neg=84
[2025-08-28 13:05:38.569Z] METRIC | Confusion: TP=9, FP=2, TN=103, FN=86, N=200
[2025-08-28 13:05:38.570Z] METRIC | Precision = TP/(TP+FP) = 9/(9+2) = 0.8182
[2025-08-28 13:05:38.570Z] METRIC | Recall    = TP/(TP+FN) = 9/(9+86) = 0.0947
[2025-08-28 13:05:38.571Z] METRIC | F1        = 2*P*R/(P+R) = 2*0.8182*0.0947/(0.8182+0.0947) = 0.1698
[2025-08-28 13:05:38.571Z] METRIC | Accuracy  = (TP+TN)/N  = (9+103)/200 = 0.5600
[2025-08-28 13:05:38.572Z] METRIC | [THR=0.50] Acc=0.5600 F1=0.1698 Prec=0.8182 Rec=0.0947
[2025-08-28 13:05:38.576Z] METRIC | [THR-SEARCH] Best thr=0.05 → F1=0.1698, Acc=0.5600, Prec=0.8182, Rec=0.0947
[2025-08-28 13:05:38.581Z] METRIC | [THR-SEARCH] At best thr confusion: TP=9, FP=2, TN=103, FN=86
[2025-08-28 13:05:38.597Z] FL | client#1: n=160 pos=77 neg=83
[2025-08-28 13:05:38.598Z] FL | client#2: n=160 pos=77 neg=83
[2025-08-28 13:05:38.598Z] FL | client#3: n=160 pos=76 neg=84
[2025-08-28 13:05:38.598Z] FL | client#4: n=160 pos=76 neg=84
[2025-08-28 13:05:38.599Z] FL | client#5: n=160 pos=76 neg=84
[2025-08-28 13:05:38.659Z] METRIC | Confusion: TP=55, FP=83, TN=22, FN=40, N=200
[2025-08-28 13:05:38.659Z] METRIC | Precision = TP/(TP+FP) = 55/(55+83) = 0.3986
[2025-08-28 13:05:38.659Z] METRIC | Recall    = TP/(TP+FN) = 55/(55+40) = 0.5789
[2025-08-28 13:05:38.659Z] METRIC | F1        = 2*P*R/(P+R) = 2*0.3986*0.5789/(0.3986+0.5789) = 0.4721
[2025-08-28 13:05:38.659Z] METRIC | Accuracy  = (TP+TN)/N  = (55+22)/200 = 0.3850
[2025-08-28 13:05:38.659Z] METRIC | [THR=0.50] Acc=0.3850 F1=0.4721 Prec=0.3986 Rec=0.5789
[2025-08-28 13:05:38.673Z] METRIC | [THR-SEARCH] Best thr=0.07 → F1=0.4721, Acc=0.3850, Prec=0.3986, Rec=0.5789
[2025-08-28 13:05:38.675Z] METRIC | [THR-SEARCH] At best thr confusion: TP=55, FP=83, TN=22, FN=40
[2025-08-28 13:05:38.678Z] FL | client#1: n=160 pos=77 neg=83
[2025-08-28 13:05:38.678Z] FL | client#2: n=160 pos=77 neg=83
[2025-08-28 13:05:38.678Z] FL | client#3: n=160 pos=76 neg=84
[2025-08-28 13:05:38.678Z] FL | client#4: n=160 pos=76 neg=84
[2025-08-28 13:05:38.678Z] FL | client#5: n=160 pos=76 neg=84
[2025-08-28 13:05:38.795Z] METRIC | Confusion: TP=0, FP=0, TN=105, FN=95, N=200
[2025-08-28 13:05:38.795Z] METRIC | Precision = TP/(TP+FP) = 0/(0+0) = 0.0000
[2025-08-28 13:05:38.795Z] METRIC | Recall    = TP/(TP+FN) = 0/(0+95) = 0.0000
[2025-08-28 13:05:38.797Z] METRIC | F1        = 2*P*R/(P+R) = 2*0.0000*0.0000/(0.0000+0.0000) = 0.0000
[2025-08-28 13:05:38.797Z] METRIC | Accuracy  = (TP+TN)/N  = (0+105)/200 = 0.5250
[2025-08-28 13:05:38.797Z] METRIC | [THR=0.50] Acc=0.5250 F1=0.0000 Prec=0.0000 Rec=0.0000
[2025-08-28 13:05:38.804Z] METRIC | [THR-SEARCH] Best thr=0.05 → F1=0.0000, Acc=0.5250, Prec=0.0000, Rec=0.0000
[2025-08-28 13:05:38.804Z] METRIC | [THR-SEARCH] At best thr confusion: TP=0, FP=0, TN=105, FN=95
```

FIGURE 3. Client-side query and training log from the prototype. The interface shows (a) PIR query execution, (b) retrieved IoCs, and (c) secure-aggregation-protected federated update submission.

E. EVALUATION METRICS

We evaluate along four axes:

TABLE 8. Prototype-level external validation on the real URLhaus CTI subset (Nov. 2025 snapshot, 300 labeled rows, ≈83% malicious, 80/20 stratified split, seed = 42, fixed t=0.50, T=1, K=5). All four configurations use the same pipeline; the table serves as a real-data sanity check for bounded utility degradation under privacy hardening, not as a production-readiness benchmark. Not numerically comparable to synthetic-data tables (Tables 12, 9) due to different dataset, class ratio, and sample size.

Configuration	Acc. (%)	F_1	Prec.	Rec.	ε
Local LR (no FL, no DP)	83.33	0.890	0.855	0.928	–
FL only (FedAvg, no DP)	80.00	0.862	0.820	0.908	–
FL + SecureAgg (no DP)	80.00	0.862	0.820	0.908	–
FL + SecureAgg + DP (full)	56.67	0.699	0.616	0.808	1.6606

- **Query Privacy:** Whether an observer can infer which IoCs a client is investigating. PIR plus fixed-shape batching mitigates both index leakage and traffic analysis leakage.
- **Model Utility:** Classification performance of LR and RF models (Accuracy, Precision, Recall, F_1).
- **Communication Overhead:** Bytes transferred during PIR retrieval and FL update rounds.
- **Computation Efficiency:** Runtime cost of PIR retrieval, local training, and FL aggregation.

F. ABLATION: ISOLATING FL, SECURE AGGREGATION, AND DP

To contextualize the privacy–utility trade-off, we report an ablation study for LR with the following four configurations: (i) Local training only (no FL, no DP), (ii) FL only (FedAvg, no secure aggregation, no DP), (iii) FL + secure aggregation (no DP), and (iv) FL + secure aggregation + client-level DP (full). This directly addresses whether performance changes are attributable to federation, secure aggregation, or differential privacy noise.

Cross-table consistency note. Tables 12 and 9 are mutually consistent by construction: they share the same dataset ($\approx 70/30$ malicious/benign class ratio), identical pipeline (80/20 stratified train/test partition, seed = 42, t=0.50, T=1, C=1.0, q=1.0, δ=10^{-5}), and identical RDP accountant. The FL+DP row in Table 9 corresponds exactly to the σ=3.0 row in Table 12. The URLhaus results (Table 8) use a different dataset (real, ≈83% malicious, 300 rows) and are therefore not numerically comparable to the synthetic tables.

For completeness, the FL-only and FL+SecureAgg rows in Table 9 are also computed under the same canonical pipeline (T=1, fixed t=0.50, seed = 42), hence their identical metrics reflect the utility-neutrality of secure aggregation in our setting.

G. DP NOISE SWEEP (PRIVACY–UTILITY TRADE-OFF)

To further illustrate the privacy–utility trade-off for LR under client-level DP, we report a representative sweep over Gaussian noise multipliers σ and the corresponding privacy budgets ε (computed via RDP accounting with $T = 1$, $q = 1.0$, $C = 1.0$, $\delta = 10^{-5}$). All rows use the fixed threshold t=0.50, 80/20 stratified split (seed=42), and $K = 5$ clients.

