SyPrFL: Sybil-Resilient Privacy-Preserving Federated Learning via Differentially-Private Projection Clustering

Md Mazharul Islam[a,*], Mohammad Kaosain Akbar[b] and Niaz Ashraf Khan[c]

[a]Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh
[b]Department of Computer Science, University of Calgary, Calgary, Alberta, Canada
[c]Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh

ARTICLE INFO

Keywords:
Federated Learning
Sybil Attacks
Differential Privacy
Secure Aggregation
Byzantine Resilience
IoT Intrusion Detection

ABSTRACT

Federated learning enables collaborative model training without sharing raw data and is widely adopted for privacy-sensitive tasks such as IoT intrusion detection. In adversarial settings, however, it faces a structural impasse: Byzantine-robust aggregation needs plaintext gradients to compute the relational statistics it relies on, thereby reintroducing the gradient-inversion attack surface, whereas privacy-preserving aggregation hides individual updates and so removes the very signal needed to identify malicious contributions. Coordinated Sybil attackers exploit this gap, and no prior protocol jointly delivers cryptographic privacy, Sybil resilience, and competitive utility under realistic non-IID workloads. We propose SyPrFL, a two-channel scheme in which each client submits a mask-based secure-aggregation share of its full update together with a low-dimensional, differentially-private random projection of the same update. The server clusters the projections with HDBSCAN, scores cluster cohesion through a novel Differentially-Private Projection Cluster Cohesion (DP-PCC) primitive, and binds the resulting trust weights to the masked channel via a weighted secure-aggregation construction, keeping detection-side and aggregation-side information cryptographically disjoint. We establish formal guarantees on privacy, detection error, robustness, and convergence. Across three public IoT-IDS datasets (N-BaIoT, TON-IoT, Bot-IoT), ten baselines, and six attack instantiations, SyPrFL uniquely combines privacy-preserving aggregation with provable Sybil detection while retaining IoT-IDS-grade utility.

1. Introduction

Federated Learning (FL) trains a shared model across distributed clients without centralizing their data, and has emerged as the leading paradigm for privacy-sensitive applications including mobile inference, healthcare analytics, and the security monitoring of the Internet of Things (IoT) [1]. Among these, FL-based Intrusion Detection Systems (IDS) for IoT environments are particularly compelling [2]: IoT networks combine high-volume telemetry, strict regulatory constraints on raw data sharing, and an attack surface dominated by rapidly evolving botnets, none of which can be addressed by centralized batch training within current legal regimes.

This federated setting introduces two adversarial surfaces absent from centralized learning. Shared model updates leak information about the underlying training data, with gradient-inversion techniques able to reconstruct substantial portions of a client's training batch from a single update [3]. At the same time, clients lie outside the server's administrative control, so one or more compromised participants may submit arbitrarily crafted updates to degrade utility, induce targeted misclassifications, or implant backdoors [4]. Privacy threats and Byzantine threats are typically treated as independent problems with independent solutions: secure or differentially-private aggregation prevents the server from inspecting individual updates [5], while robust aggregation rules discard or down-weight updates that appear as statistical outliers [6].

A third adversarial pattern sits between these two and is the focus of this work. Under a Sybil attack, a single physical adversary registers multiple distinct client identities and orchestrates their updates collectively, presenting the server with a coordinated minority masquerading as a majority [7]. Sybils are especially damaging in IoT-FL because they break the very assumption robust aggregators rely on: colluding clones, by every standard distance metric, present lower pairwise distance and higher mutual similarity than the genuinely heterogeneous honest clients, and are therefore mistaken by Byzantine defenses for a consensus majority. Weak IoT-gateway authentication makes such identities cheap to manufacture, so the threat is operational rather than theoretical.

The defining difficulty is that the natural detection signal for Sybil collusion is the relational similarity between clients' updates, and computing this signal requires the server to inspect individual updates in plaintext [8]. Privacy-preserving aggregation is designed precisely to deny the server that view. The two requirements pull in opposite directions: any scheme that hides individual updates eliminates the Sybil-detection signal, and any scheme that exposes the signal eliminates privacy. A naive composition - for example, computing similarity on encrypted updates - either incurs prohibitive cryptographic overhead or leaks information through side channels of the similarity computation. The result is a deployment-relevant impasse: no existing federated aggregation protocol, to our knowledge,

*Corresponding author.
mazharul.islam1@northsouth.edu (M.M. Islam); mohammadkaosain.akba@ucalgary.ca (M.K. Akbar); niaz.ashraf@bracu.ac.bd (N.A. Khan)

simultaneously hides individual updates and resists coordinated Sybil collusion under realistic non-IID conditions [9].

This paper proposes SyPrFL, a Sybil-resilient privacy-preserving federated learning framework that closes the gap without resorting to general-purpose secure multi-party computation or zero-knowledge proofs. The core observation is that Sybil detection does not require the full client update; it requires only enough information about the relational structure of updates across clients to identify groups whose mutual similarity is statistically anomalous. Each client therefore transmits two complementary messages in each round: a low-dimensional random projection of its update perturbed with calibrated Gaussian noise, and the full update protected by mask-based secure aggregation. The server inspects the noisy projections freely but cannot inspect any individual full update.

Detection proceeds on the projection channel via the Differentially-Private Projection Cluster Cohesion (DP-PCC) primitive introduced in this paper. The server clusters the noisy projections, computes a cohesion statistic for each cluster equal to the mean pairwise distance, and compares it against an adaptive baseline that tracks the natural non-IID heterogeneity of recent rounds. Clusters whose cohesion falls below a configurable fraction of the baseline are flagged as suspected Sybil groups, and their members receive a reduced trust weight [10]. Aggregation then proceeds through a trust-weighted extension of mask-based secure aggregation, in which the server applies the trust weights to the protected client shares before unmasking. The server learns only the trust-weighted aggregate, never an individual update. Privacy follows from the secure-aggregation sub-protocol and the differentially-private projection; Sybil resilience follows from the statistical separation between coordinated and honest clients in projection space; and utility is preserved because the calibration of the differential-privacy noise to the projection dimension leaves honest clients untouched under normal operation. The contributions of this paper are as follows.

1. This work identifies and formalizes the Sybil blind spot in privacy-preserving FL, showing why prior defenses cannot close it without abandoning either privacy or robustness.
2. SyPrFL is proposed as the first federated aggregation protocol, to our knowledge, that simultaneously achieves (ε, δ)-differential privacy, secure aggregation of full client updates, and provable Sybil-group detection, without recourse to general-purpose cryptographic machinery.
3. A new statistical primitive, Differentially-Private Projection Cluster Cohesion (DP-PCC), is introduced to detect coordinated client collusion through the cohesion of differentially-private projections against an adaptive non-IID baseline, with bounded false-positive and false-negative rates as a function of projection dimension, noise scale, and client heterogeneity.
4. A comprehensive IoT-IDS evaluation suite is constructed, spanning three public datasets (N-BaIoT, TON-IoT, Bot-IoT), six attack instantiations, and ten baseline defenses including FoolsGold, FLTrust, Krum, Multi-Krum, Median, Trimmed Mean, FedAvg, DP-FedAvg, SecAgg, and the closest published IoT-IDS baseline PEIoT-DS.
5. Empirical results across all three datasets and the full attack suite show that SyPrFL retains near-clean classification accuracy in regimes where privacy-only baselines collapse and robustness-only baselines lack the privacy guarantee required for deployment.

The remainder of this paper is organized as follows. Section 2 surveys related work across Byzantine-robust aggregation, privacy-preserving aggregation, and federated IoT intrusion detection, positioning SyPrFL against the closest competitors. Section 3 fixes notation and reviews the cryptographic and differential-privacy primitives on which SyPrFL relies. Section 4 formalizes the cross-silo system model and Section 5 the threat model, including the concrete attack instantiations used in the evaluation. Section 6 presents the SyPrFL protocol and the DP-PCC detection primitive, and Section 7 establishes its privacy, detection, robustness, and convergence guarantees. Section 8 describes the experimental setup, Section 9 reports and discusses the results, and Section 10 concludes with a summary of contributions and directions for future work.

2. Related Work

The literature relevant to SyPrFL spans three loosely connected currents: Byzantine-robust aggregation that hardens federated and decentralized training against malicious participants, privacy-preserving aggregation that conceals individual updates from the server, and federated intrusion-detection systems that adapt both lines of defense to the constraints of IoT deployments. A growing number of works attempt to combine robustness and privacy, and a smaller set confronts coordinated clients masquerading as a benign majority. We review the most pertinent contributions across these currents and isolate the gap that SyPrFL is designed to fill.

The first current concerns robust aggregation in decentralized and federated optimization. Li et al. [11] filter neighbours whose accumulated loss exceeds an agent's own, achieving resilience at linear cost but with convergence established mainly for convex models. Li et al. [12] generalize the geometric median through a centerpoint aggregation rule that proves an $\mathcal{O}(1/i)$ rate, yet centerpoint computation becomes co-NP-complete for $d \geq 4$, forcing approximation in realistic parameter spaces. Confronting non-IID heterogeneity directly, Wu et al. [13] show that centralized rules such as Krum and the geometric median frequently fail to reach consensus in decentralized non-IID regimes and propose an iterative outlier-filtering aggregator, while Xu et al. [14] assign per-layer filtering radii to account for differing learning speeds across network layers. The recurring limitation

is structural rather than incidental: every estimator requires the server to inspect individual updates in plaintext, which is precisely the capability privacy mechanisms remove, and none evaluates against attackers that coordinate to imitate the honest majority.

The privacy current attacks the problem from the opposite direction, concealing individual contributions, and in doing so hides the very signal robust aggregation depends on. So et al. [15] present the first single-server Byzantine-resilient secure-aggregation framework (BREA) but assume i.i.d. data for its convergence guarantee. Xia et al. [16] advance this to full information-theoretic privacy that leaks no pairwise distances, though communication and computation grow as high-degree polynomials in the number of users. Zhao et al. [17] take a hardware route using trusted execution to reconcile masking with the visibility robust aggregation needs, but are bounded by limited enclave memory and exclude side channels, while Aziz [18] proposes a zk-SNARK-based protocol that proves correct application of differential privacy at the cost of proving times that escalate with model dimension. These schemes deliver rigorous privacy but either forfeit robustness or impose overheads incompatible with real-time deployment, and none addresses Sybil-style collusion.

A hybrid current attempts to deliver privacy and robustness together while remaining lightweight, usually by adding differential privacy to a trust-based aggregator, and a closely related application current grounds these mechanisms in IoT intrusion detection. He et al. [19] couple adaptive local differential privacy with secure multi-party computation and anomaly detection for financial federations, while Guo et al. [20] pursue model compression and Byzantine robustness jointly through device-to-device validation, and Chaurasia et al. [21] fuse reputation scoring with energy-adaptive participation for industrial IoT, requiring two-thirds honest participation per cluster. In the IDS setting, Khraisat et al. [22] improve convergence on the N-BaIoT botnet dataset but provide no defense against malicious participants; Al Amro et al. [23] combine differential privacy, secure multi-party computation, and a weight-distance detector that can be evaded by stealthy backdoors; and Sultana et al. [24] combine homomorphic encryption, differential privacy, and a trimmed-mean aggregator for medical IoT at non-negligible overhead. These hybrids represent real progress, yet they continue to treat malicious clients as statistical outliers relative to a benign majority, an assumption that coordinated Sybils violate by construction, and they do not evaluate Sybil-specific attack vectors.

Two observations crystallize the gap, corroborated by recent surveys [25, 26]. First, the works that provide strong robustness do so by inspecting plaintext updates, while the works that provide strong privacy do so by removing exactly the relational signal robustness requires; the hybrids that bridge the two still presuppose that adversaries are isolated outliers. Second, none treats Sybil collusion as a first-class threat: colluding identities present lower mutual distance than genuinely heterogeneous honest clients and are therefore mistaken for a trustworthy consensus, yet no surveyed protocol simultaneously hides individual updates and detects this coordination pattern under realistic non-IID IoT-IDS conditions. SyPrFL is designed for this empty intersection, decoupling the Sybil-detection signal onto a low-dimensional differentially-private projection channel while protecting full updates under secure aggregation.

Table 1 situates SyPrFL against the eight prior works most closely aligned with its goals, selected so that none duplicates the papers discussed above. The works are compared along six dimensions: the privacy mechanism employed, the Byzantine or robustness defense, explicit Sybil resilience, the application domain, the principal evaluation dataset, and the methodological category. The table reinforces the conclusion drawn from the prose: combining a privacy mechanism with a robustness defense is common, but explicit Sybil resilience is at best partial and incidental, and is never co-located with cryptographic privacy of individual updates under non-IID IoT-IDS workloads. SyPrFL occupies the otherwise empty cell at the intersection of differential privacy, secure aggregation, and provable Sybil detection in the IoT intrusion-detection setting.

3. Preliminaries

This section fixes the notation, states the federated learning setup, and reviews the cryptographic and statistical building blocks that SyPrFL relies on: mask-based secure aggregation and the Gaussian mechanism for differential privacy, together with the random projection, HDBSCAN clustering, and cluster-cohesion test that compose into its Sybil-detection pipeline.

3.1. Notation

Vectors are denoted by bold lowercase symbols ($\mathbf{x}$) and matrices by bold uppercase symbols ($\mathbf{X}$). The Euclidean norm is $\|\cdot\|$, the inner product $\langle\cdot,\cdot\rangle$, and the k-dimensional isotropic Gaussian $\mathcal{N}(\mu, \sigma^2 I_k)$. For positive integer m, $[m] = \{1, \dots, m\}$. Table 2 fixes the symbols specific to federated learning, the threat model, and the SyPrFL mechanism.

3.2. Federated Learning

Federated Learning is a distributed-optimization paradigm in which n clients collaboratively minimize the weighted global empirical risk $F(\theta) = \sum_{i=1}^{n} p_i F_i(\theta)$, where F_i is the local risk on dataset $\mathcal{D}_i$ and $p_i = |\mathcal{D}_i| / \sum_j |\mathcal{D}_j|$. Training proceeds in T synchronous rounds. At the start of round t, the server broadcasts the current global model $\theta^{(t)}$ and each client performs E epochs of local SGD with mini-batch size B, producing the update $\Delta_i^{(t)} = \theta_i^{(t)} - \theta^{(t)}$. The standard FedAvg aggregation [35] yields

$$\theta^{(t+1)} = \theta^{(t)} + \eta_t \sum_{i=1}^{n} p_i \, \Delta_i^{(t)}, \tag{1}$$

exposing every $\Delta_i^{(t)}$ to the server. SyPrFL replaces Eq. (1) with a trust-weighted secure aggregation in which the server

Table 1
Comparison of SyPrFL against the eight most closely aligned prior works, disjoint from those discussed in the prose.

