

ATLAS-FL: Active Timing-Adversary-Resilient Secure Aggregation for Gateway-Assisted IoT, IIoT, and IoMT Federated Intrusion Detection

Mohammad Othman Nassar

Faculty of Information Technology - Cyber Security Department, Amman Arab University, Amman, Jordan

Abstract—Federated learning (FL) enables distributed intrusion detection without centralizing raw Internet of Things (IoT), Industrial IoT (IIoT), or Internet of Medical Things (IoMT) traffic. Secure aggregation protects update contents, but it does not authenticate the temporal context in which a protected contribution is admitted. An active network adversary may delay, suppress, replay, or slot-shift otherwise valid submissions, thereby shaping the accepted cohort, inflating survivor weights, and exposing participation through release traces. This study presents ATLAS-FL, an authenticated temporal-integrity layer for masking-based secure aggregation. The protocol binds client identity, round, slot, timing window, fresh nonce, and ciphertext digest through the Hash-based Message Authentication Code using SHA-256 (HMAC-SHA256); performs replay-cache and slot-consistency checks before Bonawitz-style mask recovery; fixes aggregation weights before arrivals are observed; and releases a fixed-length envelope at a committed time. A formal game-based analysis reduces temporal-binding forgery and cross-round replay to HMAC unforgeability and hash collision resistance, while proving no post-hoc survivor-weight inflation under the fixed-weight rule. The evaluation replicates the same protocol across Edge-IIoTset, N-BaIoT, CICIoMT2024, and WUSTL-EHMS-2020 and introduces FedBuff and FedBuff-SecAgg to separate benign straggler tolerance from adversarial timing integrity. Under the twenty-run profile, ATLAS-FL retains attacked macro-F1 of $93.1 \pm 0.4\%$ on Edge-IIoTset, $95.8 \pm 0.3\%$ on N-BaIoT, $94.9 \pm 0.4\%$ on CICIoMT2024, and $94.2 \pm 0.5\%$ on WUSTL-EHMS-2020. Accepted manipulated events remain at or below 6.8%, participation-inference area under the receiver operating characteristic curve (AUC) falls to 0.53 ± 0.02 , and weight inflation falls from 0.310 to 0.018 with 5.9% round-latency and 2.8% communication overhead at the evaluated cohort size.

Keywords—Federated learning; secure aggregation; active timing adversary; IoT intrusion detection; IIoT security; Internet of Medical Things; buffered asynchronous aggregation; differential privacy

I. INTRODUCTION

Intrusion evidence in Internet of Things (IoT), Industrial Internet of Things (IIoT), and Internet of Medical Things (IoMT) systems, including devices, gateways, industrial controllers, hospital monitoring centers, and edge services, can be collected in a centralized manner. This practice facilitates model training but increases costs associated with data exchange, while also potentially exposing operational or healthcare-related traffic. Federated learning addresses this issue by allowing participants to keep records local and share

model updates instead [1], [2]. Secure aggregation prevents the coordinator from inspecting an individual update [3]. However, these solutions are insufficient to guarantee that any protected information is not used to infer membership of an arbitrary subset of contributing parties during any round of the federation.

An adversary that can time events of the FL system can interfere with the process in ways independent of learning the content of any update or modifying its numeric value. Selective delays can cause removal of any particular client either with an exceptionally large contribution, thus skewing the statistics of the surviving ones, or as a member of a minority class, leading to renormalization of the class weights. Replay attacks can transfer a protected contribution from one round to another, while deliberate slot shifts can imprint the contribution content to incorrect time-dependent statistics. A fixed or variable drop time of the update can also allow inference of participation patterns of any clients, including those not contributing to the federation. Such opportunities for exploitation are particularly prevalent in IoT and IoMT due to the heterogeneity of the nodes, which are supported in the system through periodic communication, limited battery capacity, varying physical layer conditions, and contention in shared access points.

Timing considerations, therefore, touch upon three aspects of FL security and privacy. The first is the utilitarian value of the data, due to changes to either supervised labels or their weights. The second is the security during aggregation, as the compensation for dropped clients increases their effective representation in the federated model. The third is the confidentiality of participation in the federation, as timing of contributions is indicative of involvement in training even if the content of the updates is protected. Thus, timing interface attacks should be considered as threats to both data and model privacy, requiring independent evaluation of their impact on asynchronous optimization procedures and confidentiality of contributions.