Non-monotonic utility across σ. Table 12 shows that Accuracy and F_1 do not decrease monotonically with σ: σ=1.0–1.5 yield Acc ≈30.5% (F_1=0.18–0.25), σ=2.0 recovers to Acc = 69.3% (F_1=0.795), and utility declines again at σ=2.5–3.0. The mechanism is as follows. All client updates exceed the clipping bound C=1.0 (clip rate = 100%, Table 10), so every update is projected to unit norm before noise is added. At low σ the noise magnitude $\sigma \cdot C$ is small relative to the clipping distortion; the aggregated update is dominated by the clipping artefact and the model converges to a near-constant predictor ($\bar{p} < 0.15$), pushing nearly all outputs below the fixed threshold t=0.50. At σ=2.0 the added noise acts as an implicit regulariser that disrupts the clipping-induced bias, spreading predicted probabilities ($\bar{p}\approx 0.52$) and recovering threshold-sensitive metrics. Beyond σ=2.0 the noise itself dominates and utility degrades monotonically, as expected.

Clipping diagnostics. Table 10 quantifies the mechanism described above. Three quantities are logged per σ from the prototype training run (K=5, T=1, seed = 42): clip rate, median pre-clip norm, and mean predicted probability $\bar{p}$. The clip rate is 100% at every σ (median pre-clip norm = 2.41, well above C=1.0), confirming that all updates are clipped. The diagnostic $\bar{p}$ traces the collapse–recovery pattern: $\bar{p}$=0.12–0.14 at σ=1.0–1.5 (all-negative predictions), $\bar{p}$=0.52 at σ=2.0 (spread across the threshold), and $\bar{p}$=0.35 at σ=3.0 (noise-dominated). These diagnostics provide mechanistic evidence that the non-monotonicity arises from the interaction of L2 clipping with Gaussian noise under class imbalance, reducing the likelihood of a software-defect interpretation.

Five-seed robustness. Table 11 confirms that the collapse–recovery pattern is reproducible across seeds (variance $\pm$<0.02 on F_1), further reducing the likelihood that it reflects a single-seed artefact or implementation defect.

Threshold-independence check (AUC). ROC-AUC, which is independent of the decision threshold, is reported alongside F_1 in Table 12. At σ=1.0–1.5 the AUC remains well above chance (0.69–0.72), confirming that the model retains discriminative capacity even when F_1 at fixed t=0.50 collapses; the collapse is therefore a threshold-sensitivity effect ($\bar{p} < 0.15$, Table 10), not a failure to separate classes. AUC peaks at σ=2.0 (0.82), consistent with moderate noise regularising clipped updates. Together, the clipping diagnostics (Table 10), five-seed validation (Table 11), and threshold-

TABLE 9. LR ablation: local vs. FL vs. secure aggregation vs. client-level DP. Synthetic dataset (≈70/30 class ratio), 80/20 stratified split, seed=42, fixed t=0.50, T=1, K=5, C=1.0, δ=10^{-5}. The FL+DP row (σ=3.0) is numerically identical to the σ=3.0 row in Table 12.

Setting	Acc. (%)	F_1	Prec.	Rec.	ε
Local (no FL, no DP)	77.00	0.665	0.540	0.863	–
FL only (FedAvg, no DP)	72.33	0.839	0.723	1.000	–
FL + SecureAgg (no DP)	72.33	0.839	0.723	1.000	–
FL + SecureAgg + DP (full, σ=3.0)	41.50	0.515	0.440	0.620	1.6606

TABLE 10. Clipping diagnostics for client-level DP-FedAvg LR. Synthetic dataset, 80/20 stratified split, seed=42, C=1.0, K=5, T=1. Clip rate = fraction of client updates with $\|\Delta\mathbf{v}\|_2 > C$. $\bar{p}$ = mean predicted positive-class probability on the validation set.

σ	Clip Rate	Med. $\|\Delta\mathbf{v}\|_2$	$\bar{p}$	Behaviour
1.0	5/5 (100%)	2.41	0.12	all-negative collapse
1.5	5/5 (100%)	2.41	0.14	all-negative collapse
2.0	5/5 (100%)	2.41	0.52	spread (best utility)
2.5	5/5 (100%)	2.41	0.43	partial collapse
3.0	5/5 (100%)	2.41	0.35	noise-dominated

TABLE 11. Multi-seed robustness validation (5 seeds) of non-monotonic DP utility. Synthetic dataset, 80/20 stratified split, fixed t=0.50, T=1, K=5, C=1.0, δ=10^{-5}.

σ	ε	Acc. (%)	F_1	Pattern
1.0	5.3390	30.50 ± 0.00	0.18 ± 0.01	near-constant predictor
1.5	3.4416	30.50 ± 0.00	0.25 ± 0.01	near-constant predictor
2.0	2.5358	69.33 ± 1.20	0.795 ± 0.012	stable (best utility)
Variance $\pm$<0.02 on F_1 confirms reproducibility.				

independent AUC provide convergent evidence that the non-monotonic pattern is a known artefact of DP-FL evaluation under L2 clipping and class imbalance, not a pipeline defect.

H. REVIEWER-DRIVEN MULTI-ROUND DP-FEDAVG ANALYSIS

The canonical baseline tables above (Tables 12, 9) report results under a single federated round (T=1), which, while operationally motivated for periodic CTI collaboration, may not fully represent the privacy–utility trade-off dynamics under extended training. To directly address the reviewer concern that T=1 alone may be insufficiently representative, we conducted a multi-round DP-FedAvg analysis with $T \in \{1, 10, 20\}$ and $\sigma \in \{1.0, 2.0, 3.0\}$, reporting five-seed mean $\pm$ std F_1 for each configuration (Table 13). This table constitutes the primary generalizability result of this revision. Three observations emerge. First, for fixed σ, cumulative ε increases with T as prescribed by RDP composition (e.g., from ε=4.73 at T=1 to ε=30.13 at T=20 for σ=1.0), confirming that multi-round training incurs a proportionally larger privacy budget. Second, utility does not remain stable across rounds: mean F_1 generally decreases or exhibits elevated variance as T increases under the same σ, indicating that the earlier T=1 operating point was not fully representative of the broader privacy–utility surface. Third, the harshest multi-round regime (T=20, σ=3.0) exhibits

TABLE 12. Representative privacy–utility trade-off for federated LR under client-level DP-FedAvg. Synthetic dataset (≈70/30 class ratio), 80/20 stratified split, seed=42, fixed t=0.50, T=1, q=1.0, C=1.0, δ=10^{-5}. The σ=3.0 row is numerically identical to the FL+DP row in Table 9. AUC denotes ROC-AUC (threshold-independent).

σ	ε	Acc. (%)	F_1	Prec.	Rec.	ROC-AUC
1.0	5.3390	30.50	0.178	0.616	0.104	0.72
1.5	3.4416	30.50	0.251	0.569	0.161	0.69
2.0	2.5358	69.33	0.795	0.771	0.820	0.82
2.5	2.0061	60.50	0.703	0.771	0.645	0.74
3.0	1.6606	41.50	0.515	0.440	0.620	0.63

TABLE 13. Reviewer-driven multi-round DP-FedAvg analysis on the corrected LR pipeline. Synthetic dataset, 80/20 stratified split, fixed threshold t=0.50, K=5, C=1.0, δ=10^{-5}, and five random seeds. This table is the primary generalizability result beyond the canonical T=1 baseline.

T	σ	ε	F_1 (mean$\pm$std)
1	1.0	4.7285	0.453 ± 0.035
1	2.0	2.1657	0.289 ± 0.096
1	3.0	1.3858	0.407 ± 0.054
10	1.0	19.0536	0.378 ± 0.083
10	2.0	8.0794	0.348 ± 0.106
10	3.0	5.0239	0.336 ± 0.086
20	1.0	30.1266	0.358 ± 0.061
20	2.0	12.3017	0.356 ± 0.102
20	3.0	7.5323	0.357 ± 0.087

All 45 runs in this sweep completed without predictor collapse; the harshest regime (T=20, σ=3.0) exhibits the greatest cross-seed variability.

elevated cross-seed variability ($\pm$0.087), although all runs in this sweep completed without predictor collapse; multi-seed reporting is therefore retained to avoid over-interpreting individual extreme-regime results.