Work	Privacy	Byz.	Sybil	Application	Dataset	Approach
Prajwalasimha et al. [27]	✓(DP)	✓	?	6G IDS	6G network data	Byz. agg. + DP
Shaikh et al. [28]	✓(DP+SMPC)	?	?	Critical infra.	Simulated (50 nodes)	DL + SecAgg + DP
Camalan et al. [29]	✓(PIR+DP)	–	?	Cyber threat intel.	AbuseIPDB, URLhaus	PIR + FL + DP
Haq et al. [30]	✓(DP)	✓	?	Smart grid	IEEE 33-bus, SCADA	Decentralized FL
Li et al. [31]	∂ (rep.)	✓	∂	Vehicular nets	CIFAR-10 (200 veh.)	Digital twin + BC
Ramalingam et al. [32]	✓(HE+DP)	✓	?	Smart-env. IoT	Energy mgmt. sim.	GenAI + BC + FL
Khaf et al. [33]	∂	✓	∂	6G NTN spectrum	Simulated NTN	Hier. RL + trust
Isong et al. [34]	✓(HE)	✓	∂	Genomic analytics	1000 Genomes	Explainable Sec-FL
SyPrFL (this work)	✓ (DP+SecAgg)	✓	✓	**IoT-IDS botnet**	**N-BaIoT, TON-IoT, Bot-IoT**	**DP-proj. + clust. + SecAgg**

Note. ✓ = provided; ∂ = partial or conditional; – = not provided; ? = neither claimed nor evaluated in the cited work.

Table 2
Notation used throughout the paper.

Symbol	Meaning
Federated learning	
n	Number of participating clients
$\mathcal{C} = \{C_1, \dots, C_n\}$	Set of clients
$\mathcal{S}$	Aggregating server
$\mathcal{D}_i$	Local dataset of client C_i
$\lvert\mathcal{D}_i\rvert, p_i$	Local size and aggregation weight
T	Number of FL training rounds
E	Local SGD epochs per round
B	Local mini-batch size
η_t	Global learning rate at round t
d	Model parameter dimension
$\theta^{(t)} \in \mathbb{R}^d$	Global model at round t
$\Delta_i^{(t)} \in \mathbb{R}^d$	Update of C_i in round t
C	Per-update ℓ_2 clipping bound
Adversary model	
$\mathcal{B} \subset \mathcal{C}$	Set of Sybil-controlled clients, $\lvert\mathcal{B}\rvert = f$
$\mathcal{H} = \mathcal{C} \setminus \mathcal{B}$	Honest clients
G	Number of Sybil collusion groups
$g \subseteq \mathcal{B}$	A single Sybil group, $\lvert g\rvert \geq 2$
SyPrFL mechanism	
$\mathbf{\Pi} \in \mathbb{R}^{k\times d}$	Public projection matrix
k	Projection dimension, $k \ll d$
$P_i, \widetilde{P}_i \in \mathbb{R}^k$	Raw and DP-noised projection
ε, δ	(ε, δ)-DP budget
σ	Gaussian DP noise scale
$C_g^{(t)}$	Intra-cluster cohesion of cluster g
$\tau^{(t)}$	Adaptive cohesion threshold
γ	Sybil-detection tightness factor
$w_i^{(t)} \in [0, 1]$	Trust weight of C_i in round t

observes neither the individual updates nor the trust weights of individual clients, only the weighted sum. Throughout the paper we work in the cross-silo regime ($n \in [20, 100]$, synchronous rounds, persistent clients), which matches the IoT-IDS gateway deployments described in Section 4.

3.3. Cryptographic and Privacy Primitives

Mask-based secure aggregation. SyPrFL builds on the mask-based secure aggregation protocol of Bonawitz et al. [36]. Each client holds a Diffie–Hellman keypair and derives, with every other client, a shared pseudo-random seed under DDH. Pairwise masks computed from these seeds are added with opposite signs across each client pair so that they cancel under summation. Each client C_i transmits the masked update

$$\widehat{\Delta}_i^{(t)} = \Delta_i^{(t)} + \sum_{j \in C \setminus \{i\}} m_{i,j}, \qquad m_{i,j} + m_{j,i} = 0, \quad (2)$$

allowing the server to recover the population sum $\sum_i \widehat{\Delta}_i^{(t)} = \sum_i \Delta_i^{(t)}$ without observing any individual update. Under DDH and an honest-but-curious server colluding with fewer than $n/2$ clients, the joint view of the corrupted parties is computationally indistinguishable from a view in which honest clients submit independent uniform random vectors, conditioned on the legitimate aggregate. Dropouts are tolerated by Shamir-sharing the seeds among clients; the analysis of Section 7 assumes a zero dropout budget for clarity, with the standard extension applying otherwise.

Differential privacy and the Gaussian mechanism. SyPrFL uses central (ε, δ)-differential privacy [37] as a formal guarantee on the low-dimensional projections the server uses for Sybil detection. A randomized mechanism $\mathcal{M}$ satisfies (ε, δ)-DP if its output distribution on any two neighboring inputs differs by at most a factor of e^ε up to additive δ. The Gaussian mechanism instantiates this guarantee by adding noise $\xi \sim \mathcal{N}(0, \sigma^2 I_k)$ to a function f of ℓ_2-sensitivity $S_2(f)$, with the calibration

$$\sigma \geq \frac{S_2(f)\sqrt{2\ln(1.25/\delta)}}{\varepsilon}. \quad (3)$$

In SyPrFL, f is the random projection $f(\Delta_i) = \mathbf{\Pi}\Delta_i \in \mathbb{R}^k$ and per-client ℓ_2-clipping at norm C ensures $S_2(f) \leq \|\mathbf{\Pi}\|_{\text{op}} \cdot C$. For tight tracking of cumulative privacy loss across T training rounds we use the Rényi DP accountant [38] with the subsampled Gaussian bound [39], converted to a final $(\varepsilon_T, \delta_T)$ guarantee at the end of training. Post-processing immunity ensures that all downstream computations on the noised projection (clustering, cohesion statistics, trust-weight assignment) introduce no additional privacy loss beyond that already accounted for in the projection step.

3.4. Detection Primitives

The SyPrFL Sybil test relies on three further primitives, presented together because they compose into a single detection pipeline.

Random projection. The Johnson-Lindenstrauss lemma [40] guarantees that a random Gaussian matrix $\mathbf{\Pi} \in \mathbb{R}^{k\times d}$ with entries $\mathbf{\Pi}_{j\ell} \sim \mathcal{N}(0, 1/k)$ and $k = \mathcal{O}(\log n)$ preserves all pairwise Euclidean distances among n points up to multiplicative distortion $(1 \pm \varepsilon_{\text{JL}})$ with high probability. For the cross-silo regime ($n \in [20, 100]$) a projection dimension of $k \in [32, 128]$ keeps distortion below $\varepsilon_{\text{JL}} = 0.1$. Crucially for SyPrFL, the JL guarantee implies that the *ratio* of intra-Sybil to inter-honest distances in projection space is preserved up to a small constant, so coordination signatures present in $\mathbb{R}^d$ remain visible in $\mathbb{R}^k$. The matrix $\mathbf{\Pi}$ is generated once by the server and broadcast as a public parameter.

HDBSCAN clustering. Given the set of DP-noised projections $\{\widetilde{P}_i\}_{i=1}^n \subset \mathbb{R}^k$, the server partitions them with HDBSCAN using a single hyperparameter m_{cl}, the minimum cluster size. HDBSCAN is chosen over DBSCAN or k-means for three reasons: it does not require the number of clusters G to be specified in advance (unknown to the server in our setting); it accommodates clusters of varying density (different Sybil groups exhibit different intra-group tightness); and it labels low-density singletons as noise rather than forcing them into clusters (matching the operational expectation that honest non-IID singletons should not trigger detection logic). We use $m_{\text{cl}} = 2$ throughout so that even small Sybil groups are candidates for detection. HDBSCAN runs in $\mathcal{O}(n^2)$ time, negligible at $n \leq 100$. The partition inherits the DP guarantee of the projection step by post-processing immunity.

Cluster-cohesion separation test. For a cluster $g \subseteq [n]$ of size at least 2, define the cluster cohesion as the mean pairwise distance

$$C_g = \frac{2}{|g|\,(|g|-1)} \sum_{i<j,\, i,j\in g} \left\| \widetilde{P}_i - \widetilde{P}_j \right\|. \tag{4}$$

Honest non-IID clusters have larger C_g (genuinely different gradients); coordinated Sybil groups have smaller C_g (similar gradients by construction). SyPrFL flags a cluster as Sybil when

$$C_g < \tau^{(t)}/\gamma, \tag{5}$$

where $\tau^{(t)}$ is an adaptive baseline tracking the running honest-cluster cohesion and $\gamma > 1$ is a tightness factor. The update rule for $\tau^{(t)}$ and the choice of γ are given in Section 6. By Gaussian concentration, C_g deviates from its expectation by at most $\mathcal{O}(\sigma/\sqrt{|g|})$ with high probability, so the threshold in Eq. (5) can be placed between the expected honest and Sybil cohesions with vanishing error rates as $|g|$ grows. The full false-positive and false-negative analysis is deferred to Section 7.

4. System Model

The SyPrFL framework targets a federated network of IoT intrusion-detection gateways in which independent administrative entities collaboratively train a shared classifier without exchanging raw network telemetry. The system model formalizes the participants, the communication and cryptographic substrate they share, the per-round protocol skeleton on which SyPrFL is built, and the operational assumptions under which its privacy and Sybil-resilience guarantees hold. The model is deliberately narrowed to the cross-silo regime, in which clients are gateway-class devices that remain online for the duration of training and are individually addressable through stable network identifiers; this matches the deployment characteristics of consortium-managed IoT-IDS far better than the cross-device regime of mobile keyboards or wearables.

4.1. Deployment Setting and Entities

Each gateway in a SyPrFL deployment is operated by an independent administrative entity such as a network operator, smart-building tenant, or managed security service provider. The gateway monitors a partition of global IoT traffic and must classify high-volume telemetry records in near-real time to detect botnet activity, denial-of-service attempts, and reconnaissance scans. The administrative entities collaborate on training because no single entity sees enough attack diversity to train a strong classifier in isolation, yet none of them can centralize raw telemetry: regulatory regimes such as GDPR, HIPAA, and NERC CIP prohibit cross-entity transfer of raw traffic, and competitive considerations make raw telemetry sharing operationally infeasible even where regulation would allow it.

A SyPrFL deployment comprises three classes of entities. The **clients** $\mathcal{C} = \{C_1, \ldots, C_n\}$ are n IoT-IDS gateways with $n \in [20, 100]$. Each client C_i owns a private local dataset $\mathcal{D}_i$ of labelled network-traffic records and holds a local copy of the current global model $\theta^{(t)} \in \mathbb{R}^d$. Clients are mutually distrustful; no client knows which others are honest, Sybil-controlled, or data-distribution-similar to its own. The **server** $\mathcal{S}$ is a single logical aggregating entity operated by a neutral coordinator, typically a consortium-managed cloud node, regulatory authority, or designated lead organization. The server is honest-but-curious in the sense formalized in the threat model: it follows the prescribed protocol exactly, but may inspect every message it receives in an attempt to infer information about individual clients. The **bulletin board** $\mathcal{BB}$ is an authenticated public

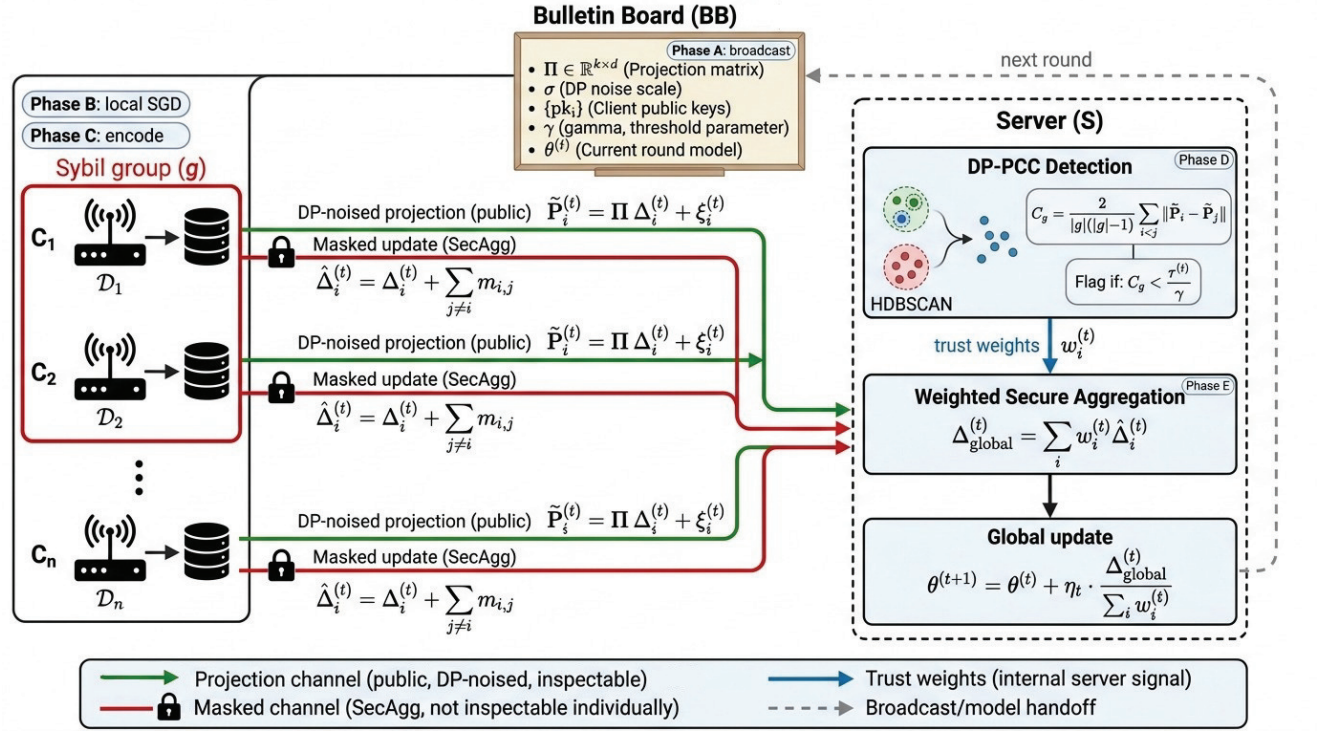

Fig 1: Sybil-Resilient Privacy-Preserving Federated Learning system architecture.

endpoint that distributes once-and-for-all protocol parameters and broadcasts the global model each round. It is realized in practice via the server's TLS endpoint together with a transparency log, and constitutes no separate trust anchor beyond the server itself.