Prior secure-aggregation work provides strong protection for update contents and malicious-input handling [3]–[8], while differential privacy controls information in released outputs [9], [10], [11]. Timing-defense research shows that encryption alone does not hide communication metadata [12], [13]. FedBuff addresses benign stragglers through buffered asynchronous aggregation and remains compatible with secure aggregation and DP [14]. TriSAFE is the closest integrated comparator because it combines transcript-bound secure

aggregation, DP, fixed-window timing defenses, and cover traffic in gateway-assisted IoT FL [15]. The unresolved question is whether authenticated temporal admission provides additional protection when the adversary actively selects who is delayed, dropped, replayed, or shifted.

The earlier ATLAS-FL revision evaluated IoT and IIoT benchmarks only. This version adds two genuine healthcare-security benchmarks. CICIoMT2024 contains 40 IoMT devices, including 25 physical and 15 simulated devices, 18 attacks, and Wi-Fi, MQTT, and Bluetooth traffic [16]. WUSTL-EHMS-2020 contains 16,318 healthcare-monitoring records with 35 network-flow features, eight biometric features, and one label [17]. Their inclusion changes IoMT from a prospective application into an empirical protocol-replication setting.

The study does not claim that the constituent primitives are new. Its contribution is the protocol-level coordination of authenticated schedule binding, timing admission, masking-aware rejection, fixed-weight aggregation, DP accounting, and cover-consistent release, together with formal security games and a four-benchmark comparison that includes an asynchronous baseline.

The study is organized around research objectives, formal security claims, and experimentally evaluated design objectives rather than empirical hypothesis statements.

The contributions are:

- A formal temporal-integrity definition and four security games covering schedule-to-ciphertext forgery, cross-round replay, slot substitution, and aggregate-release participation distinguishability.

- A complete protocol instantiation using X25519 [18], the HMAC-based Extract-and-Expand Key Derivation Function using SHA-256 (HKDF-SHA256) [19], and the Hash-based Message Authentication Code using SHA-256 (HMAC-SHA256) [20], [21], ChaCha20-derived pairwise masks [22], Shamir recovery [23], fixed-point encoding, and aggregate-level Gaussian differential privacy (DP).
- An auditable timing-risk controller with hard cryptographic overrides and validation-calibrated thresholds for accept, buffer, cover-trigger, and reject decisions.
- A no-renormalization aggregation rule that preserves pre-declared client coefficients and admits an explicit bound on missing-information deviation and client-level sensitivity.
- A four-benchmark evaluation spanning IoT, IIoT, and IoMT, with device-native and simulated-client non-IID partitions, participation-inference attacks, component ablations, and cryptographic/operational profiling.
- A matched comparison with FedBuff and FedBuff-SecAgg that distinguishes benign asynchronous efficiency from resilience to adversarially induced staleness and transcript manipulation.

A. Research Objectives and Evidence Map

The study is guided by research objectives derived from the identified security and systems gaps. These objectives are not empirical hypotheses; rather, they define the formal guarantees, experimental questions, and operational trade-offs evaluated in the subsequent sections. Table I maps each research objective to its primary formal or empirical evidence.

TABLE I. RESEARCH-OBJECTIVE TRACEABILITY

ID	Research objective/evaluation question	Primary evidence
RO1	To what extent does ATLAS-FL preserve intrusion-detection utility when selected clients are delayed, dropped, replayed, slot-shifted, or forced beyond their committed deadline?	Tables XIII and XV; Fig. 7–9
RO2	What formal guarantees does authenticated schedule-to-ciphertext binding provide against cross-round replay, slot substitution, and temporal-context forgery under the stated cryptographic assumptions?	Section V; Table VII; Fig. 2; Appendix A
RO3	How effectively do fixed pre-declared weights prevent survivor-weight inflation and limit DP-relevant aggregate instability under selective client absence?	Section V-D; Tables XVII and XIX; Fig. 4 and 11
RO4	To what extent does fixed-time, fixed-length encrypted zero-cover release reduce participation inference under the defined release-observer model?	Section V-E; Tables XVI and XIX; Fig. 10
RO5	How consistently do the timing-resilience and aggregate-stability effects replicate across IoT, IIoT, and genuine IoMT benchmarks?	Tables XIII and XVIII; Fig. 7
RO6	What utility, staleness, and security trade-offs arise when ATLAS-FL is compared with FedBuff and FedBuff-SecAgg under benign stragglers and strategically induced staleness?	Table XIV; Fig. 8; Table XXII
RO7	What cryptographic, communication, latency, CPU, memory, and scalability costs are introduced at the gateway?	Tables XX–XXI; Fig. 13