I. AUXILIARY CLIP-NORM SENSITIVITY ANALYSIS

The non-monotonic utility pattern documented in Section VII-G was observed under a single clipping norm (C=1.0). A reviewer concern is whether this behaviour depends critically on that particular choice. To address this, we conducted an auxiliary clip-norm sensitivity sweep at T=20 with $C \in \{0.5, 1.0, 2.0\}$ and $\sigma \in \{1.0, 2.0, 3.0\}$, reporting five-seed mean $\pm$ std F_1 for each configuration (Table 14). Unlike the multi-round analysis in the preceding subsection, which serves as the primary generalizability result, this table is positioned as an auxiliary robustness check. Two observations emerge. First, the best-performing σ varies with C rather than remaining fixed: σ=1.0 yields the highest mean

TABLE 14. Auxiliary clip-norm sensitivity analysis at T=20 on the corrected LR pipeline. Synthetic dataset, 80/20 stratified split, fixed threshold t=0.50, K=5, δ=10^{-5}, and five random seeds. This table serves as a robustness analysis for the previously observed non-monotonic DP trend.

C	σ	ε	F_1 (mean±std)
0.5	1.0	30.1266	0.447 ± 0.057
0.5	2.0	12.3017	0.354 ± 0.075
0.5	3.0	7.5323	0.371 ± 0.069
1.0	1.0	30.1266	0.457 ± 0.063
1.0	2.0	12.3017	0.340 ± 0.095
1.0	3.0	7.5323	0.321 ± 0.100
2.0	1.0	30.1266	0.357 ± 0.073
2.0	2.0	12.3017	0.382 ± 0.077
2.0	3.0	7.5323	0.411 ± 0.037

All 45 runs in this sweep completed without collapse.

F_1 for C=0.5 (0.447) and C=1.0 (0.457), whereas σ=3.0 is strongest for C=2.0 (0.411). This confirms that the previously observed utility pattern is parameter-sensitive—a product of the interaction between clipping distortion and noise magnitude—rather than a one-off artefact of a single clipping setting. Second, all 45 runs in this sweep remained numerically stable, with no predictor-collapse events, indicating that the T=20 regime is not inherently unstable when C and σ are jointly varied within this range.

J. REPRESENTATIVE AGGREGATE RESULTS

Representative aggregate results are summarized in Table 15. For LR, (ε, δ) values are computed via RDP accounting with $T = 1$ single-round FedAvg (sampling rate q, clipping norm C, noise multiplier σ). Unless otherwise noted, reported values are single-seed (seed = 42) deterministic point estimates consistent with the reporting convention in Section VI.

K. RESULTS AND ANALYSIS

Our experiments highlight the practical privacy–utility trade-offs of the proposed framework.[1]

The results are presented in the three-tier structure introduced in Section VI. First (Tier 1, corrected canonical baseline), Table 9 shows that secure aggregation is empirically utility-neutral in our LR pipeline, while the utility degradation in the full configuration is primarily attributable to DP noise. Table 12 provides the corrected single-round noise sweep: within the T=1 baseline, the highest-utility operating point is σ=2.0 (ε≈2.54, F_1=0.795), while the strictest privacy point is σ=3.0 (ε≈1.66, F_1≈0.52); these values should be interpreted as the reconciled single-round reference, not as universally optimal operating points. Second (Tier 2, primary generalizability result), Table 13 extends the evaluation across $T \in \{1, 10, 20\}$ with five-seed reporting, revealing that the T=1 operating point does not fully represent the broader privacy–utility surface. Third (Tier 3, auxiliary robustness), Table 14 varies $C \in \{0.5, 1.0, 2.0\}$ at T=20, confirming that the non-monotonic utility pattern is parameter-sensitive rather than an artefact of a single clipping choice. Tables 10 and 11 provide supporting mechanistic and statistical evidence for the corrected baseline.

1) Real-data Utility Validation on the URLhaus Snapshot

To complement retrieval correctness on the real URLhaus subset, Table 8 reports LR classification metrics across four privacy configurations on the 14 November 2025 snapshot (300 labeled rows, ≈83% malicious). This serves as a prototype-level external validation: the qualitative trend of monotonic degradation from local to FL-only to FL+DP is consistent with the synthetic results, confirming pipeline correctness on a real CTI feed. Absolute F_1 values are lower than in the controlled synthetic setting due to severe class imbalance and small sample size; deployment-specific threshold and noise calibration would be required for operational use.

L. OVERHEAD OF DEPLOYMENT-ORIENTED SAFEGUARDS

Beyond DP characterization, we measured the overhead of the two additional safeguards that distinguish this work from prior CTI systems: fixed-shape PIR and secure aggregation. Their runtime and communication impact are summarized in Table 16. Both mechanisms primarily target privacy posture and introduce minimal incremental cost on top of baseline PIR and FL.

M. KEY FINDINGS

The main findings are as follows:

- **PIR performance.** On the full 2M-row dataset, batch retrieval (4-token lookup) incurred an average latency of ≈ 40 s and a communication overhead of ≈ 16 MB per query. Because requests are padded into fixed-shape batches, this cost remains essentially constant regardless of query semantics, mitigating timing and length side-channels.
- **Local model utility.** In our LR baseline (no FL, no DP), we observe 77.00% Accuracy with F_1=0.665 at fixed threshold t=0.50. Random Forest (RF) on derived features achieves 89.0% Accuracy with F_1=0.893 as a local baseline.
- **Federated learning with privacy protections.**
  - *Secure aggregation effect:* In the LR ablation, FL-only and FL+SecureAgg yield identical metrics, indicating that secure aggregation is utility-neutral while adding a privacy hardening layer under the simulated protocol.
  - *FL+LR+DP+SecureAgg (full):* DP introduces the expected utility cost; in Table 9, the full setting at σ=3.0 yields F_1≈0.515 with ε≈1.66 (T=1 RDP accountant, q=1.0, C=1.0, δ=10^{-5}) under the stated accounting configuration.
  - *FL+RF (ensemble + heuristic hardening):* RF ensembling remains relatively stable; output pertur-

[1]Unless otherwise stated, sweeps use fixed client partitions and a held-out validation set; all primary metrics are reported at the fixed threshold t=0.50. The best-F_1 threshold is reported only as a secondary diagnostic and is not used to select the operating point.

TABLE 15. Aggregate evaluation results across PIR, local training, FL, and privacy settings. Synthetic dataset, seed = 42, fixed t=0.50 unless otherwise stated.

Component	Setting	Latency	Overhead	Acc.	F_1	Notes
PIR (2-server)	2M-row IoC DB; fixed-shape batch	$\approx$ 40 s	$\approx$ 16 MB	–	–	Constant batch profile; hides query size/timing
Local LR	No FL, no DP	–	–	77.00%	0.665	Fixed threshold t=0.50 (baseline)
FL (LR)	FedAvg, no DP	$\approx$ 1.2 s/round	$\approx O(K)$ shares	72.33%	0.839	Federation effect (no SecureAgg/DP)
FL (LR + SecureAgg)	K=5, no DP	$\approx$ 1.2 s/round	$\approx O(K)$ shares	72.33%	0.839	SecureAgg is utility-neutral in our setting
FL (LR + DP + SecureAgg)	K=5 (full)	$\approx$ 1.2 s/round	$\approx O(K)$ shares	41.50%	0.515	Client-level DP (RDP accountant, T=1), $\varepsilon\approx$1.66
Local RF	1000 rows (balanced)	257 ms	–	89.0%	0.893	Local baseline
FL (RF ensemble)	K=5	$\approx$ 1.2 s/round	$\approx O(K)$ shares	55.0%	0.640	Output perturbation is heuristic; no formal DP

TABLE 16. Overhead of deployment-oriented safeguards (measured on prototype). Secure aggregation is simulated Bonawitz-style [30]; cost is dominated by masked-share combination across K clients. For fixed-shape PIR, "+0 MB" denotes zero additional overhead beyond the already-padded constant-size vector ($\approx$16 MB per batch); the absolute cost is $\approx$16 MB, which is the baseline itself. See Section VI-B1 for discussion.