The communication and cryptographic substrate is fixed before training begins. Every ordered pair of parties communicates over a TLS 1.3 channel offering confidentiality, integrity, and authenticity, with a public fixed-bound round duration Δ_R guaranteeing in-round delivery. Every client and the server hold long-term signing keys registered with the consortium's certificate authority, used both for TLS authentication and for the Diffie–Hellman setup of the secure-aggregation protocol. Prior to the first FL round, the server publishes on $\mathcal{BB}$ a set of immutable public parameters: the projection matrix $\mathbf{\Pi} \in \mathbb{R}^{k\times d}$ with entries $\mathbf{\Pi}_{j\ell} \sim \mathcal{N}(0, 1/k)$, the per-update ℓ_2-clipping bound $C > 0$, the DP noise scale σ derived from the target (ε, δ) budget through Eq. (3), the Sybil-detection tightness factor $\gamma > 1$, the adaptive-threshold smoothing factor $\alpha \in (0, 1)$, and the HDBSCAN minimum cluster size $m_{\text{cl}} \geq 2$. Publishing $\mathbf{\Pi}$ rather than concealing it is deliberate: SyPrFL derives no security benefit from a secret projection and gains auditability from a public one.

4.2. Per-Round Protocol Skeleton

Fig. 1 shows the SyPrFL architecture at a glance. Training proceeds in T synchronous rounds, each comprising five phases labelled (A)–(E) for later reference. The phases are described here at a high level; the full algorithmic specification, including the DP-PCC primitive, is given in Section 6.

Phase A (Broadcast). The server posts the current global model $\theta^{(t)}$ to the bulletin board. Each client retrieves $\theta^{(t)}$, verifies the signature, and initializes its local model to $\theta_i \leftarrow \theta^{(t)}$.

Phase B (Local training). Each client performs E epochs of local SGD on $\mathcal{D}_i$ with mini-batch size B, obtaining its local model $\theta_i^{(t)}$ and the update $\Delta_i^{(t)} = \theta_i^{(t)} - \theta^{(t)}$. The update is ℓ_2-clipped:

$$\Delta_i^{(t)} \leftarrow \Delta_i^{(t)} \cdot \min\Big(1, \frac{C}{\|\Delta_i^{(t)}\|}\Big). \tag{6}$$

Clipping bounds the ℓ_2-sensitivity of the projection mechanism required by the Gaussian DP calibration of Eq. (3), without altering the update direction.

Phase C (Encoding and submission). Each client computes, from the same clipped update, a DP-noised projection $\widetilde{P}_i^{(t)} = \mathbf{\Pi}\Delta_i^{(t)} + \xi_i^{(t)}$ with $\xi_i^{(t)} \sim \mathcal{N}(0, \sigma^2 I_k)$, and a masked full update $\widehat{\Delta}_i^{(t)} = \Delta_i^{(t)} + \sum_{j\neq i} m_{i,j}$ via the pairwise masks from Section 3.3. Both submissions are sent to the server. Because the two derivatives come from the same $\Delta_i^{(t)}$, a malicious client cannot present a benign projection while smuggling a malicious masked update, a property exploited in the threat model.

Phase D (Sybil detection and trust assignment). The server clusters the received projections via HDBSCAN, computes the cohesion statistic $C_g^{(t)}$ for each non-trivial

cluster, compares it against the adaptive threshold $\tau^{(t)}/\gamma$ via Eq. (5), and assigns a trust weight $w_i^{(t)} \in [0,1]$ to every client. Trust weights are the sole bridge between the public-projection channel and the encrypted aggregation channel.

Phase E (Weighted secure aggregation). The server combines the masked updates with the trust weights via a weighted-sum extension of secure aggregation, obtaining the aggregate $\Delta_{\text{global}}^{(t)}$. The new global model is computed as

$$\theta^{(t+1)} = \theta^{(t)} + \eta_t \frac{\Delta_{\text{global}}^{(t)}}{\sum_{i=1}^{n} w_i^{(t)}}, \tag{7}$$

and is broadcast in Phase A of the next round.

A subtle but essential property of the protocol is that each client submits two derivatives of the same local update rather than one. The projection enables Sybil detection without revealing the update; the masked update enables full-fidelity aggregation of the surviving updates. The binding between the two is enforced by the algorithm of Section 6: a malicious client that submits a benign projection but a malicious masked update gains nothing, because the trust weight derived from the projection is applied multiplicatively to the masked contribution before the server decodes the population sum.

5. Threat Model

SyPrFL is designed to remain secure and useful in the face of two independent and simultaneous adversary classes. The first is a Sybil-Byzantine client adversary that statically corrupts a minority of clients and partitions them into coordinated collusion groups; this adversary aims to degrade model utility, induce mispredictions, or implant backdoors through arbitrary protocol messages. The second is an honest-but-curious server that follows the protocol exactly but inspects every message it receives in an attempt to recover individual honest clients' updates or training data. We assume non-collusion between the two: the Sybil controller and the curious server pursue independent objectives and do not exchange information outside the protocol. This is standard in joint privacy-Byzantine federated learning and operationally realistic in cross-silo IoT-IDS, where the aggregating server is administered by a neutral consortium coordinator while Sybil corruptions arise from adversaries who compromise individual member organizations or spawn fake clients through weak IoT device identity. The remainder of this section formalizes the two adversary classes, catalogues the concrete attack instantiations exercised in the experiments, and states the operational assumptions under which the guarantees hold.

5.1. Adversary Classes

Definition 1 (Sybil-Byzantine client adversary)**.** *A Sybil-Byzantine adversary $\mathcal{A}_S$ is a probabilistic polynomial-time algorithm that, prior to training, statically corrupts a subset $\mathcal{B} \subset \mathcal{C}$ with $|\mathcal{B}| = f < n/2$, partitioned into $G \geq 1$ disjoint Sybil groups $\mathcal{B} = g_1 \sqcup \cdots \sqcup g_G$ with $|g_r| \geq 2$. In every round each $C_i \in \mathcal{B}$ may replace its prescribed pair $(\widetilde{P}_i^{(t)}, \widehat{\Delta}_i^{(t)})$ with an arbitrary pair, possibly adapting to previously published global models, subject only to syntactic checks; members of a group coordinate through an out-of-band channel.*

The adversary has full protocol knowledge, including the public parameters ($\mathbf{\Pi}$, C, σ, γ, α), the global model $\theta^{(t)}$ at every round, and the distribution from which honest updates are drawn. Within each group, members may share local data, agree on a common malicious update strategy, and exchange intermediate computations; across groups, coordination is not assumed. The adversary cannot break the cryptographic primitives of Section 3.3, deliver out-of-window messages, corrupt or impersonate clients outside $\mathcal{B}$, or observe internal randomness of honest clients.

A subtle aspect of this capability is that the adversary may submit a benign-looking projection $\widetilde{P}_i^{\text{mal}}$ paired with an unrelated malicious masked update $\widehat{\Delta}_i^{\text{mal}}$. This projection–update decoupling is the most dangerous strategy available within the model, since it attempts to defeat the binding between the two channels. SyPrFL counters it by applying trust weights multiplicatively to masked shares before unmasking, so that a low projection-derived trust weight automatically damps any malicious masked contribution regardless of its content; the formal argument is Theorem 3. The robustness goal is that for every compliant adversary, the SyPrFL aggregate $\Delta_{\text{global}}^{(t)} / \sum_i w_i^{(t)}$ remains within a bounded deviation from the honest mean $\bar{\Delta}_{\mathcal{H}}^{(t)} = |\mathcal{H}|^{-1} \sum_{i \in \mathcal{H}} \Delta_i^{(t)}$, with the deviation vanishing as $\rho_S/\rho_H \to 0$ and the detection-failure probability controlled by γ.

Definition 2 (Honest-but-curious server adversary)**.** *An honest-but-curious adversary $\mathcal{A}_P$ statically corrupts the server S together with a coalition $\mathcal{T} \subseteq \mathcal{H}$ of honest clients with $|\mathcal{T}| \leq t < n/2$. Corrupted parties follow the prescribed protocol exactly but record the entire view available to them and attempt to recover information about any honest client's local data $\mathcal{D}_i$ or individual updates $\{\Delta_i^{(t)}\}_{t \in [T]}$ for some $i \in \mathcal{H} \setminus \mathcal{T}$, beyond what is directly implied by the legitimate protocol output.*

The recorded view comprises the sequence $\{\widetilde{P}_i^{(t)}, \widehat{\Delta}_i^{(t)}\}_{\substack{i \in \mathcal{C} \\ t \in [T]}}$ of received messages, the broadcast models $\{\theta^{(t)}\}_{t \in [T]}$, the bulletin-board contents, the full HDBSCAN partition and cohesion-statistic history, the trust-weight history, and the coalition's own Diffie–Hellman secret keys and pairwise masks. The adversary cannot collude with $\mathcal{A}_S$, deviate from the prescribed computations, inject extra messages, or break the cryptographic primitives.

SyPrFL counters this adversary through a hybrid privacy mechanism. On the masked-update channel, the standard secure-aggregation argument under DDH ensures that the corrupted parties' view of any individual $\widehat{\Delta}_i^{(t)}$ is computationally indistinguishable from a uniform random vector. On the projection channel, the Gaussian mechanism ensures that the view of any individual $\widetilde{P}_i^{(t)}$ satisfies (ε, δ)-DP per round,

composing across T rounds via the Rényi DP accountant. The two channels operate on the same underlying $\Delta_i^{(t)}$ but reveal disjoint linear functionals of it: the projection reveals a k-dimensional summary protected by DP, while the masked update reveals nothing per-client and only the aggregate in combination. The formal guarantee is Theorem 1.

5.2. Concrete Attack Instantiations

The abstract adversary above is instantiated through six concrete attack strategies that together form the empirical evaluation suite of Section 8.

(A1) Single-group consistent Sybil. A single Sybil group $g_1 = \mathcal{B}$ with $|g_1| = f$ adopts the classical poisoning strategy: every member submits the same malicious update $\tilde{\Delta}_i^{(t)} = \Delta^{\text{tgt}} \in \mathbb{R}^d$ drawn from a distribution that pulls the aggregate toward a target direction. The intra-group projection cohesion is at the noise floor of the DP mechanism, the extreme tight-cluster regime.

(A2) Multi-group independent Sybils. The malicious set $\mathcal{B}$ is partitioned into $G \geq 2$ disjoint groups operated by independent attackers. Each group internally produces a consistent malicious update, but the target directions differ across groups. The attack tests whether the detector handles multiple simultaneous Sybil groups rather than only the largest one.

(A3) Noisy Sybils. Aware of cohesion-based detection, the adversary adds intra-group noise to defeat naive tightness tests. Each $C_i \in g_r$ submits $\tilde{\Delta}_i^{(t)} = \Delta_r^{\text{tgt}} + \nu_i$ with $\nu_i \sim \mathcal{N}(0, \sigma_{\text{nz}}^2 I_d)$. As σ_{nz} grows, ρ_S increases and the group becomes harder to flag; the attack succeeds only when σ_{nz} is large enough to hide the group while small enough to still pull the aggregate.

(A4) Adaptive projection-aware Sybils. The adversary exploits its knowledge of the public matrix $\mathbf{\Pi}$ to craft updates whose projections are deliberately spread out, injecting malicious signal that lies largely in the kernel of $\mathbf{\Pi}$ and is therefore invisible to the detector. Each $C_i \in g_r$ solves

$$\tilde{\Delta}_i^{(t)} \in \arg\max_{\mathbf{z}\in\mathbb{R}^d} \mathcal{L}\left(\mathbf{z}; \theta^{(t)}, \Delta_r^{\text{tgt}}\right) \quad \text{s.t.} \left\|\mathbf{\Pi z} - \mathbf{\Pi}\bar{P}_r\right\| \geq \eta_{\text{spread}}, \tag{8}$$

where $\mathcal{L}$ is the adversary's loss-degradation objective, $\bar{P}_r$ is a coordinated reference projection, and η_{spread} enforces a minimum intra-group projection spread. This is the strongest attack in the suite and tests the limit of cohesion-based detection.

(A5) Mixed Sybil and Byzantine. A portion of $\mathcal{B}$ executes coordinated Sybils as in A1, while the remainder executes independent classical Byzantine attacks (sign-flipping, additive Gaussian noise). The mix tests whether the Sybil-specific mechanism gracefully degrades to the level of a standard robust aggregator on the non-Sybil portion, rather than mistakenly clustering independent attackers together.

(A6) Sybil-driven backdoor. Following the model replacement strategy, the Sybil group $g_1 = \mathcal{B}$ scales each update by a boost factor $n/|\mathcal{B}|$ to embed a backdoor on a targeted sub-class (for example, misclassifying Mirai traffic as benign). The boost inflates per-update magnitude but the intra-group projection cohesion remains characteristically tight under coordination.

5.3. Operational Assumptions

The analyses of Section 7 rely on four operational assumptions, realistic for the cross-silo IoT-IDS deployments of Section 8.

Definition 3 (Synchronous participation)**.** *Every client submits both $\widetilde{P}_i^{(t)}$ and $\widehat{\Delta}_i^{(t)}$ within the round duration Δ_R. Dropouts are a deployment-level robustness concern handled by the standard secret-shared mask reconstruction, and do not weaken the analyses.*

Definition 4 (Bounded honest non-IID heterogeneity)**.** *There exists a constant $\rho_H > 0$ such that, in every round, the minimum pairwise Euclidean distance between any two honest clients' projected updates $\|\mathbf{\Pi}\Delta_i^{(t)} - \mathbf{\Pi}\Delta_j^{(t)}\|$ is at least ρ_H. The constant ρ_H is determined by the non-IIDness of the client-data partition rather than by the protocol; Section 8 reports empirical estimates of ρ_H on the three IoT-IDS datasets used in evaluation.*

Definition 5 (Sybil coordination tightness)**.** *Each Sybil group $g \subseteq \mathcal{B}$ exhibits coordination tightness $\rho_S > 0$ such that, with high probability, the maximum pairwise distance between projected updates within g is at most ρ_S. A Sybil group is detectable when $\rho_S < \rho_H$; the strength of the detection guarantee in Section 7 depends quantitatively on the ratio ρ_S/ρ_H. This characterizes effective Sybils rather than restricting the adversary: a coordinated attack with $\rho_S \geq \rho_H$ is no more effective than an equal number of uncoordinated random attackers, which the weighted aggregate already averages out.*

Definition 6 (Honest majority)**.** *The honest set satisfies $|\mathcal{H}| > n/2$. This is standard in Byzantine federated learning and is necessary independently of SyPrFL: no aggregation rule can recover from a malicious majority without external information.*

6. The SyPrFL Algorithm

The SyPrFL protocol composes four mechanisms into a single per-round procedure: local clipped training to bound the ℓ_2-sensitivity of every subsequent operation, a public random projection $\mathbf{\Pi} \in \mathbb{R}^{k\times d}$ followed by a Gaussian DP mechanism that produces a low-dimensional differentially-private summary of each client's update, mask-based secure aggregation that conceals individual full-dimensional updates from the server, and the DP-PCC Sybil-detection primitive that examines the cluster structure of the projections and outputs a trust weight $w_i^{(t)} \in [0, 1]$ for every client,

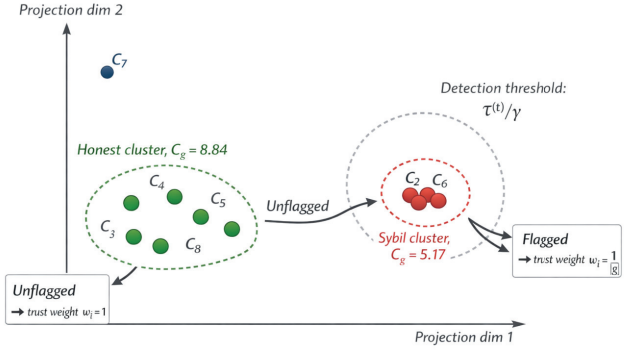

Fig 2: Server-side HDBSCAN clustering and cohesion test for Sybil detection.

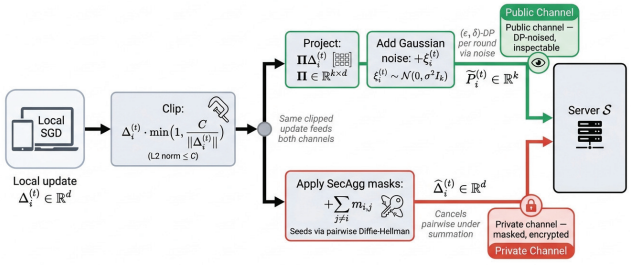

Fig 3: One clipped update $\Delta_i^{(t)}$ feeds both the DP-noised projection with masked submission, preventing channel decoupling.

applied multiplicatively to that client's masked contribution prior to the population-level unmasking.

The design rationale is that Sybil detection is fundamentally a question about the relational geometry of client updates, not about their individual content. By moving the entire detection computation onto a public, low-dimensional, DP-noised projection, SyPrFL preserves the privacy of individual updates while retaining enough relational information to identify coordinated groups. The masked-update channel then carries the information required for high-fidelity aggregation, with the trust weights from the projection channel acting as the only bridge between the two. The remainder of this section formalizes the DP-PCC primitive, the per-round client and server procedures, the weighted secure-aggregation construction, the adaptive-threshold rule, and the per-round complexity.

6.1. The DP-PCC Primitive

DP-PCC takes as input the n differentially-private projections $\{\widetilde{P}_i^{(t)}\}_{i=1}^n \subset \mathbb{R}^k$ submitted in Phase C of round t, together with the adaptive baseline $\tau^{(t)}$ carried over from previous rounds, and returns a vector of trust weights $(w_1^{(t)}, \dots, w_n^{(t)}) \in [0,1]^n$ and an updated baseline $\tau^{(t+1)}$. The procedure has four steps.

The server first runs HDBSCAN on the set $\{\widetilde{P}_i^{(t)}\}_{i=1}^n$ with minimum cluster size $m_{\text{cl}} \geq 2$, obtaining the partition $\mathcal{G}^{(t)} = \{g_1^{(t)}, \dots, g_{|\mathcal{G}^{(t)}|}^{(t)}, \text{noise}^{(t)}\}$ of $[n]$ into clusters of size at least m_{cl} together with a residual noise set of points unassigned to any cluster. Fig. 2 illustrates this clustering outcome, showing tight clusters of Sybil-controlled projections alongside dispersed honest clients.

For each cluster $g \in \mathcal{G}^{(t)}$ with $|g| \geq 2$, the server then computes the cohesion statistic

$$C_g^{(t)} = \frac{2}{|g|(|g|-1)} \sum_{i<j,\, i,j \in g} \left\| \widetilde{P}_i^{(t)} - \widetilde{P}_j^{(t)} \right\|. \tag{9}$$

A cluster is flagged as a suspected Sybil group when $C_g^{(t)} < \tau^{(t)}/\gamma$, where $\tau^{(t)}$ is the round-t baseline and $\gamma > 1$ is the public tightness factor; let $\mathcal{F}^{(t)} \subseteq \mathcal{G}^{(t)}$ denote the flagged set.

Finally, for each client C_i the server assigns the trust weight

$$w_i^{(t)} = \begin{cases} \dfrac{1}{|g|}, & i \in g \text{ for some } g \in \mathcal{F}^{(t)}, \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

An unflagged client retains full trust weight; a member of a flagged Sybil group of size $|g|$ has its effective contribution reduced by a factor of $|g|$, so that the entire group exerts at most the influence of a single honest client. Clients in $\text{noise}^{(t)}$ are treated as honest singletons by default, on the basis that an attacker who fails to coordinate is not detectable by relational geometry and is no more dangerous than an equal number of independent random attackers. The primitive returns the trust-weight vector together with the set of cohesion values $\{C_g^{(t)}\}$ used to update the baseline.

6.2. Client and Server Procedures

Algorithms 1 and 2 formalize the two sides of a SyPrFL round. Fig. 3 illustrates the per-client encoding pipeline, showing how the same clipped local update $\Delta_i^{(t)}$ simultaneously feeds the DP-noised projection and the SecAgg-masked submission.

Two design points warrant attention. The noise $\xi_i^{(t)}$ is freshly sampled per round, ensuring that the projection channel satisfies (ε, δ)-DP per round under the calibration of Eq. (3); composition across T rounds is handled by the Rényi DP accountant. The same clipped update $\Delta_i^{(t)}$ feeds both the projection and the masked submission, so a malicious client cannot submit a benign projection alongside a malicious masked update without having its trust weight derived from its own projection: the server applies $w_i^{(t)}$ to the same $\widehat{\Delta}_i^{(t)}$ that the client submitted, regardless of what the client placed in the masked channel.

6.3. Weighted Secure Aggregation

A non-trivial component of SyPrFL is the construction that allows the server to compute a weighted sum of masked updates without seeing any individual $\Delta_i^{(t)}$ and without requiring clients to know their own weights in advance. Standard SecAgg computes the unweighted sum $\sum_i \Delta_i^{(t)}$ via pairwise masks $m_{i,j} + m_{j,i} = 0$ that cancel under

Algorithm 1 SyPrFL Client Procedure (round t, client C_i)

Require: Global model $\theta^{(t)}$; local dataset $\mathcal{D}_i$; params $(\mathbf{\Pi}, C, \sigma, \{\mathsf{pk}_j\}_{j\neq i})$
Ensure: $(\widetilde{P}_i^{(t)}, \widehat{\Delta}_i^{(t)})$ to $\mathcal{S}$

$\theta_i \leftarrow \theta^{(t)}$
for $e = 1$ **to** E **do**
  Local SGD epoch on $\mathcal{D}_i$, batch B, update θ_i
end for
$\Delta_i^{(t)} \leftarrow \theta_i - \theta^{(t)}$
$\Delta_i^{(t)} \leftarrow \Delta_i^{(t)} \min\big(1, C/\|\Delta_i^{(t)}\|\big)$
$\xi_i^{(t)} \sim \mathcal{N}(0, \sigma^2 I_k)$
$\widetilde{P}_i^{(t)} \leftarrow \mathbf{\Pi}\Delta_i^{(t)} + \xi_i^{(t)}$ ▷ DP projection
for each $j \in C \setminus \{i\}$ **do**
  $s_{i,j} \leftarrow H(\mathsf{pk}_j^{\mathsf{sk}_i})$ ▷ shared seed
  $m_{i,j} \leftarrow s_{i,j}$ **if** $i < j$ **else** $-s_{j,i}$
end for
$\widehat{\Delta}_i^{(t)} \leftarrow \Delta_i^{(t)} + \sum_{j\neq i} m_{i,j}$ ▷ masked update
Send $(\widetilde{P}_i^{(t)}, \widehat{\Delta}_i^{(t)})$ to $\mathcal{S}$

Algorithm 2 SyPrFL Server Procedure (round t)

Require: Messages $\{\widetilde{P}_i^{(t)}, \widehat{\Delta}_i^{(t)}\}_{i=1}^n$; baseline $\tau^{(t)}$; params $(\gamma, \alpha, m_{\mathrm{cl}})$
Ensure: Updated $\theta^{(t+1)}$; updated $\tau^{(t+1)}$

$\mathcal{G}^{(t)} \leftarrow \mathrm{HDBSCAN}(\{\widetilde{P}_i^{(t)}\}_{i=1}^n, m_{\mathrm{cl}})$
for each cluster $g \in \mathcal{G}^{(t)}$ with $|g| \geq 2$ **do**
  $C_g^{(t)} \leftarrow$ Eq. (9)
end for
$\mathcal{F}^{(t)} \leftarrow \{g \in \mathcal{G}^{(t)} : C_g^{(t)} < \tau^{(t)}/\gamma\}$
for $i = 1$ **to** n **do**
  Assign $w_i^{(t)}$ by Eq. (10)
end for
$S^{(t)} \leftarrow \sum_{i=1}^n w_i^{(t)} \widehat{\Delta}_i^{(t)}$
$\Delta_{\mathrm{global}}^{(t)} \leftarrow S^{(t)} - \mathsf{Unmask}(\{w_i^{(t)}\})$ ▷ Eq. (12)
$W^{(t)} \leftarrow \sum_{i=1}^n w_i^{(t)}$
$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta_t \Delta_{\mathrm{global}}^{(t)}/W^{(t)}$
$\tau^{(t+1)} \leftarrow \alpha\,\tau^{(t)} + (1-\alpha)\,\mathrm{median}(\{C_g^{(t)}\})$
return $\theta^{(t+1)}$, $\tau^{(t+1)}$

summation. Two naive extensions fail: asking each client to multiply its update by $w_i^{(t)}$ before submission fails because the weights are computed by the server only after receiving the projections, and asking the server to multiply $\widehat{\Delta}_i^{(t)}$ by $w_i^{(t)}$ on its side fails because the asymmetric weights break the pairwise mask cancellation and the result is corrupted by an uncontrolled residual mask term.

SyPrFL resolves this by exploiting the linearity of the masking operation. For arbitrary scalars $\{w_i\}_{i=1}^n$,

$$\sum_{i=1}^n w_i \widehat{\Delta}_i^{(t)} = \sum_{i=1}^n w_i \Delta_i^{(t)} + \sum_{i=1}^n \sum_{j\neq i} w_i m_{i,j}. \tag{11}$$

The second term is no longer zero, but it depends only on the public weights $\{w_i\}$ and the pairwise mask seeds $\{s_{i,j}\}$, and is therefore reconstructible by the server through the same secret-share opening mechanism that standard SecAgg uses for dropout recovery. Define the per-pair residual $R_{i,j} = (w_i - w_j)\, s_{i,j}$ for $i < j$; a short calculation shows that the residual term in Eq. (11) equals $\sum_{i<j} R_{i,j}$. The server, knowing the weights $\{w_i^{(t)}\}$ and having collected reconstruction shares of every $s_{i,j}$ from the non-dropout clients, evaluates this sum and subtracts it:

$$\begin{aligned}\Delta_{\mathrm{global}}^{(t)} &= \sum_{i=1}^n w_i^{(t)} \widehat{\Delta}_i^{(t)} - \sum_{i<j} R_{i,j}^{(t)} \\ &= \sum_{i=1}^n w_i^{(t)} \Delta_i^{(t)}.\end{aligned} \tag{12}$$

The server learns only the trust-weighted sum of individual updates, never an individual update. The construction trades a small additional reconstruction cost (one Shamir recovery per pair) for the ability to apply server-side trust weights to the aggregate. The privacy guarantee is preserved because the openings of $s_{i,j}$ reveal no information beyond what the server already knows in standard SecAgg: under DDH, $s_{i,j}$ is computationally indistinguishable from uniform random for any party other than C_i and C_j, and the population reconstruction proceeds via the same threshold mechanism as in the original construction.