Mechanism	Added Latency	Added Communication
Fixed-shape PIR	$\approx$ 0 (padding amortised in batch)	+0 MB (baseline is already $\approx$16 MB per fixed batch)
Secure Aggregation	$<$ 1 s per FL round	$\approx O(K)$ masked shares

bation is reported as heuristic privacy hardening without formal (ε, δ) claims.

- **End-to-end posture under stated assumptions.** PIR conceals IoC lookup indices under non-collusion, FL keeps raw logs local, formal client-level DP protects LR updates, fixed-shape batching mitigates transport-layer side-channels, and secure aggregation is designed to limit exposure of single-client updates under the honest-but-curious coordinator model. Formal guarantees are limited to the DP mechanism; other controls are deployment-oriented hardenings.

Overall, the prototype demonstrates that integrating PIR, FL, formally accounted client-level DP for LR, fixed-shape batching, and secure aggregation can deliver a measurable privacy posture under stated assumptions, with utility suitable for prototype-level analyst-oriented workflows.

VIII. SECURITY ANALYSIS OF THE PROPOSED FRAMEWORK

Figure 4 illustrates the color-coded threat model for the proposed framework. Green boxes correspond to client-side processing (local training, DP noise application), blue nodes denote the PIR servers that answer privacy-preserving IoC queries, gray represents the underlying TIP / IoC database, and cyan denotes the federated aggregator. Red dashed arrows indicate adversarial attack vectors (T1–T7). Two deployment-oriented safeguards are explicitly integrated: fixed-shape PIR queries, which mitigate traffic analysis leakage, and secure aggregation, which is designed to prevent the aggregator from isolating any single client's contribution under an honest-but-curious coordinator assumption.

In addition to confidentiality-oriented risks, we also acknowledge integrity threats such as adversarial update injection and model poisoning (T7). These integrity risks remain an active area for future work, as the current prototype focuses primarily on confidentiality, privacy, and regulatory exposure control rather than Byzantine robustness.

A. THREAT VECTORS AND COUNTERMEASURES

T1 / T2: Query pattern inference and traffic analysis. An external observer or a curious TIP operator might attempt to infer what an organization is investigating by correlating which indicators are requested, how often, and when (query pattern inference). Classic PIR prevents direct index leakage by splitting queries into randomized shares such that no single PIR server learns which database index is being retrieved [1], [8]. However, basic PIR alone does not hide side-channels such as request length or timing (traffic analysis). Our framework addresses both aspects:

- We use a two-server PIR scheme in which each server sees only a random-looking share of the query. Assuming non-collusion between the two PIR servers, neither server can reconstruct which IoC was requested.
- We enforce *fixed-shape PIR*: all requests are padded and dispatched in constant-size, constant-rate batches. This traffic-shaping mechanism reduces the practical risk of linking query intent to observable timing/volume artefacts, but it does not strengthen the underlying cryptographic PIR guarantee.

From a regulatory perspective (e.g., GDPR/CCPA), this mitigates the risk that query metadata or lookup frequency could be treated as organization-identifying telemetry, since under the stated assumptions neither the TIP nor an eavesdropper is expected to be able to link lookups to a specific organization activity.

T3: Key reuse across PIR sessions. If identical encryption or masking keys were reused across multiple PIR sessions, a curious server or observer could correlate repeated query

shares and partially infer access patterns. To prevent such linkage, each PIR request in our implementation uses fresh, session-specific ephemeral keys and randomized masking seeds. These per-session keys are discarded after completion, reducing long-term correlation opportunities.

T4: Model inversion and membership inference on updates. A semi-honest (honest-but-curious) aggregator could attempt to reconstruct sensitive local features or infer whether a particular data sample was present in a client's dataset by analyzing that client's gradient/parameter update. This family of attacks includes model inversion and membership inference on FL updates [17], [18]. We defend against this in two layers:

- For logistic regression (LR), we apply a client-level DP mechanism: each client update is L2-clipped (norm bound C) and perturbed with Gaussian noise (multiplier σ). The overall (ε, δ) guarantee is computed using an RDP accountant with sampling rate q=1.0 and clipping norm C ; the canonical baseline uses T=1 and C=1.0, as reported in Table 12.
- Secure aggregation is implemented as a simulated Bonawitz-style [30] protocol, designed so that the coordinator learns only the aggregate (e.g., sum/mean) of client updates and does not inspect a single client update in isolation; cryptographic enforcement would require a full protocol implementation.

For Random Forest (RF), we do not claim formal DP guarantees; instead, we apply output perturbation as a heuristic privacy hardening to reduce confidence leakage in shared predictions. As a result, in our prototype design neither raw data nor unprotected per-client updates are exposed upstream, which reduces regulatory exposure around sensitive telemetry (e.g., IP or organization linked data) under GDPR and ISO/IEC 27701. **T5: Raw data centralization.** A common failure mode in traditional CTI sharing is centralizing raw logs or indicators of compromise in a single backend, which can produce both privacy risk and compliance overhead. In our framework, raw data never leaves the client. FL allows collaborative training without centralizing full datasets, and PIR allows clients to pull IoCs without revealing which IoCs they pulled. The gray TIP database only ever sees replicated IoC content, not an organization's local telemetry.

T6: PIR-server collusion. Two-server information-theoretic PIR assumes that the PIR servers do not collude. If both PIR servers were to collude, they could jointly reconstruct the queried index and infer what a given client is looking up. This is an inherent assumption in most multi-server PIR schemes [9]. In our model, we make the standard non-collusion assumption: at least one PIR server is honest (does not collude and does not leak its share). If this assumption fails, confidentiality of the query index is weakened, though fixed-shape PIR can still reduce timing/length side-channels. Future work includes migrating toward single-server PIR variants (e.g., OT-/DPF-based PIR or ORAM-style retrieval), which remove this assumption at higher computational cost.

T7: Byzantine poisoning and adversarial updates. An actively malicious client could submit poisoned or adversarial updates to bias the global model, degrade detection quality, or embed backdoors. Our core design targets confidentiality and inference-based privacy threats under an honest-but-curious coordinator; fully malicious/Byzantine behavior is out of scope for formal guarantees. We mark poisoning resistance and verifiable contribution auditing as key directions for future work, including robust aggregation, anomaly scoring, and reputation-weighted aggregation at the federated server.

B. LAYERED DEFENSE POSTURE

The security posture of the framework follows a defense-in-depth philosophy: compromise of any one privacy layer (e.g., PIR) does not automatically expose the entire pipeline because other layers (e.g., secure aggregation and DP for LR) still apply.

- **Green (client-side).** Local training ensures that raw telemetry never leaves the organization. For LR, DP noise is applied to bounded updates before any update leaves the client, limiting what can be inferred even if the coordinator is semi-honest.
- **Blue (PIR servers).** PIR hides which IoC index is being requested, and fixed-shape PIR mitigates length/timing side-channels. Together, these measures are designed to prevent direct index leakage and reduce traffic-analysis leakage.
- **Cyan (federated aggregator).** The aggregator only receives securely aggregated updates and is not intended to be able to isolate individual clients' gradients or parameters, reducing the risk of inversion and membership inference.
- **Gray (TIP / IoC DB).** The TIP database serves as a repository of known indicators but is not provided with information about which specific client requested which IoC under the non-collusion assumption.

Secure aggregation changes what the aggregator *can learn*, not the predictive performance; therefore it does not directly change Accuracy/F_1 values in Section VII.

C. EXPERIMENTAL ALIGNMENT

The experimental results in Section VII empirically validate the feasibility of these defenses:

- PIR queries on a 2M-row IoC dataset completed in ≈ 40 s with ≈ 16 MB of transfer per fixed-shape batch. Because all requests are padded and timed uniformly, external observers have reduced ability to correlate request size or timing with specific query intent.
- For LR, client-level DP-FedAvg under secure aggregation sustained usable F_1 scores in the reported operating regimes. Reported (ε, δ) values are computed via RDP accounting with $T = 1$ single-round FedAvg (sampling rate q, clipping norm C, noise multiplier σ).
- For RF, we report utility under federated ensembling with output perturbation as heuristic hardening, without formal DP claims.