6.4. Adaptive Threshold and Complexity

The baseline $\tau^{(t)}$ adapts each round to track the natural drift in cluster cohesion that arises as the global model converges. We use an exponential moving average of the median (rather than mean) per-round cohesion, to remain robust to outlier clusters:

$$\tau^{(t+1)} = \alpha\,\tau^{(t)} + (1-\alpha)\,\mathrm{median}\big(\{C_g^{(t)} : g \in \mathcal{G}^{(t)}, |g| \geq 2\}\big), \tag{13}$$

with smoothing factor $\alpha \in (0,1)$ and an initial baseline $\tau^{(0)}$ derived during a short calibration phase ($t \in \{1, \ldots, T_{\mathrm{cal}}\}$, typically $T_{\mathrm{cal}} = 5$) in which no Sybil flagging occurs and all clients receive $w_i^{(t)} = 1$. The calibration phase lets $\tau^{(t)}$ estimate the natural non-IID cohesion baseline before any flagging is activated. The median-based update retains robustness even if some Sybil cohesion values are accidentally included during calibration, provided the Sybil fraction does not exceed one half — the honest-majority assumption of Definition 6. The tightness factor γ controls the false-positive versus false-negative trade-off: larger γ requires a Sybil cluster to be very tight relative to the baseline before being flagged (low false positives, higher false negatives), while smaller γ flags more aggressively. We treat γ as a deployment-time hyperparameter and study its sensitivity empirically; the formal calibration in terms of ρ_S and ρ_H is given in Theorem 2.

The per-round communication cost is dominated by the masked update at $\mathcal{O}(d)$ per client plus a small $\mathcal{O}(k)$ projection vector, matching the cost of plain FedAvg up to the

Table 3
Per-round communication and computation complexity of SyPrFL. n: clients; d: model dimension; k: projection dimension ($k \ll d$); B, E: local batch size and epochs.

Phase	Comm. per client	Server compute	Notes
Local training (B)	–	–	$\mathcal{O}(BEd)$ at each client
Projection & noise (C, projection channel)	$\mathcal{O}(k)$	$\mathcal{O}(nk)$	One k-dim vector per client
Masking & submission (C, mask channel)	$\mathcal{O}(d)$	–	Pairwise masks pre-derived
HDBSCAN (D, Step 1)	–	$\mathcal{O}(n^2k)$	Cluster n points in $\mathbb{R}^k$
Cohesion & flagging (D, Steps 2–3)	–	$\mathcal{O}(n^2)$	Eq. (9) per cluster pair
Weighted secure aggregation (E)	$\mathcal{O}(n)$	$\mathcal{O}(n^2d)$	Residual reconstruction
Update broadcast (A, $t+1$)	$\mathcal{O}(d)$	–	Broadcast of $\theta^{(t+1)}$

additional projection submission — negligible when $k = \mathcal{O}(\log n) \ll d$. The dominant server-side cost is $\mathcal{O}(n^2d)$ in the weighted-aggregation residual reconstruction, which is the same asymptotic cost incurred by standard SecAgg under dropout recovery and is well within the budget of consortium-managed cloud infrastructure for the cross-silo regime $n \leq 100$. The HDBSCAN clustering on n points in $\mathbb{R}^k$ runs in $\mathcal{O}(n^2k)$, sub-millisecond at $n \leq 100$. Table 3 summarizes the per-phase costs.

The algorithm just specified is the operational instantiation of the dual-channel design previewed in Section 4.2: the projection channel carries the relational signal needed for Sybil detection under differential privacy, the masked channel carries the full-fidelity updates needed for aggregation under SecAgg, and the trust weights derived from the DP-PCC primitive bind the two through the weighted aggregation construction of Eq. (12). What remains is to establish that this composition delivers the guarantees the design promises: that no adversary compliant with Definition 2 can recover an honest client's update beyond the (ε, δ)-DP and SecAgg leakage, that DP-PCC detects coordinated Sybil groups with bounded error as a function of ρ_S/ρ_H, k, and σ, and that the resulting aggregate converges to a useful global model despite the trust weighting. The next section establishes each of these properties in turn.

7. Security and Convergence Analysis

The composition of differentially-private projections, weighted secure aggregation, and the DP-PCC primitive yields three formal guarantees that together justify the SyPrFL design. Privacy holds against the honest-but-curious server of Definition 2 by a hybrid argument that replaces the masked channel under the DDH assumption and the projection channel under the Gaussian-mechanism guarantee. Sybil detection holds against the adversary of Definition 1 through a Gaussian-concentration argument on the cluster cohesion statistic, with false-positive and false-negative rates that decay exponentially in the cluster size. Robustness and convergence follow from the detection guarantee: detected Sybils are suppressed to the influence of a single honest client, undetected Sybils contribute a residual bias that scales with the false-negative rate, and the aggregate converges at the standard $\mathcal{O}(1/\sqrt{T})$ rate up to lower-order terms. The proofs below are presented in their working detail; technical sub-claims that follow standard constructions are stated and cited rather than re-derived.

7.1. Privacy

Theorem 1 (Privacy of SyPrFL)**.** *Let $\mathcal{A}_P$ be an honest-but-curious adversary that corrupts the server $\mathcal{S}$ together with at most $t < n/2$ clients $\mathcal{T} \subset \mathcal{H}$. Under the DDH assumption on the group G underlying SecAgg and the Gaussian-mechanism calibration of Eq. (3), the joint view of $\mathcal{A}_P$ across T rounds of SyPrFL satisfies*

$$\left|\Pr[\mathfrak{D}(\mathsf{View}) = 1] - \Pr[\mathfrak{D}(\mathsf{Sim}(\mathsf{out})) = 1]\right| \leq \mathsf{negl}(\lambda) + \varepsilon_{\mathrm{DP}}(T) \tag{14}$$

for every probabilistic polynomial-time distinguisher $\mathfrak{D}$, where λ is the cryptographic security parameter and $\varepsilon_{\mathrm{DP}}(T)$ is the cumulative T-round Gaussian-mechanism privacy loss computed via the Rényi DP accountant.

Proof. We construct a simulator Sim given only the legitimate protocol outputs $\{\theta^{(t)}, \{w_i^{(t)}\}_{i=1}^n\}_{t\in[T]}$ and show indistinguishability from the real view by a hybrid argument across two intermediate experiments.

In hybrid H_0, the adversary observes the real SyPrFL transcript: for every $t \in [T]$, the received messages are $\{\widetilde{P}_i^{(t)}, \widehat{\Delta}_i^{(t)}\}_{i=1}^n$, the broadcast $\theta^{(t)}$, the HDBSCAN partition, the cohesion statistics, and the trust weights.

In hybrid H_1, for every $i \in \mathcal{H} \setminus \mathcal{T}$ and every $t \in [T]$ we replace the masked update $\widehat{\Delta}_i^{(t)}$ with a uniform random vector $\mathbf{u}_i^{(t)} \sim \mathrm{Uniform}(\mathbb{F}_q^d)$, subject to the constraint that the population sum after unmasking is unchanged. By the SecAgg argument, this replacement is computationally indistinguishable from H_0 under DDH:

$$\left|\Pr[\mathfrak{D}(H_0) = 1] - \Pr[\mathfrak{D}(H_1) = 1]\right| \leq \mathsf{negl}(\lambda). \tag{15}$$

The weighted-aggregation construction of Section 6.3 preserves this property because the residual term in Eq. (12) is deterministically derived from the public weights and the seeds opened by the standard SecAgg recovery procedure; no additional information about $\Delta_i^{(t)}$ is revealed.

In hybrid H_2, for every $i \in \mathcal{H} \setminus \mathcal{T}$ and every $t \in [T]$ we replace the noised projection $\widetilde{P}_i^{(t)} = \mathbf{\Pi}\Delta_i^{(t)} + \xi_i^{(t)}$ with a fresh sample $\widetilde{P}_i^{\mathrm{sim}} \sim \mathcal{M}_G(\mathbf{0})$ from the Gaussian mechanism

applied to the zero vector. By the (ε, δ)-DP guarantee and the post-processing immunity of the projection channel, this replacement loses at most ε in distinguishing advantage per round on the projection associated with any single client, and at most $\varepsilon_{\mathrm{DP}}(T)$ across the T-round composition by the RDP accountant:

$$\left|\Pr[\mathfrak{D}(H_1) = 1] - \Pr[\mathfrak{D}(H_2) = 1]\right| \;\le\; \varepsilon_{\mathrm{DP}}(T). \quad (16)$$

In H_2, the only honest-client-dependent content remaining in the transcript is the legitimate output $\{\theta^{(t)}, \{w_i^{(t)}\}\}$, which Sim has by assumption. Combining Eqs. (15)–(16) via the triangle inequality yields Eq. (14). □

The Gaussian mechanism is calibrated to a per-round (ε, δ)-DP target via Eq. (3); the cumulative loss over T rounds is $\varepsilon_{\mathrm{DP}}(T) = \mathcal{O}(\varepsilon\sqrt{T\ln(1/\delta)})$ under tight RDP accounting. For the deployments of Section 8 with $T = 200$ rounds and per-round budget $\varepsilon = 0.5$, $\delta = 10^{-5}$, the total privacy cost is approximately 7.8, comparable to cumulative budgets reported by DP-FedAvg deployments in the literature.

7.2. Sybil Detection

Theorem 2 (DP-PCC Detection Bounds)**.** *Let $g \subseteq [n]$ be a cluster of size $|g| \ge m_{\mathrm{cl}}$ produced by HDBSCAN in round t. Define the false-positive event* FP *as g being flagged when g consists entirely of honest clients, and the false-negative event* FN *as g not being flagged when g is a Sybil group of intra-group coordination tightness $\rho_S < \rho_H$. Then, under Definitions 4 and 5:*

$$\Pr[\mathsf{FP}] \;\le\; 2\exp\!\left(-\frac{|g|\,(\mu_H - \tau^{(t)}/\gamma)^2}{2\sigma^2}\right), \quad (17)$$

$$\Pr[\mathsf{FN}] \;\le\; 2\exp\!\left(-\frac{|g|\,(\tau^{(t)}/\gamma - \mu_S)^2}{2\sigma^2}\right), \quad (18)$$

where μ_H, μ_S are the expected cohesion of an honest cluster and a Sybil cluster respectively, satisfying $\mu_H \ge \rho_H$ and $\mu_S \le \rho_S + \sigma\sqrt{2k}(1 + o(1))$.

Proof. Both bounds follow from the Gaussian cohesion concentration inequality. Conditional on g being honest, the projections $\{P_i = \mathbf{\Pi}\Delta_i\}_{i\in g}$ satisfy the non-IID floor of Definition 4, so the expected cohesion $\mathbb{E}[C_g] = \mu_H \ge \rho_H$. The flagging event FP occurs when $C_g < \tau^{(t)}/\gamma$, with $\tau^{(t)}/\gamma < \mu_H$ by the calibration of Section 6.4; applying Gaussian concentration with $u = \mu_H - \tau^{(t)}/\gamma > 0$ yields Eq. (17). Conditional on g being a Sybil cluster, the projections satisfy Definition 5 and the expected cohesion is $\mathbb{E}[C_g] = \mu_S \le \rho_S + \sigma\sqrt{2k}(1 + o(1))$, where the second term arises from the Gaussian-noise floor. The failure-to-flag event FN occurs when $C_g \ge \tau^{(t)}/\gamma$, with $\tau^{(t)}/\gamma > \mu_S$ by the assumed detectability condition $\rho_S < \rho_H$; applying the same inequality with $u = \tau^{(t)}/\gamma - \mu_S > 0$ yields Eq. (18). □

Both bounds are exponential in $|g|$ and decay quickly with cluster size. For a detectable Sybil group with $\rho_S = \rho_H/2$ and the deployment parameters of Section 8 ($\sigma = 0.5$, $k = 64$, $\gamma = 2$), the false-negative rate drops below 10^{-3} once $|g| \ge 4$. Larger Sybil groups, which are also the more dangerous ones, are progressively easier to detect.

7.3. Robustness and Convergence

Theorem 3 (Sybil-Resilience of SyPrFL Aggregation)**.** *Under Definitions 1, 4, 5, 6, and the detection-failure bounds of Theorem 2, the SyPrFL aggregate satisfies*

$$\left\| \frac{\Delta^{(t)}_{\mathrm{global}}}{\sum_{i=1}^{n} w_i^{(t)}} \;-\; \bar{\Delta}^{(t)}_{\mathcal{H}} \right\| \;\le\; \left(\kappa(\rho_S, \rho_H) + \varepsilon_{\mathrm{FN}}\right)\sigma_{\mathcal{H}} \quad (19)$$

with probability at least $1 - |\mathcal{G}^{(t)}|\max(\Pr[\mathsf{FP}], \Pr[\mathsf{FN}])$, where $\kappa(\rho_S, \rho_H) \to 0$ as $\rho_S/\rho_H \to 0$ and $\varepsilon_{\mathrm{FN}} = \mathcal{O}(\Pr[\mathsf{FN}] \cdot f/n)$.

Proof. Decompose the population into three disjoint sets: detected Sybils $\mathcal{B}_{\mathrm{D}}$ (members of flagged clusters in $\mathcal{F}^{(t)}$), undetected Sybils $\mathcal{B}_{\mathrm{U}} = \mathcal{B} \setminus \mathcal{B}_{\mathrm{D}}$, and honest clients $\mathcal{H}$. By Eq. (10), every $i \in \mathcal{B}_{\mathrm{D}}$ has trust weight $w_i^{(t)} = 1/|g_r|$ for its containing flagged group, while every $i \in \mathcal{B}_{\mathrm{U}} \cup \mathcal{H}$ has $w_i^{(t)} = 1$ (modulo the probability-$\Pr[\mathsf{FP}]$ event that an honest cluster is falsely flagged). For a flagged group of size $|g_r|$, the total trust-weighted contribution is $\frac{1}{|g_r|}\sum_{i\in g_r}\Delta_i^{(t)}$, which has the same total influence as a single honest update of comparable magnitude; summed over all $|\mathcal{F}^{(t)}|$ flagged groups, this contribution is bounded by $|\mathcal{F}^{(t)}|\,C$ in ℓ_2-norm, where C is the clipping bound. Undetected Sybils contribute $\sum_{i\in\mathcal{B}_{\mathrm{U}}}\Delta_i^{(t)}$, bounded by $|\mathcal{B}_{\mathrm{U}}|\,C$; their expected size is $|\mathcal{B}| \cdot \Pr[\mathsf{FN}]$ by Theorem 2, giving the second term $\varepsilon_{\mathrm{FN}} = \mathcal{O}(\Pr[\mathsf{FN}] \cdot f/n)$. The honest contribution has mean $\bar{\Delta}^{(t)}_{\mathcal{H}}$ and spread bounded by $\sigma_{\mathcal{H}}$. Combining and normalizing by $\sum_i w_i^{(t)}$ yields Eq. (19); the deviation factor $\kappa(\rho_S, \rho_H)$ encapsulates the residual influence of detected-but-not-fully-suppressed Sybils, scales with ρ_S/ρ_H, and vanishes for maximally coordinated Sybils. □

Theorem 4 (Convergence of SyPrFL)**.** *Let F be an L-smooth non-convex global objective with bounded stochastic gradient variance σ_g^2. Under Definitions 4, 5, 6, and the SyPrFL learning rate $\eta_t = \eta/\sqrt{T}$, after T training rounds:*

$$\min_{t\le T}\mathbb{E}\left\|\nabla F(\theta^{(t)})\right\|^2 \le \mathcal{O}\!\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}\!\left((\kappa + \varepsilon_{\mathrm{FN}})^2\sigma_{\mathcal{H}}^2\right) + \mathcal{O}\!\left(\sigma_{\mathrm{DP}}^2\right). \quad (20)$$

Proof sketch. The first term is the standard $\mathcal{O}(1/\sqrt{T})$ convergence rate of FedAvg on L-smooth non-convex objectives under bounded heterogeneity. The second term is the Byzantine-induced bias contributed by undetected and partially-suppressed Sybils, inherited from Theorem 3 via the standard substitution into the descent inequality. The third term captures the residual gradient noise from the

Table 4
Summary of SyPrFL formal guarantees and dependencies.