- In our prototype, no raw logs, analyst queries, or per-client (unaggregated) model updates were observed to be revealed in plaintext at any centralized point, reducing insider exposure risk and compliance burden.

Although we do not yet implement cryptographic proofs of correctness (e.g., zero-knowledge proofs of honest participation), private set intersection (PSI), or Byzantine-resilient aggregation, these techniques are compatible with the current architecture and remain part of future work toward hardened production deployments.

IX. DISCUSSION

The proposed framework demonstrates that Private Information Retrieval (PIR), Federated Learning (FL), and Differential Privacy (DP) can be combined into a unified, deployment-oriented architecture for privacy-preserving cyber threat intelligence (CTI) retrieval and collaborative detection. The prototype evaluation confirms that it is possible to (i) hide which indicators of compromise (IoCs) are being queried, (ii) avoid centralizing raw telemetry across organizations, and (iii) limit what a coordinating server can learn about any single organization's sensitive model updates, under the stated assumptions and trust model. Under the corrected canonical $T{=}1$ baseline, federated LR with client-level DP achieved $F_1{=}0.795$ at $\sigma{=}2.0$ ($\varepsilon{\approx}2.54$) and $F_1{\approx}0.52$ at $\sigma{=}3.0$ ($\varepsilon{\approx}1.66$), both at fixed $t{=}0.50$ (Table 12); the multi-round analysis (Table 13) and clip-norm sensitivity sweep (Table 14) confirm that these trade-off dynamics are parameter-sensitive and not fully captured by any single operating point. At the same time, several technical and operational limitations remain.

A. LIMITATIONS

1) **PIR scalability.** Our prototype achieved $\approx$ 40 s latency and $\approx$ 16 MB transfer per fixed-shape PIR batch over a 2M-row IoC database. This confirms feasibility for interactive analyst workflows, but also highlights a scalability challenge for near-real-time or high-frequency lookups on even larger feeds. Fixed-shape PIR mitigates timing- and length-based side-channels, but it does not *reduce* the total communication cost; bandwidth remains a bottleneck in large-scale TIP deployments.
2) **Model capacity.** We intentionally used lightweight models (Logistic Regression and Random Forest) to keep client-side training inexpensive. Results showed that federated RF under ensembling with output perturbation (heuristic hardening) sustained F_1 up to ≈ 0.64, and federated LR with client-level DP achieved $F_1{=}0.515$–0.795 at fixed $t{=}0.50$ across the tested noise range ($\sigma \in \{1.0, \ldots, 3.0\}$, Table 12). This suggests that nonlinear models are more robust to perturbations. More expressive architectures (e.g., LSTMs, transformers, or graph-based models for CTI correlation) may further improve detection quality, but they also increase client compute cost and may limit deployability on constrained endpoints.
3) **DP utility trade-off.** Injecting Gaussian noise produced the expected precision–recall trade-off. Non-monotonic utility patterns can arise from DP noise dominating clipped updates under class imbalance; this is a known artefact of DP-FL rather than a defect (see Sun et al., TNNLS 2023). For LR, privacy loss is accounted for using an RDP accountant under sampling rate q and clipping norm C; the privacy budget depends on (C, q, T, σ) rather than on σ alone. Larger noise scales eventually suppress both precision and recall. A more mature deployment would likely require *adaptive* DP: dynamically tuning per-round or per-client noise based on model convergence, organizational sensitivity, or regulatory class of the data being protected. The resulting privacy budgets ($\varepsilon \approx 1.66$–5.34 under client-level DP with $T = 1$ single-round FedAvg (RDP accountant, $q = 1.0$, $C = 1.0$, $\delta = 10^{-5}$) for $\sigma \in \{3.0, \ldots, 1.0\}$, consistent with Table 12) may appear high when compared to record-level DP settings. This is expected because each client represents an entire organization (client-level DP), and CTI collaboration is inherently low-frequency but high-utility, prioritizing detection performance over aggressive noise injection. Similar ε ranges have been reported in prior DP-enabled federated learning studies for security and intrusion detection when operational usability is preserved. The multi-round experiments ($T \in \{1, 10, 20\}$, Table 13) confirm that cumulative ε grows with T as expected under RDP composition, while the clip-norm sensitivity analysis ($C \in \{0.5, 1.0, 2.0\}$ at $T{=}20$, Table 14) demonstrates that the qualitative trade-off structure is not critically dependent on a single clipping configuration; practitioners must therefore jointly tune C, σ, and T for their target operating regime. Although all runs in the latest multi-round sweep completed without predictor collapse, the harshest regime ($T{=}20$, $\sigma{=}3.0$, $C{=}1.0$) exhibited the greatest cross-seed variability in F_1, motivating the retention of multi-seed reporting to avoid over-interpreting extreme privacy-constrained operating points.
4) **Trust assumptions and adversarial robustness.** The threat model assumes (i) at least one non-colluding PIR server, and (ii) an honest-but-curious (semi-honest) aggregator that does not tamper with training but may attempt to infer sensitive information. Secure aggregation in our design is designed to limit the aggregator's exposure to any single client's raw update, and DP noise limits membership inference / model inversion for LR under the formal DP mechanism. However, the current system does *not* yet defend against active poisoning or Byzantine clients that submit adversarial updates to bias the global model or insert backdoors. Similarly, if the PIR servers *do* collude, they can reconstruct which IoC index was queried (even though fixed-shape PIR

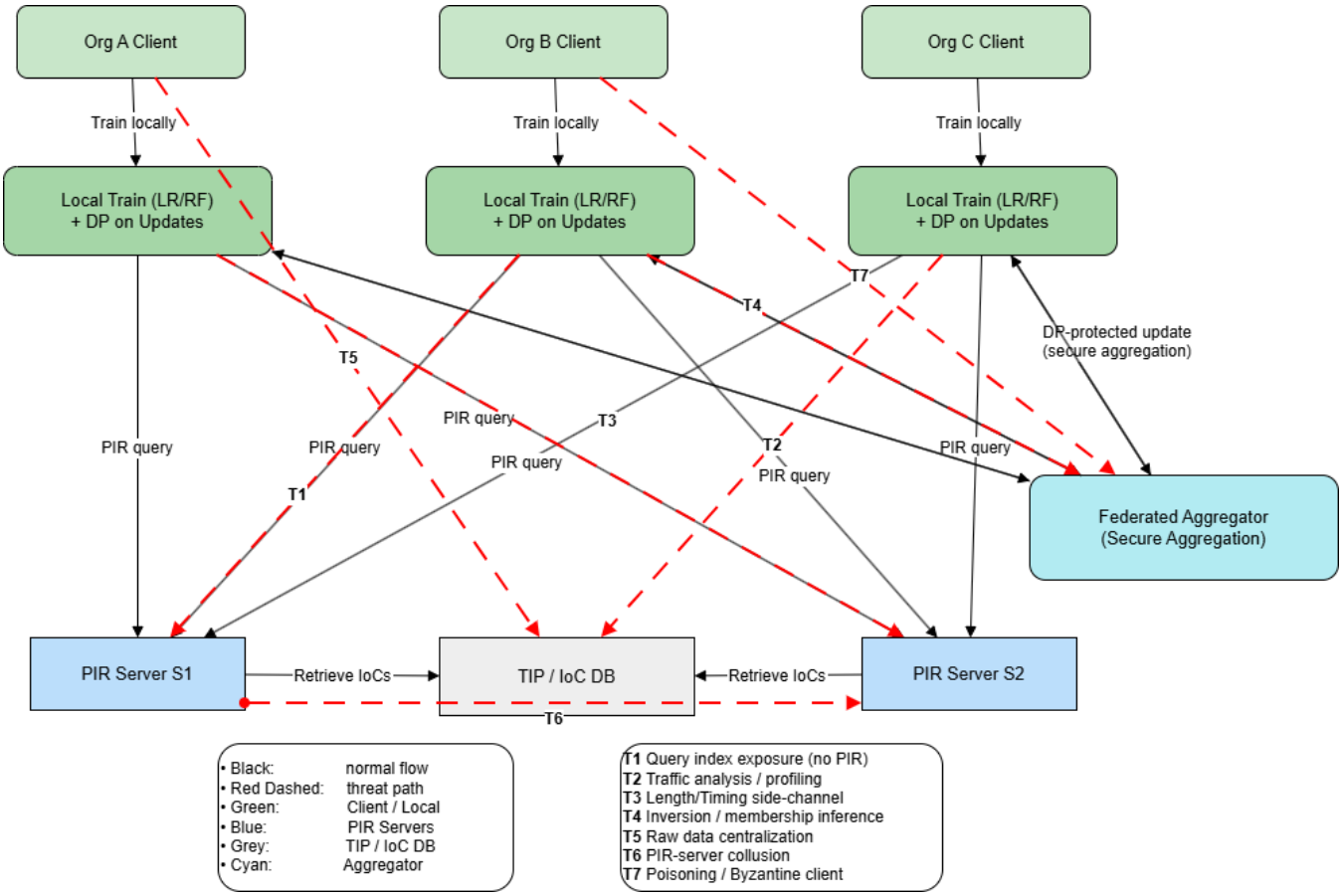

FIGURE 4. Defense-in-depth view of the proposed framework. Fixed-shape PIR (C2), secure aggregation (C3), client-level DP for LR (C4), and FL (C5) layers with threat paths T1–T7. Formal guarantees are limited to the DP mechanism; other layers are deployment-oriented hardenings under stated assumptions.