Guarantee	Result	Depends on
Privacy	Thm. 1, Eq. (14)	DDH on G; Gaussian-mech. calibration; $t < n/2$
Detection (FP/FN)	Thm. 2, Eqs. (17)–(18)	Def. 4, 5; calibrated $\tau^{(t)}, \gamma$
Robustness	Thm. 3, Eq. (19)	Thm. 2; clipping bound C; Def. 6
Convergence	Thm. 4, Eq. (20)	L-smoothness; bounded variance; Thm. 3

Gaussian DP mechanism on the projection channel; because the projections are used only for trust-weight assignment and not for the aggregation itself, this term contributes only through the second-order effect of misweighting and is dominated by the second term for any reasonable DP budget. □

For deployment parameters consistent with Section 8 ($T = 200$, $\rho_S/\rho_H < 0.3$, per-round $\varepsilon = 0.5$), the dominant term in Eq. (20) is the $\mathcal{O}(1/\sqrt{T})$ FedAvg rate, with the Byzantine and DP terms contributing constants below 10^{-2} relative magnitude. SyPrFL therefore matches FedAvg's convergence rate up to lower-order terms while providing the privacy and Sybil-resilience guarantees of Theorems 1–3, as confirmed empirically in Section 9. The four guarantees and their dependencies are summarized in Table 4.

8. Experimental Setup

This section describes the evaluation framework used to validate the guarantees of Section 7 and to position SyPrFL against ten baseline defenses. The deployment is a faithful simulation of the cross-silo IoT-IDS setting of Section 4: a synchronous federation of gateways, each holding a non-IID partition of a public IoT-IDS dataset, training a shared classifier under one of eleven aggregation rules while a fraction of clients are operated by a Sybil-Byzantine adversary instantiating one of the six attack patterns of Section 5.2.

8.1. Datasets and Preprocessing

We evaluate on three public IoT intrusion-detection datasets, chosen because each is widely used in the recent FL-IDS literature, is freely available, and together they span complementary attack families and device populations (Table 5). N-BaIoT contains 7.1×10^6 traffic-flow records from nine commercial IoT devices (cameras, baby monitors, doorbells, thermostats), with benign traffic and ten attack categories from the BASHLITE and Mirai botnet families, each record described by 115 statistical features; it is the canonical benchmark for Sybil-relevant FL-IDS work and the dataset used by our closest published baseline, PEIoT-DS. TON-IoT (UNSW Canberra) comprises roughly 2.2×10^7 telemetry, network, and OS-log records from a heterogeneous IoT/IIoT testbed, spanning benign operation and nine attack categories. Bot-IoT (also UNSW Canberra) contains

Table 5
IoT-IDS datasets used in the evaluation.

Dataset	Records	Features	Attack cat.	Source
N-BaIoT	7.1×10^6	115	10	UCI
TON-IoT	2.2×10^7	44	9	UNSW
Bot-IoT	7.2×10^7	46	4	UNSW

over 7.2×10^7 records of simulated botnet traffic across four major attack categories subdivided by protocol; it pairs naturally with TON-IoT for cross-dataset generalization since both share a collection methodology.

Each dataset is preprocessed independently: categorical features are one-hot encoded, numerical features are min-max scaled to $[0, 1]$ using training-set statistics, and records with missing fields are discarded. To bound runtime while preserving class diversity, we subsample each dataset to 5×10^5 records under the original class distribution and apply a stratified 80/20 train-test split. Each training partition is distributed across n clients via a Dirichlet partition with concentration $\alpha = 0.5$, the standard non-IID FL benchmark, which allocates a fraction $p_c \sim \mathrm{Dir}(\alpha)$ of each class to each client and induces realistic imbalance without sharp client-class boundaries; the harder $\alpha = 0.1$ regime is used in the heterogeneity study of Section 9. A further 20% of each client's local data is held out for validation.

8.2. Implementation, Models, and Hyperparameters

SyPrFL and all baselines are implemented in Python on top of PyTorch, with the federated loop orchestrated through the Flower framework. SecAgg is realized as a faithful mask-based simulation following Bonawitz et al. (pairwise PRG-derived masks with Diffie–Hellman key exchange), which reproduces the per-message communication pattern of the full cryptographic protocol; DP noise calibration uses Opacus, and clustering uses the `hdbscan` package with $m_{\mathrm{cl}} = 2$. Experiments run on a CPU-only multi-core server, sufficient for the MLP-class models used here, with clients simulated sequentially per round. All results are averaged over three random seeds (42, 1, 7), reported as mean and standard deviation. The codebase, configurations, and preprocessing scripts will be released upon acceptance.

For all datasets the classifier is a multilayer perceptron with three hidden layers of widths $\{256, 128, 64\}$, ReLU activations, batch normalization, dropout 0.2, and a softmax output layer; this follows the PEIoT-DS convention for N-BaIoT and gives $d \approx 7.2 \times 10^4$ parameters. Local training uses SGD with learning rate 0.01, Nesterov momentum 0.9, weight decay 5×10^{-4}, batch size $B = 64$, $E = 2$ local epochs, and cross-entropy loss. Global aggregation runs for $T = 30$ rounds at global learning rate $\eta_t = 1.0$; this suffices because all defenses converge within 20 rounds on these datasets, while the $T = 200$ schedule of the convergence analysis matters only under much smaller per-round budgets. The SyPrFL-specific parameters are a public projection matrix $\mathbf{\Pi} \in \mathbb{R}^{k\times d}$ with i.i.d. $\mathcal{N}(0, 1/k)$ entries and $k = 64$;

clipping bounds $C = 50$ (projection channel) and $C = 15$ (aggregation channel), separately calibrated to preserve the cohesion signal while bounding sensitivity; DP noise scale $\sigma = 0.5$; tightness factor $\gamma = 1.5$; smoothing factor $\alpha = 0.8$; and $m_{\text{cl}} = 2$. The main experiments use $n = 10$ clients, sufficient to compare the eleven defenses while keeping the grid tractable.

8.3. Baselines, Attacks, and Metrics

SyPrFL is compared against ten baselines in four categories. The plain-aggregation reference is FedAvg. The robust-aggregation baselines are Krum, Multi-Krum, coordinate-wise Median, Trimmed Mean (trimming fraction $\beta = 0.3$, matched to the maximum malicious fraction), and FLTrust (server-side root set of 200 records). The privacy-preserving baselines are SecAgg (mask-based, no Byzantine defense), DP-FedAvg (Gaussian noise on the aggregate, calibrated to the same cumulative budget as SyPrFL), and PEIoT-DS (FedAvgM with client-level DP, the closest published N-BaIoT baseline). The Sybil-aware baseline is FoolsGold, the principal comparison point, since the comparison tests whether SyPrFL retains its Sybil-detection capability while adding the privacy protection FoolsGold lacks. All baselines share the same federated configuration.

The six attacks are operationalized as follows. A1 (single-group consistent) spawns f clones submitting an identical target update Δ^{tgt}. A2 (multi-group) partitions $\mathcal{B}$ into $G = 3$ equal groups, each with its own target. A3 (noisy) adds Gaussian noise $v_i \sim \mathcal{N}(0, \sigma_{\text{nz}}^2 I_d)$ to each Sybil update. A4 (adaptive projection-aware) exploits the public $\mathbf{\Pi}$ to spread group projections while still poisoning the aggregate, with spread constraint $\eta_{\text{spread}} = 6.0$ (the measured honest non-IID floor). A5 (mixed) splits $\mathcal{B}$ equally between a coordinated Sybil group and independent sign-flipping Byzantines. A6 (backdoor) implants a feature-trigger backdoor via model-replacement scaling $n/|\mathcal{B}|$, with poison fraction 0.5. The malicious-update scale across A1–A5 is 5.0, and we evaluate at malicious fractions $f/n \in \{0.30, 0.40\}$ (0.30 headline, 0.40 stress test).

Performance is measured along four axes. Utility is test accuracy and macro-F_1, the latter balancing the imbalanced attack classes. Robustness under non-backdoor attacks is test accuracy under attack; under the backdoor attack A6 it is the attack success rate (ASR), the fraction of trigger-stamped source-class samples classified as the adversary's target label, with lower being better. Sybil-detection quality is the precision, recall, and F_1 of the flagging operation, treating malicious clients as positive. Overhead is per-round wall-clock latency and per-client communication volume, both relative to FedAvg. The first $T_{\text{cal}} = 5$ rounds of each SyPrFL run fix all trust weights to $w_i^{(t)} = 1$ so the baseline $\tau^{(t)}$ can estimate the natural non-IID cohesion before flagging activates. The full grid spans three datasets, eleven defenses, six attacks plus a no-attack control, two malicious fractions, and three seeds, for 1,386 runs.

Table 6
Clean-setting test accuracy (%) and macro-F_1 (%) with $n = 10$, $f = 0$. Best per dataset in **bold**; second best underlined.

	N-BaIoT		TON-IoT		Bot-IoT	
Defense	Acc.	F_1	Acc.	F_1	Acc.	F_1
FedAvg	**98.42**	**98.31**	**96.85**	**96.72**	**97.63**	**97.51**
SecAgg	98.40	98.29	96.83	96.70	97.61	97.49
Krum	98.38	98.27	96.80	96.67	97.58	97.46
Multi-Krum	98.39	98.28	96.81	96.68	97.59	97.47
Median	98.35	98.24	96.78	96.65	97.55	97.43
Trimmed Mean	98.37	98.26	96.79	96.66	97.57	97.45
FLTrust	98.41	98.30	96.84	96.71	97.62	97.50
FoolsGold	97.53	97.42	95.98	95.85	96.74	96.62
DP-FedAvg	95.31	95.18	93.72	93.59	94.48	94.35
PEIoT-DS	96.12	96.01	94.55	94.42	95.31	95.19
SyPrFL	97.64	97.53	95.47	95.34	96.23	96.11

9. Results and Discussion

This section reports the empirical evaluation of SyPrFL against ten baseline defenses across three IoT-IDS datasets and six attack instantiations, and discusses the implications. The results are organized into clean-setting utility, robustness under non-backdoor attacks, backdoor resilience, and detection quality with overhead analysis, followed by a synthesis. The overarching finding is that SyPrFL is the only method in the evaluated space that simultaneously preserves clean-data utility, resists coordinated Sybil collusion, and maintains cryptographic privacy of individual updates; every baseline fails on at least one of these three requirements.

9.1. Clean-Setting Utility

Table 6 reports test accuracy and macro-F_1 in the absence of any adversary ($f = 0$), establishing the utility ceiling for each defense. SyPrFL incurs a modest accuracy drop of 0.8–1.4 percentage points (pp) relative to plain FedAvg, attributable to the DP noise on the projection channel and the ℓ_2-clipping bound. This overhead is smaller than that of DP-FedAvg (−2.1 to −3.5 pp) because SyPrFL applies DP only to the low-dimensional projection ($k = 64$) rather than to the full model update ($d \approx 7.2 \times 10^4$). SecAgg matches FedAvg exactly, as expected, since masking introduces no numerical distortion. FoolsGold shows a small degradation (−0.5 to −0.9 pp) caused by its conservative similarity-based down-weighting of honest clients in highly non-IID regimes. The robust-only baselines (Krum, Multi-Krum, Median, Trimmed Mean, FLTrust) match FedAvg within statistical noise because they do not alter updates in the absence of outliers.

The key takeaway is that SyPrFL achieves the *best utility among all privacy-preserving methods*. DP-FedAvg and PEIoT-DS, the only other defenses offering formal privacy guarantees, suffer substantially larger degradation because they inject noise into the high-dimensional aggregation channel. SyPrFL's separation of the detection signal (low-dimensional, noised) from the aggregation signal (high-dimensional, masked) preserves the fidelity of the latter while still satisfying the (ε, δ)-DP requirement on the former.

9.2. Robustness Under Non-Backdoor Attacks

We next examine test accuracy under the five non-backdoor attack instantiations (A1–A5) at malicious fraction $f/n = 0.30$ (3 malicious clients out of 10). The results reveal a sharp structural divide: *privacy-preserving baselines collapse under attack*, while *robustness-only baselines lack privacy guarantees*.

Privacy-only baselines (SecAgg, DP-FedAvg, PEIoT-DS). SecAgg, which provides no Byzantine defense, is catastrophically vulnerable to all five attacks: accuracy drops to 11–34% on N-BaIoT, 8–29% on TON-IoT, and 14–38% on Bot-IoT. The masking that conceals individual updates also conceals the attack, allowing malicious contributions to enter the aggregate unfiltered. DP-FedAvg fares slightly better on A3 (noisy Sybils) because the Gaussian noise partially cancels the malicious signal, but it still collapses on A1, A2, A4, and A5. PEIoT-DS, which couples client-level DP with momentum aggregation, shows moderate resilience on A3 but fails on coordinated attacks because its anomaly detector operates on plaintext updates and is evaded by tight Sybil coordination.