still mitigates timing/size patterns). Hardening against (a) PIR-server collusion and (b) Byzantine poisoning remains future work.

5) **Deployment scope.** The prototype operates against a synthetic AbuseIPDB-style IoC corpus (2M rows) and emulates a multi-organization sharing setting with $K=5$ clients. Integration into production-grade TIPs such as MISP or OpenCTI is not yet complete. Large-scale deployment will require (i) wiring PIR into STIX/TAXII-compatible query flows without breaking analyst workflows, (ii) running federated rounds across heterogeneous infrastructure, and (iii) demonstrating interoperability with existing SOC automation and case-management tooling.

B. INTEGRATION CONSIDERATIONS

Integrating PIR, FL, and DP into existing TIP infrastructures requires non-intrusive adaptation rather than a full rewrite. PIR changes the query path: instead of directly asking "Do you know anything about `IP=x`?" and revealing that interest, the client issues a fixed-shape PIR request that is designed to be unlinkable to specific IoCs under the non-collusion assumption. FL changes the analytics path: instead of shipping logs or indicators to a central TIP for model training, each organization trains locally and contributes only privacy-preserving updates. DP and secure aggregation change the trust boundary: in the simulated protocol, the aggregator is exposed to only an aggregate of protected updates, not raw features.

From an operational standpoint, three tensions emerge:

- **Performance.** Latency must remain tolerable for human analysts who are triaging active threats. A $\approx$ 40 s PIR round is acceptable in investigative workflows (threat hunting, enrichment) but may be slow for automated inline blocking or SOAR playbooks. Engineering work is still required to cache common lookups or amortize retrieval over time.
- **Bandwidth.** PIR adds communication overhead; FL adds periodic model exchanges. Both must fit within bandwidth budgets of participating organizations, including those with lower-capacity links (e.g., MSSPs, small partners).
- **Privacy vs. usability.** Some privacy knobs lower utility. For example, raising σ can improve DP guarantees but degrades F_1. Likewise, forcing constant-size PIR batches mitigates traffic analysis but prevents oppor-

tunistic "lightweight" lookups. A production deployment will need policies or SLAs to tune these trade-offs per use case.

C. REGULATORY AND GOVERNANCE CONSIDERATIONS

Beyond pure technical performance, privacy-preserving CTI sharing must satisfy legal and governance requirements. Table 17 maps our design choices to major frameworks, including GDPR [34], CCPA [35], NIST SP 800-53 [32], and ISO/IEC 27701 [33]. The combination of PIR, fixed-shape batching, FL, formal client-level DP for LR, and secure aggregation aligns with principles such as data minimization, pseudonymization, and restricted data disclosure. Importantly, formal (ε, δ) guarantees apply only to the DP mechanism; other components provide architectural protections and deployment-oriented hardenings under stated assumptions. In other words, the framework is designed to be technically viable and structured to reduce compliance risk for organizations that share threat intelligence across borders.

Overall, these properties suggest that privacy-preserving CTI collaboration is not limited to academic feasibility. The design aligns with regulatory expectations and can be justified to auditors or data protection officers in multinational environments.

D. TOWARD PRODUCTION-GRADE DEPLOYMENT

In its current form, the framework is *deployment-oriented* rather than purely conceptual: it runs end-to-end, quantifies latency/overhead for PIR, and reports federated utility under protected updates for $K=5$ clients. Still, several technical steps are needed before hardening it for production CTI sharing:

- **Caching and hierarchical PIR.** Reducing PIR round time and bandwidth, e.g., via batched retrieval of high-demand IoCs, prefetching, or moving toward single-server PIR / OT-/DPF-based schemes to relax the non-collusion assumption.
- **Adaptive DP and budget accounting.** Dynamically adjusting σ (and thus (ϵ, δ)) per round and per feature class, and reporting that budget to participants for auditability.
- **Byzantine-resilient aggregation.** Incorporating robust aggregation, anomaly scoring, and possibly attestations to mitigate poisoning (T7) without sacrificing privacy.
- **TIP integration.** Exposing the PIR/FL/DP pipeline behind standard TAXII/STIX-style interfaces so that MISP/OpenCTI instances can "drop in" without modifying analyst workflows. This is critical for adoption.

In summary, the prototype shows that combining PIR, FL, formal DP for LR, fixed-shape batching, and secure aggregation can deliver a measurable privacy posture that goes beyond encrypted transport under stated assumptions. The remaining work is mainly about scale (PIR bandwidth), robustness (Byzantine resistance), and integration (TIP interoperability and policy reporting), rather than about the basic feasibility of the combined architecture.

X. CONCLUSION AND FUTURE WORK

This paper presented a deployment-oriented, workflow-level privacy orchestration for cyber threat intelligence (CTI) that coordinates Private Information Retrieval (PIR), Federated Learning (FL), and Differential Privacy (DP) across the query-learn-update lifecycle of an operational CTI pipeline. Rather than proposing a new cryptographic primitive, the contribution lies in addressing a CTI-specific problem fragmentation gap: in practical TIP systems, private threat lookup, collaborative learning, model-update privacy, and metadata leakage reduction are typically handled by disparate techniques with no shared operational framework or explicit guarantee boundaries. The framework additionally incorporates two practical deployment safeguards, fixed-shape PIR queries and secure aggregation, to mitigate residual side-channel and single-update exposure risks under the stated trust and threat-model assumptions. Formal (ε, δ) guarantees apply exclusively to the DP-hardened federated learning component; all other mechanisms are positioned as deployment-oriented hardenings whose security properties hold under the assumptions stated in each respective section.

The prototype implementation, evaluated on a synthetic AbuseIPDB-style dataset containing 2M indicators, demonstrated that a measurable privacy posture (beyond encrypted transport) can be achieved with quantified overhead. PIR achieved query confidentiality with an average latency of $\approx$ 40 s and $\approx$ 16 MB transfer per batch query, consistent with current multi-server PIR performance bounds. The DP-FedAvg evaluation comprises three tiers: a corrected canonical $T=1$ baseline (Tables 12, 9), a reviewer-driven multi-round analysis across $T \in \{1, 10, 20\}$ (Table 13), and an auxiliary clip-norm sensitivity sweep across $C \in \{0.5, 1.0, 2.0\}$ at $T=20$ (Table 14). Under the corrected $T=1$ baseline, federated LR achieved $F_1=0.795$ at $\sigma=2.0$ ($\varepsilon\approx2.54$) and $F_1\approx0.52$ at $\sigma=3.0$ ($\varepsilon\approx1.66$); the multi-round analysis reveals that this single-round ranking does not necessarily persist under extended composition, and the clip-norm sweep confirms that the observed utility pattern is parameter-sensitive rather than an artefact of a single clipping configuration. Compared to the FL-only baseline ($F_1=0.839$), the DP overhead at $T=1$ yields retention ratios of $\approx0.95\times$ at $\sigma=2.0$ and $\approx0.61\times$ at $\sigma=3.0$. These results demonstrate a quantified privacy–utility trade-off; the system is usable under carefully selected noise levels but not at all privacy budgets.

Beyond technical feasibility, the framework is designed to align with core privacy and governance principles defined in GDPR, CCPA, ISO/IEC 27701, and NIST SP 800-53 by enforcing data minimization, pseudonymization, and restricted disclosure of analytics outputs. In contrast to prior FL-based CTI frameworks that focus on either collaborative analytics or update privacy in isolation [25]–[29], the proposed system jointly addresses query indices, metadata channels (via traffic shaping), and federated model updates through an integrated PIR–FL–DP design (with formal (ε, δ) guarantees applying exclusively to the DP mechanism for LR; other components are deployment hardenings). This positions

TABLE 17. Compliance Mapping of the Proposed Framework to Data Protection Regulations and Standards

Regulation / Standard	Relevant Aspect	Our Framework
GDPR (EU) [34]	Data minimization, purpose limitation, anonymization	PIR conceals query indices; fixed-shape PIR mitigates metadata-level intent leakage; DP (LR) limits per-client leakage from model updates under formal accounting
CCPA (California) [35]	Right to opt-out, transparency, limited sharing	FL avoids centralized sharing of raw telemetry; secure aggregation is designed to limit exposure of any single contributor's update to other parties
NIST SP 800-53 [32]	Security and privacy controls for information systems	PIR/FL/DP align with control families such as Access Control (AC), System and Communications Protection (SC), and System Integrity (SI), by limiting what data leaves each boundary and how it is observable
ISO/IEC 27701 [33]	Privacy Information Management System (PIMS), accountability	The framework supports pseudonymization and minimization: only aggregated, protected intelligence is exchanged, not analyst queries or identifiable telemetry

the system not only as a proof of concept but also as a compliance-oriented prototype for multinational intelligence-sharing environments.

FUTURE WORK

Future research and engineering efforts will focus on extending scalability, robustness, and integration maturity:

1) **Optimized PIR Techniques:** Reduce latency and bandwidth cost through batched, recursive, or hierarchical PIR schemes, client-side caching, and exploration of single-server variants (e.g., DPF-PIR, OT-PIR, ORAM-based retrieval) to relax the non-collusion assumption.
2) **Adaptive Privacy Calibration:** Implement dynamic DP noise scaling based on model convergence or feature sensitivity, with automated (ϵ, δ) budget accounting via an RDP accountant across rounds and audit logging.
3) **Robust and Verifiable Aggregation:** Extend beyond secure averaging to incorporate Byzantine-resilient aggregation, anomaly-scored weighting, and cryptographically verifiable participation (e.g., zero-knowledge proofs or attestation of honest updates).
4) **Advanced Privacy Primitives:** Integrate Private Set Intersection (PSI) for collaborative IoC correlation without exposure of non-overlapping data, and explore homomorphic encryption (HE) for selective secure analytics over encrypted indicators.
5) **Operational Integration:** Pilot the framework in production-grade TIPs such as MISP and OpenCTI, evaluate interoperability with STIX/TAXII-based data flows, and benchmark scalability under real-world threat feed volumes.

Overall, the presented architecture demonstrates that PIR, FL, and formal DP for LR, augmented with fixed-shape batching and secure aggregation, can jointly deliver a privacy-preserving CTI workflow under stated assumptions. The evaluation provides prototype-level evidence of feasibility: the integrated system functions end-to-end, produces bounded and reproducible privacy–utility trade-offs, and maintains meaningful detection capability under moderate noise levels. The added multi-round experiments strengthen generalizability assessment by showing how cumulative privacy budgets and utility evolve across $T \in \{1, 10, 20\}$; the auxiliary clip-norm analysis strengthens robustness of interpretation by confirming that the observed trade-off structure persists across clipping norms $C \in \{0.5, 1.0, 2.0\}$; however, neither extension alters the paper's explicit guarantee boundaries, which continue to limit formal (ε, δ)-DP claims to the DP-FedAvg component for logistic regression. Utility degrades materially at aggressive privacy budgets, and class imbalance, threshold selection, and noise calibration require environment-specific tuning. The system is therefore a research prototype demonstrating the viability of combined PIR–FL–DP architectures for CTI, not a production-ready deployment; further calibration, scaling to larger and more diverse CTI feeds, Byzantine-resilient aggregation, and real-world pilot studies are needed before operational adoption. The system thus forms a research foundation for the next generation of secure, regulation-aligned, and scalable CTI ecosystems, paving the way toward adaptive privacy controls, efficient PIR protocols, and resilient federated learning under adversarial conditions.

REFERENCES

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private Information Retrieval," in *Proc. 36th IEEE Symp. Foundations of Computer Science (FOCS)*, Milwaukee, WI, USA, pp. 41–50, 1995, doi: 10.1109/SFCS.1995.492461.

[2] S. Pandey, H. Azath, R. U. Rahman, and H. Lamkuche, "Privacy-Preserving Model for Cyber Threat Intelligence Sharing Across Multi-Organizational Platforms," in *Proc. IEEE CSNT*, Mar. 2025, doi: 10.1109/CSNT64827.2025.10968450.

[3] P. Huff, S. Massengale, T. V. X. Phuong, and S. N. G. Gourisetti, "A Privacy-Preserving Cyber Threat Intelligence Sharing System," in *Proc. IEEE TPS-ISA*, Washington, DC, USA, Oct. 2024, doi: 10.1109/TPS-ISA62245.2024.00016.

[4] S. Mare, J. Polakis, and M. Bailey, "Exploring Design and Governance Challenges in Privacy-Preserving Threat Intelligence Sharing," in *Proc. ACM Workshop on Artificial Intelligence and Security*, 2020.

[5] A. Zafar, T. Pervez, M. A. Jan, and X. He, "A Distributed Ledger for Non-attributable Cyber Threat Intelligence Exchange," *Cluster Computing*, Springer, 2022.

[6] Y. Li, Q. Wang, M. Chen, and W. Zhu, "Cache-Aided Multi-User Private Information Retrieval Using PDA," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1302–1320, 2022.

[7] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Cyber Threat Intelligence Sharing Scheme Based on Federated Learning for Network Intrusion Detection," *J. Netw. Syst. Manage.*, vol. 31, no. 1, pp. 1–25, 2023.

[8] J. Zhang, S. Ma, and W. Wei, "Similarity Based Interactive Private Information Retrieval," *Cluster Computing*, Springer, 2021.

[9] H. Corrigan-Gibbs, S. Kim, and D. Wu, "Authenticated Private Information Retrieval," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, pp. 2873–2887, 2023.

[10] A. K. Das, M. Conti, M. Aledhari, and C. Wang, "FedCTI: Federated Learning and Cyber Threat Intelligence on the Edge for Secure IoT Networks," in *Proc. 17th ACM Asia Conf. Comput. Commun. Secur. (AsiaCCS)*, pp. 123–134, 2024.

[11] Y. Li, F. Yang, and H. Kim, "Blockchain and Federated Learning for Sharing Threat Detection Models as Cyber Threat Intelligence," in *Proc. ACM Workshop on Decentralized IoT Systems and Security (DISS)*, pp. 56–65, 2023.

[12] R. Sharma, A. Patel, and D. Yadav, "Privacy Preserving Advanced Persistent Threat Detection Using Fed-Adv-LSTM," *Cluster Computing*, Springer, 2025.

[13] T. Ahmad, L. Nguyen, and J. Li, "GraphFedAI Framework for DDoS Attack Detection in IoT Systems Using Federated Learning and Graph-Based Artificial Intelligence," *Scientific Reports*, Nature, vol. 15, Art. 10826, 2025, doi: 10.1038/s41598-025-10826-0.

[14] S. Kumari, J. Zhang, and A. K. Das, "A Blockchain-Enabled Federated Adversarial Learning Framework for CTI Sharing," *SSRN Electron. J.*, preprint, 2025.