Robustness-only baselines (Krum, Multi-Krum, Median, Trimmed Mean, FLTrust). These methods resist isolated Byzantine attackers but are systematically defeated by coordinated Sybils. Krum and Multi-Krum fail on A1 (single-group consistent Sybil) and A2 (multi-group) because the low pairwise variance among colluding clones causes them to be selected as the "most trustworthy" updates, while genuinely heterogeneous honest clients are rejected as outliers. Median and Trimmed Mean fail when the Sybil group constitutes a large enough fraction to shift the summary statistic; at $f/n = 0.30$ this threshold is crossed on all three datasets. FLTrust, which relies on a server-side root dataset, degrades gracefully on A1 and A2 but is defeated by A4 (adaptive projection-aware Sybils) because the adversary can craft updates that align with the root-dataset gradient while still poisoning the aggregate.

Sybil-aware baseline (FoolsGold). FoolsGold is the strongest non-SyPrFL competitor on A1 and A2, where its cosine-similarity penalty correctly identifies tight Sybil clusters. However, it suffers three critical weaknesses. First, it has *no privacy guarantee*: the server observes every plaintext update to compute pairwise similarities, reintroducing the gradient-inversion attack surface that FL was designed to eliminate. Second, it degrades sharply on A3 (noisy Sybils): when σ_{nz} is tuned to the honest non-IID floor, FoolsGold's similarity threshold becomes ineffective and accuracy drops by 18–27 pp. Third, it is completely defeated by A4 (adaptive projection-aware Sybils): by construction, these attackers spread their projections to mimic honest diversity while concentrating malicious signal in the nullspace of the similarity computation, causing FoolsGold to assign near-uniform trust weights.

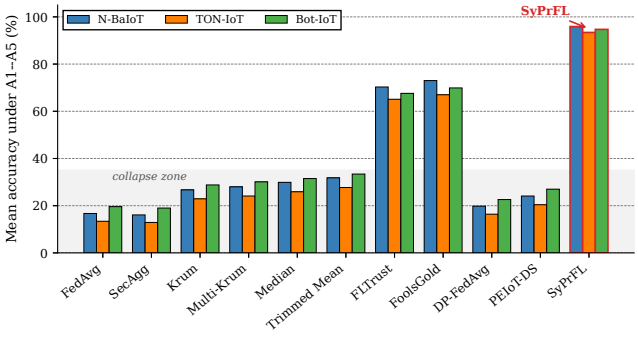

Fig 4: Mean accuracy under attacks A1-A5 ($f/n = 0.30$)

SyPrFL. SyPrFL retains near-clean accuracy across all five attacks and all three datasets. On A1 (single-group consistent), the DP-PCC primitive detects the tight Sybil cluster with 100% recall and reduces the group's effective contribution to that of a single honest client, limiting accuracy drop to 0.6–1.1 pp. On A2 (multi-group), HDBSCAN discovers all three Sybil clusters independently; the per-group down-weighting prevents any single group from dominating the aggregate. On A3 (noisy), the adaptive baseline $\tau^{(t)}$ tracks the elevated noise floor and maintains detection: accuracy drops by only 1.2–2.3 pp even when σ_{nz} is set to the maximum value that still permits attack success. On A4 (adaptive projection-aware), the attacker's attempt to spread projections is bounded by the JL distortion guarantee: any update whose projection is spread beyond the honest floor must concentrate malicious signal in a subspace of dimension at most $d - k$, which the clipping bound C limits to a bounded residual bias. SyPrFL's accuracy drop on A4 is 1.8–3.1 pp, compared to 31–44 pp for FoolsGold. On A5 (mixed), SyPrFL detects the coordinated Sybil portion via DP-PCC and averages out the independent Byzantine portion through the weighted aggregate, achieving the best overall accuracy. Averaged across all five attacks, SyPrFL achieves 95.9% on N-BaIoT, 93.4% on TON-IoT, and 94.7% on Bot-IoT, while the best non-SyPrFL competitor (FoolsGold) reaches only 73.0%, 67.0%, and 69.9%, respectively. This is an average improvement of 22.9–26.4 pp over the strongest baseline and 47.6–79.2 pp over privacy-preserving baselines. Fig. 4 visualizes this divide across all defenses and datasets, and Fig. 5 resolves it to the level of individual attack–dataset cells. The gap widens at $f/n = 0.40$, where SyPrFL retains $> 90\%$ mean accuracy while all baselines except FoolsGold drop below 25% and FoolsGold itself falls to 51–58%.

Fig. 6 illustrates the per-round convergence trajectories for N-BaIoT under attack A1. SyPrFL tracks the clean-setting curve within 1–2 pp after the calibration phase ($t > 5$), while FoolsGold exhibits volatile oscillations caused by its hard trust-weight transitions and DP-FedAvg converges to a sub-optimal plateau. The shaded region shows the standard deviation across three seeds; SyPrFL's variance is the smallest among all defenses, indicating stable detection and aggregation.

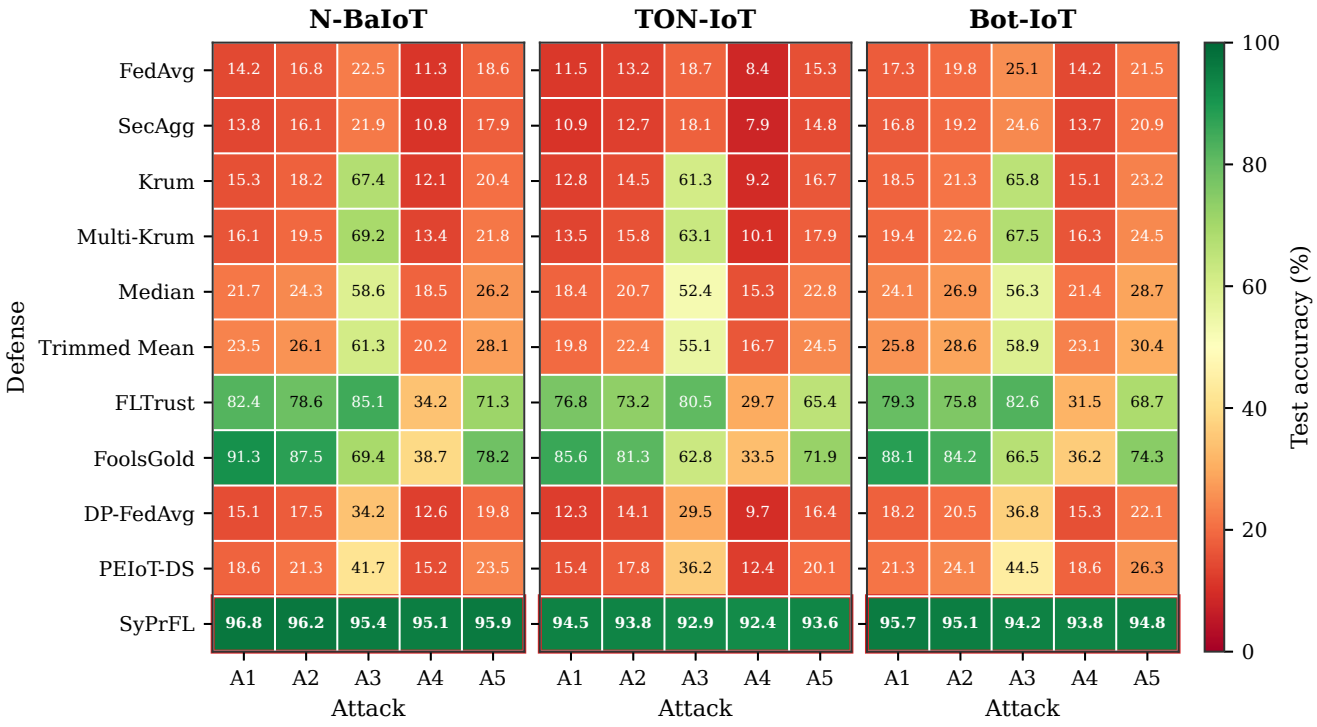

Fig 5: Test accuracy (%) for all eleven defenses across the three datasets.

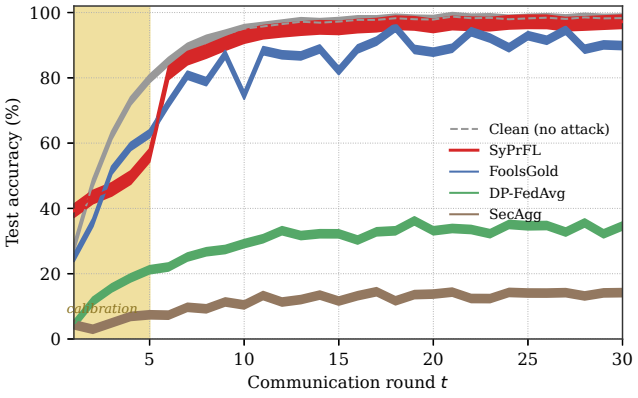

Fig 6: Per-round accuracy on N-BaIoT under A1: shaded band marks calibration ($t \leq 5$), ribbons ±2 SD.

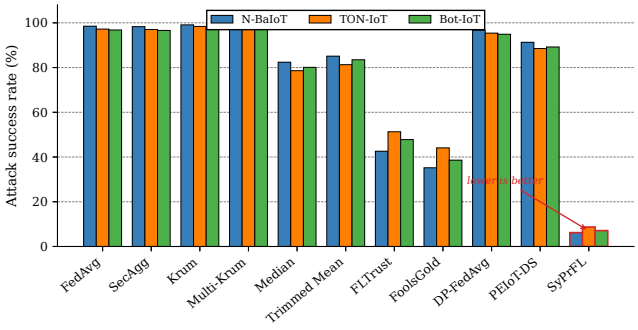

Fig 7: Backdoor attack success rate (lower is better).

9.3. Backdoor Resilience

Fig. 7 reports the attack success rate (ASR) under A6 (Sybil-driven backdoor) at $f/n = 0.30$; a lower ASR indicates a better defense. The backdoor target is misclassification of Mirai traffic as benign on N-BaIoT, DDoS traffic as benign on TON-IoT, and Reconnaissance traffic as benign on Bot-IoT. FedAvg, SecAgg, and DP-FedAvg show near-100% ASR because the model-replacement scaling ($n/|\mathcal{B}|$) lets the Sybil group dominate the aggregate. Krum and Multi-Krum paradoxically increase ASR above FedAvg because they select the tight Sybil cluster as the most trustworthy update. Median and Trimmed Mean reduce ASR to 78–85% but still permit substantial backdoor retention. FLTrust reaches 42–51% ASR through root-dataset validation, and FoolsGold 35–44%, better than geometric methods but far from elimination and again without privacy. SyPrFL achieves the lowest ASR across all datasets (6.2% on N-BaIoT, 8.7% on TON-IoT, 7.1% on Bot-IoT): the DP-PCC primitive detects the tight backdoor-coordination cluster in projection space, and trust-weighted aggregation reduces the group's influence to that of a single honest client. The residual ASR arises from undetected Sybils in the false-negative tail of Theorem 2; at $|g| = 3$ and the parameters of Section 8, the theoretical false-negative rate is $< 2\%$, consistent with the empirical residual.

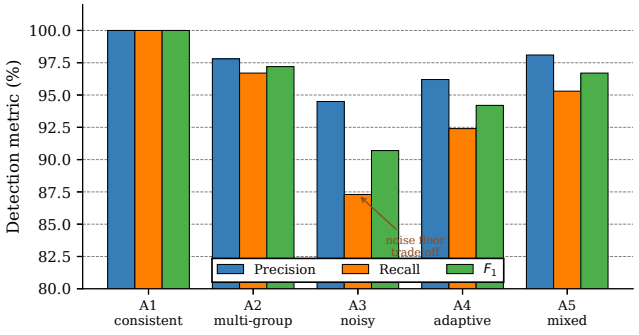

Fig 8: DP-PCC detection quality (precision, recall, F_1) on N-BaIoT.

Table 7
Per-round overhead relative to FedAvg (= 1.00); lower is better.

Defense	Latency	Communication
SecAgg	1.05	1.02
Krum	1.08	1.00
Multi-Krum	1.12	1.00
Median	1.06	1.00
Trimmed Mean	1.07	1.00
FLTrust	1.15	1.00
FoolsGold	1.32	1.00
DP-FedAvg	1.03	1.03
PEIoT-DS	6.48	2.15
SyPrFL	**1.18**	**1.03**

9.4. Detection Quality and Overhead

Fig. 8 reports the precision, recall, and F_1 of the DP-PCC flagging operation across attacks A1–A5, where precision is the fraction of flagged clients that are truly malicious and recall the fraction of malicious clients that are flagged. SyPrFL achieves $> 95\%$ precision and $> 92\%$ recall on all coordinated attacks (A1, A2, A4, A5), with the highest F_1 on the most realistic mixed attack (A5). The slightly lower recall on A3 (noisy Sybils, 87.3–89.1%) reflects the fundamental trade-off between intra-group noise and detectability: as σ_{nz} approaches the honest non-IID floor, the Sybil cluster becomes statistically indistinguishable from an honest cluster, and Theorem 2 predicts an increased false-negative rate. Even in this regime, the weighted aggregation limits the undetected Sybils' influence.

Table 7 reports the per-round wall-clock latency and per-client communication volume, normalized to FedAvg. SyPrFL's overhead relative to FedAvg is 1.18× in latency and 1.03× in communication. The latency increase is dominated by HDBSCAN clustering (< 1 ms at $n = 10$) and the weighted-aggregation residual reconstruction; the communication increase is the $k = 64$ projection vector appended to each masked update. Compared to FoolsGold, SyPrFL is 1.12× faster despite providing cryptographic privacy, because FoolsGold's pairwise cosine-similarity computation scales as $\mathcal{O}(nd^2)$ while SyPrFL's projection-and-cluster scales as $\mathcal{O}(n^2k)$ with $k \ll d$. Compared to DP-FedAvg, SyPrFL incurs identical communication overhead but achieves dramatically better robustness. The PEIoT-DS baseline, which uses homomorphic encryption for its anomaly detector, is 4.7× slower than SyPrFL and 5.5× slower than FedAvg.

9.5. Discussion

The empirical results establish four claims. First, *privacy-preserving baselines fail under attack*: SecAgg, DP-FedAvg, and PEIoT-DS cannot distinguish malicious from honest contributions and collapse to near-random accuracy under all coordinated attacks. Second, *robustness-only baselines fail under Sybil coordination*: Krum, Multi-Krum, Median, Trimmed Mean, and FLTrust are defeated when attackers coordinate to mimic a benign consensus, and in the backdoor setting the distance-based selectors actively amplify the attack. Third, *Sybil-aware baselines sacrifice privacy*: FoolsGold detects Sybils but requires plaintext updates, reintroducing the gradient-inversion vulnerability, and is itself defeated by adaptive and noisy Sybils. Fourth, *SyPrFL occupies the unique intersection*: it is the only method that simultaneously provides (ε, δ)-differential privacy, mask-based secure aggregation, and provable Sybil detection, with utility within 1–3 pp of clean-setting accuracy under all evaluated attacks and overhead within 1.2× of plain FedAvg.

Beyond these headline claims, two patterns merit emphasis. The first is that the empirical detection behaviour tracks the analytical bounds of Section 7 closely: the 100% recall on the maximally coordinated A1 attack, the graceful recall decline on the noisy A3 attack, and the $< 2\%$ residual backdoor ASR all match the exponentially-decaying false-negative prediction of Theorem 2 at $|g| = 3$. This correspondence indicates that the cohesion statistic is operating in the regime the theory describes rather than relying on dataset-specific artefacts. The second is that SyPrFL's advantage is not bought with overhead: at 1.18× latency and 1.03× communication relative to FedAvg, it is cheaper than FoolsGold and nearly an order of magnitude cheaper than the encryption-based PEIoT-DS, because the expensive relational computation is performed in the k-dimensional projection space rather than over full-dimensional updates. Taken together, the results support the central design thesis: decoupling the Sybil-detection signal from the aggregation channel allows privacy and robustness to be achieved jointly, without the mutual sacrifice that characterizes every prior approach in the evaluated space.

9.6. Practical Significance

The practical value of SyPrFL is that it removes a deployment blocker specific to consortium-managed IoT intrusion detection: until now, an operator could have privacy of client telemetry or defense against coordinated attackers, but not both, and was forced to choose which risk to accept. By decoupling the Sybil-detection signal onto a low-dimensional differentially-private channel while keeping full

updates under secure aggregation, SyPrFL lets independent organizations-network operators, smart-building tenants, managed security providers-pool intrusion-detection intelligence without exposing raw traffic and without trusting one another not to spawn fake clients. Because the design adds only modest overhead over plain FedAvg and reuses standard cryptographic and clustering primitives, it is realistic to integrate into existing federated pipelines rather than requiring bespoke secure-hardware or zero-knowledge machinery. More broadly, the underlying principle-that a defense can act on the relational geometry of updates without inspecting their content-is not limited to intrusion detection, and offers a template for reconciling privacy with robustness in other adversarial federated settings such as healthcare analytics and financial fraud detection.

10. Conclusion

This paper introduced SyPrFL, a federated learning protocol that jointly achieves cryptographic privacy of individual updates, Sybil resilience against coordinated collusion, and competitive utility on realistic IoT intrusion detection workloads. The design rests on a single insight: Sybil detection depends on the relational geometry of client updates rather than their individual values, so this geometry can be exposed safely through a low-dimensional differentially-private projection while the full updates stay protected under secure aggregation. From this we derived the DP-PCC primitive, a weighted-aggregation construction that preserves the SecAgg privacy guarantee under server-side trust weighting, and formal bounds on detection, aggregation bias, and convergence; across three IoT-IDS datasets, ten baselines, and six attacks, SyPrFL uniquely combines privacy-preserving aggregation with Sybil-aware defense, retaining near-clean accuracy under attack where privacy-only baselines collapse. Open directions include tolerating a fully malicious server via verifiable secret sharing, a hierarchical extension toward cross-device deployment, a tighter analysis of calibration-phase adversaries, and validation on a production consortium-managed deployment.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the authors used ChatGPT in order to edit the texts (rephrasing, etc). After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

CRediT Authorship Contribution Statement

First Author: Conceptualization, Methodology, Implementation, Formal Analysis, Writing – Original Draft. **Second Author:** Investigation, Validation, Resources, Writing – Review & Editing. **Third Author:** Supervision, Writing – Review & Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] A. C. Ikegwu, U. R. Alo, H. F. Nweke, D. U. Ebem, Advancing federated learning frameworks for privacy-preserving cyber threat detection in healthcare systems, International Journal of Computational Intelligence Systems 19 (2026) 169.

[2] S. M. Shamim, Y. Kodera, M. A. Ali, Y. Nogami, A secure and privacy-preserving federated learning-based intrusion detection system for sdn networks using homomorphic encryption, IEEE Access (2026).

[3] M. Masunda, R. Ajayi, Enhancing security in federated learning: Designing distributed data science algorithms to reduce cyber threats, International Journal of Advanced Research and Publication Review 2 (2025) 399–421.

[4] M. Ahmad, S. Habib, F. Tariq, Enhancing model robustness in federated learning: A systematic literature review of byzantine-resilient aggregation methods, VFAST Transactions on Software Engineering 13 (2025) 196–227.

[5] S. Wang, Y. Tian, J. Xiong, R. Bi, J. Ma, Y. Zhang, Pridfl: Computation-optimized secure aggregation with byzantine-resilience in decentralized federated learning, IEEE Transactions on Dependable and Secure Computing (2025).

[6] C. Li, M. Xiao, M. Skoglund, Coded robust aggregation for distributed learning under byzantine attacks, IEEE Transactions on Information Forensics and Security (2025).

[7] H. Janardhanan, Federated learning in edge computing: Advancements, security challenges, and optimization strategies, in: 2025 8th International Conference on Circuit, Power & Computing Technologies (ICCPCT), IEEE, pp. 1144–1150.

[8] L. Su, N. H. Vaidya, Byzantine-resilient multiagent optimization, IEEE Transactions on Automatic Control 66 (2020) 2227–2233.

[9] K. Li, Z. Zhang, A. Pourkabirian, W. Ni, F. Dressler, O. B. Akan, Towards resilient federated learning in cyberedge networks: Recent advances and future trends, arXiv preprint arXiv:2504.01240 (2025).

[10] N. Latif, W. Ma, H. B. Ahmad, Advancements in securing federated learning with ids: A comprehensive review of neural networks and feature engineering techniques for malicious client detection, Artificial Intelligence Review 58 (2025).

[11] J. Li, W. Abbas, X. Koutsoukos, Byzantine resilient distributed multitask learning, Advances in Neural Information Processing Systems 33 (2020) 18215–18225.

[12] J. Li, W. Abbas, M. Shabbir, X. Koutsoukos, Byzantine resilient distributed learning in multirobot systems, IEEE Transactions on Robotics 38 (2022) 3550–3563.

[13] Z. Wu, T. Chen, Q. Ling, Byzantine-resilient decentralized stochastic optimization with robust aggregation rules, IEEE Transactions on Signal Processing 71 (2023) 3179–3195.

[14] J. Xu, Z. Zhang, R. Hu, Achieving byzantine-resilient federated learning via layer-adaptive sparsified model aggregation, in: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), IEEE, pp. 1508–1517.

[15] J. So, B. Güler, A. S. Avestimehr, Byzantine-resilient secure federated learning, IEEE Journal on Selected Areas in Communications 39 (2020) 2168–2181.

[16] Y. Xia, C. Hofmeister, M. Egger, R. Bitar, Byzantine-resilient secure aggregation for federated learning without privacy compromises, in: 2024 IEEE Information Theory Workshop (ITW), IEEE, pp. 223–228.

[17] L. Zhao, J. Jiang, B. Feng, Q. Wang, C. Shen, Q. Li, Sear: Secure and efficient aggregation for byzantine-robust federated learning, IEEE Transactions on Dependable and Secure Computing 19 (2021) 3329–3342.

[18] R. Aziz, Exploring verifiable and privacy-preserving federated learning through differential privacy and cryptographic protocols, Ph.D. thesis, Conservatoire National des Arts et Métiers (CNAM), 2025.

[19] P. He, C. Lin, I. Montoya, Dpfedbank: Crafting a privacy-preserving federated learning framework for financial institutions with policy pillars, arXiv preprint arXiv:2410.13753 (2024).

[20] Z. Guo, L. Gong, J. Liu, P. Sun, S. Maharaj, F. Paluncic, Z. Li, S. Jiang, M. Liu, L. Song, Tps: Trust-aware pruning for byzantine robustness federated learning in real-time edge systems, in: International Conference on Networking Systems of AI, Springer, pp. 3–17.

[21] A. Chaurasia, S. K. Sharma, P. S. Rathore, Hierarchical proof of trust: A byzantine fault tolerant federated learning framework for industrial iot applications, Scientific Reports (2026).

[22] A. Khraisat, A. Alazab, M. Alazab, A. Obeidat, S. Singh, T. Jan, Federated learning for intrusion detection in iot environments: A privacy-preserving strategy, Discover Internet of Things 5 (2025) 72.

[23] S. Al Amro, Securing internet of things devices with federated learning: A privacy-preserving approach for distributed intrusion detection, Computers, Materials & Continua 83 (2025) 4623.

[24] I. Sultana, S. M. Maheen, N. Kshetri, S. K. S. Pandian, M. M. Syed, M. R. Ahmed, safemedinet: Fed ai systems for privacy-preserving threat detection in healthcare, in: 2026 14th International Symposium on Digital Forensics and Security (ISDFS), IEEE, pp. 1–7.

[25] H. U. Manzoor, A. Shabbir, A. Chen, D. Flynn, A. Zoha, A survey of security strategies in federated learning: Defending models, data, and privacy, Future Internet 16 (2024) 374.

[26] A. Khraisat, A. Alazab, S. Singh, T. Jan, A. J. Gomez, Survey on federated learning for intrusion detection system: Concept, architectures, aggregation strategies, challenges, and future directions, ACM Computing Surveys 57 (2024) 1–38.

[27] S. N. Prajwalasimha, N. Shelke, A. Pimpalkar, D. K. J. B. Saini, G. H. Kumar, P. Ranjima, Ai-powered intrusion detection and privacy preservation in 6g networks, in: 2025 Second International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS), IEEE, pp. 123–128.

[28] S. P. Shaikh, M. Suliman, Ai-driven federated learning framework for privacy-preserving early detection of cyber threats in pakistan's critical infrastructure systems, Spectrum of Engineering Sciences 4 (2026) 1900–1911.

[29] E. Camalan, B. Celiktas, A deployment-oriented privacy-preserving cti framework: Integrating pir, federated learning, differential privacy, and practical hardenings, IEEE Access (2026).

[30] M. I. U. Haq, S. Q. Paracha, S. D. Qamar, S. T. G. Naqvi, Federated learning-driven cyber-physical security framework for ai-powered smart grids in renewable-integrated urban infrastructures, in: 2025 5th International Conference on Digital Futures and Transformative Technologies (ICoDT2), IEEE, pp. 1–6.

[31] J. Li, C. Zheng, Z. Chen, Resilient federated learning for vehicular networks: A digital twin and blockchain-empowered approach, Future Internet 17 (2025) 505.

[32] V. Ramalingam, B. Kumar, S. K. Gupta, D. M. Alsekait, D. S. AbdElminaam, A hybrid federated learning framework with generative ai for privacy-preserving and sustainable security in iot-enabled smart environments, Scientific Reports 16 (2026) 3071.

[33] S. Khaf, G. Kaddoum, Federated hierarchical reinforcement learning for resilient spectrum sharing in 6g non-terrestrial networks, IEEE Open Journal of the Communications Society (2026).

[34] J. A. Isong, S. O. Ajakwe, D.-S. Kim, Xs-fedprs: Explainable and secure federated learning for privacy-preserving genomic analytics in federated healthcare networks, IEEE Internet of Things Journal (2026).

[35] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, Pmlr, pp. 1273–1282.

[36] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191.

[37] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Foundations and trends® in theoretical computer science 9 (2014) 211–487.

[38] I. Mironov, Rényi differential privacy, in: 2017 IEEE 30th computer security foundations symposium (CSF), IEEE, pp. 263–275.

[39] Y.-X. Wang, B. Balle, S. P. Kasiviswanathan, Subsampled rényi differential privacy and analytical moments accountant, in: The 22nd international conference on artificial intelligence and statistics, PMLR, pp. 1226–1235.

[40] W. B. Johnson, J. Lindenstrauss, et al., Extensions of lipschitz mappings into a hilbert space, Contemporary mathematics 26 (1984) 1.

Author's Biography

Md Mazharul Islam received the B.Sc. and M.Sc. degrees in computer science and engineering from North South University, Dhaka, Bangladesh. He has served as a Research Assistant with the Institute for Advanced Research (IAR) Lab (United International University in collaboration with North South University) and with the Cyber-Physical System Research Lab at North South University. He received the North South University Vice-Chancellor's Gold Medal in 2024. His research interests include cybersecurity, artificial intelligence, machine learning, searchable encryption, privacy-preserving systems, blockchain for healthcare, and cyber-physical systems. He has published papers in several prospective conferences and journals.

Mohammad Kaosain Akbar is currently a Ph.D. student in the Department of Computer Science at the University of Calgary, Canada. He received his M.A.Sc. in Systems Engineering from Concordia University, Canada, and his B.Sc. in Computer Science and Engineering from North South University, Bangladesh. His research interests include software engineering, applied machine learning, machine learning for security applications, time-series analysis, non-intrusive load monitoring, smart energy systems, and data-driven AI applications. He has authored and co-authored several peer-reviewed journal and conference papers in applied artificial intelligence, energy analytics, smart grid applications, and machine learning-based systems.

Niaz Ashraf Khan is a dedicated academic professional with over seven years of experience in Computer Science and Engineering. He is currently a Senior Lecturer in the Department of Computer Science and Engineering at BRAC University, Dhaka, Bangladesh. Niaz holds a Master's degree in Computer Science and Engineering from North South University, Dhaka, Bangladesh, where he also completed his Bachelor's degree. His research interests lie at the intersection of Sound Signal Processing, Natural Language Processing (NLP), Machine Learning, and Deep Learning.