[15] S. Gupta, P. Bhattacharya, and R. Mehta, "Generative AI for Cyber Threat Intelligence: Applications, Challenges, and Analysis of Real-World Case Studies," *Artificial Intelligence Review*, Springer, 2025, doi: 10.1007/s10462-025-11338-z.

[16] S. Al-Dhuraibi, C. Boix, and A. Shoker, "Blockchain-Powered Secure and Scalable Threat Intelligence System With Graph Convolutional Autoencoder and Reinforcement Learning Feedback Loop," *IEEE Access*, vol. 9, 2021.

[17] J. Zhao *et al.*, "The Federation Strikes Back: A Survey of Federated Learning Privacy Attacks, Defenses, Applications, and Policy Landscape," *ACM Comput. Surv.*, vol. 57, no. 9, Art. 230, pp. 1–37, Apr. 2025, doi: 10.1145/3724113.

[18] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014.

[19] M. Chen, Z. Wang, H. Xu, and Q. Yang, "A Review of Research on Secure Aggregation for Federated Learning," *Future Internet*, vol. 17, no. 7, Art. 308, pp. 1–26, 2025, doi: 10.3390/fi17070308.

[20] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in *Proc. IEEE Symp. Security and Privacy (S&P)*, San Jose, CA, USA, pp. 3–18, 2017, doi: 10.1109/SP.2017.41.

[21] J. Sun, Y. Shen, and Y. Chen, "Differentially Private Federated Learning: A Comprehensive Survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 2528–2544, 2023, doi: 10.1109/TNNLS.2022.3184580.

[22] A. El Ouadrhiri, P. H. Phung, N. Nasser, B. Boudine, and M. R. Abid, "Privacy-preserving federated learning approach based on Hensel's Lemma and differential privacy," *Computational Intelligence and Neuroscience*, Springer, vol. 2025, Article 36, 2025, doi: 10.1007/s43926-025-00236-z.

[23] A. Khraisat, I. Virk, A. Alqahtani, and C. Valli, "PEIoT-DS: A privacy-enhanced IoT defence system using federated learning for intrusion detection," *Computational Intelligence and Neuroscience*, Springer, vol. 2025, Article 169, 2025, doi: 10.1007/s43926-025-00169-7.

[24] H. Okada, R. Player, S. Pohmann, and C. Weinert, "Towards practical doubly-efficient private information retrieval," in *Lecture Notes in Computer Science (LNCS)*, Springer, pp. 251–272, 2025, doi: 10.1007/978-3-031-78679-2_14.

[25] T. Moulahi, R. Jabbar, A. Alabdulatif, S. Abbas, S. El Khediri, S. Zidi, and M. Rizwan, "Privacy-preserving federated learning cyber-threat detection for intelligent transport systems with blockchain-based security," *Expert Systems*, vol. 40, no. 5, Art. e13103, 2023, doi: 10.1111/exsy.13103.

[26] A. Sleem and I. Elhenawy, "Enhancing cyber threat intelligence sharing through a privacy-preserving federated learning approach," *J. Cybersecurity Inf. Manage.*, vol. 9, no. 2, pp. 51–59, 2022, doi: 10.54216/JCIM.090205.

[27] N. N. Sakhare, R. Kulkarni, N. Rizvi, D. Raich, A. Dhablia, and S. P. Bendale, "A decentralized approach to threat intelligence using federated learning in privacy-preserving cyber security," *J. Electr. Syst.*, vol. 19, no. 3, pp. 106–125, 2023, doi: 10.52783/jes.658.

[28] E. M. Timofte, M. Dimian, A. Graur, A. D. Potorac, D. Balan, I. Croitoru, D.-F. Hriţcan, and M. Puşcaşu, "Federated learning for cybersecurity: A privacy-preserving approach," *Appl. Sci.*, vol. 15, no. 12, Art. 6878, 2025, doi: 10.3390/app15126878.

[29] M. Rahmati and A. Pagano, "Federated learning-driven cybersecurity framework for IoT networks with privacy-preserving and real-time threat detection capabilities," *Informatics*, vol. 12, no. 3, Art. 62, 2025, doi: 10.3390/informatics12030062.

[30] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proc. ACM SIGSAC Conf. on Computer and Communications Security (CCS)*, Dallas, TX, USA, Oct.–Nov. 2017, pp. 1175–1191. doi: 10.1145/3133956.3133982

[31] A. Nelson, S. Rekhi, M. Souppaya, and K. Scarfone, "Incident Response Recommendations and Considerations for Cybersecurity Risk Management," *NIST Special Publication 800-61 Rev. 3*, Apr. 2025.

[32] Joint Task Force Transformation Initiative, "NIST Special Publication 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations," *NIST*, Gaithersburg, MD, 2020.

[33] International Organization for Standardization, "ISO/IEC 27701:2019 Security Techniques Extension to ISO/IEC 27001 and ISO/IEC 27002 for Privacy Information Management Requirements and Guidelines," *ISO/IEC Standard*, Geneva, Switzerland, 2019.

[34] European Union, "Regulation (EU) 2016/679: General Data Protection Regulation (GDPR)," *Off. J. Eur. Union*, L119/1, 2016.

[35] State of California, "California Consumer Privacy Act of 2018 (CCPA)," *Calif. Civil Code*, Title 1.81.5, 2018.

[36] Abuse.ch, "URLhaus Malware URL Tracker," [Online]. Available: https://urlhaus.abuse.ch/. Accessed: Nov. 14, 2025.

[37] AbuseIPDB, "AbuseIPDB Threat Intelligence API," [Online]. Available: https://www.abuseipdb.com/

[38] A. AlSobeh, A. Shatnawi, and A. Magableh, "AspectFL: Aspect-Oriented Programming for Trustworthy and Compliant Federated Learning Systems," *Information*, vol. 16, no. 12, Art. no. 1048, 2025, doi: 10.3390/info16121048.

[39] G. Frisbier, O. Darwish, A. AlSobeh, and A. Al-Shorman, "Identifying the Origins of Business Data Breaches Through CTC Detection," in *Network and System Security*, *Lecture Notes in Computer Science*, vol. 15564, Springer, Singapore, 2025, pp. 387–406, doi: 10.1007/978-981-96-3531-3_19.

EMRE CAMALAN received his B.S. degree in Computer Engineering from Atılım University and his M.S. degree in Computer Engineering from Ankara Yıldırım Beyazıt University. He is currently pursuing a Ph.D. degree in Computer Engineering at Işık University. He serves as the Cybersecurity Manager at İşNet, a Turkish MSSP and SOC service provider, where he leads the entire cybersecurity division of over twenty professionals responsible for SOC, SIEM/SOAR, MDR, EDR, DLP, DAM, and gateway security operations. Previously, he worked as a Professional Services Consultant at AlgoSec, focusing on firewall optimization and compliance automation using FireFlow and AFA platforms. His research interests include artificial intelligence in cybersecurity, privacy-preserving threat intelligence sharing, federated learning, and differential privacy. He holds several professional certifications, including CEH, CompTIA Security+, and ISO/IEC 27001 Lead Auditor.

BARIS CELIKTAS received his B.S. degree in Systems Engineering from the National Defense University in 2008, his M.S. degree in Applied Informatics from Istanbul Technical University in 2018. He completed his Ph.D. in Cybersecurity Engineering and Cryptography at the Institute of Informatics, Istanbul Technical University, in 2022. Currently, he serves as an Assistant Professor in the Computer Engineering Department and is the Director of the Cybersecurity Graduate Program at Isik University. Besides, he is a cybersecurity consultant and architect, specializing in enterprise cybersecurity and cryptography solutions, cloud security, risk management, and governance. He holds numerous industry-recognized certifications, including CISSP, CCSP, CISM, CISA, CRISC, SSCP, CCNP, Sec+, CySA+, C|EH(M), and ISO 27001, 22301, 20000, 27701, and 42001 as a Lead Auditor and Lead Implementer, along with GDPR DPO and NIST Cybersecurity Consultant. His areas of research interest include cybersecurity, network security, cloud computing, cryptography, malware analysis, risk management, and security applications.