II. RELATED WORK AND EXACT RESEARCH GAP

A. Secure Aggregation and Malicious-Input Validation

Several studies have developed methods for secure aggregation in federated learning, including the pairwise-masking secure aggregation with dropout recovery from Bonawitz et al. [3]. Additional methods such as RoFL, ACORN, Armadillo, ELSA, and Aion introduce features for

validating participant input, identifying malicious inputs from participants, and ensuring robustness over multiple rounds of learning [4]–[7], [24]. The Lotto system specifically protects the server from malicious participants by validating the set of participants that contribute to the aggregation [25]. While these approaches have established high standards for the security and correctness of federated learning protocols, most of these secure aggregation systems primarily focus on validating

participant inputs and determining the set of participants rather than on securing the temporal context of the aggregated information. Complementary work addresses verifiable secure aggregation [26], decentralized adversarial-resilient routing [27], and communication-threat mitigation through moving-target defense [28]; these protect adjacent robustness properties rather than slot-specific temporal admission.

ATLAS-FL, however, works in tandem with existing secure aggregation systems, enabling them to authenticate the temporal context of the data before it is accepted into the system for masking and aggregation. This temporal authentication process occurs before a masked update is deleted to recover from a timed-out participant. Deleting masked data without proper recovery would leave uncanceled masks in the system, potentially compromising security.

B. Differential Privacy and Fixed Aggregation Sensitivity

Differential privacy ensures a limited influence of any one data subject on the output of an algorithm [9], [10], [11]. Differential privacy securely aggregates data by adding noise to the aggregated output. However, an adversary could modify the set of participants in the network and still have the server renormalize the weights of the surviving participants. This process would change the sensitivity of the aggregation function in ways that are difficult to predict. ATLAS-FL fixes the coefficients of the aggregation function before the arrivals of the data subjects are observed and uses a Rényi-DP accountant to determine the privacy budget of the aggregation function.

While this technique does not replace differential privacy, it removes one of the pathways by which the temporal aspects of the aggregation function could change.

C. Asynchronous and Buffered Federated Learning

FedBuff is a method for efficiently buffering and aggregating asynchronous updates from participants in a federated learning network, allowing the system to continue working even with clients that have varying completion times [14]. Similar concepts have been explored in other studies focusing on the dynamics of message queues and the impact of delayed updates [33]. While these methods are effective at handling clients that are slower to complete tasks and in increasing the overall throughput of federated learning, they do not provide authentication for the participants or allow for verification of a participant's assigned round, slot, or deadline. Furthermore, they may accept updates from clients that are attempting to manipulate the system with delayed or replayed updates as ordinary asynchronous contributions.

Heterogeneous federated optimization has also motivated FedProx, FedNova, adaptive federated optimization, and SCAFFOLD [29]–[32], which address optimization under heterogeneous clients rather than temporal-authentication guarantees.

FedBuff is therefore limited in its ability to deal with clients attempting to maliciously delay or replay updates. A version of FedBuff that uses a masked and differentially private backend for aggregation does not, however, exhibit these limitations.

D. Timing Privacy and Release Defenses

Research into systems that protect the timing privacy of communication such as Loopix [12] and adaptive-padding [13] demonstrates that communication timing reveals information about the participants and their activities, even if their messages are encrypted. While ATLAS-FL uses some of the same concepts for timing privacy, it focuses only on securing the aggregate-release interface rather than on providing overall anonymity for the communication network.

E. Federated IDS in IoT, IIoT, and IoMT

Edge-IIoTset and N-BaIoT support IoT/IIoT and botnet-oriented validation [34], [35]. Recent FL-IDS studies examine privacy, optimization, energy, and anomaly detection [36], [37], [38], [39], [40], [41], [42]. CICIoMT2024 and WUSTL-EHMS-2020 provide genuine IoMT contexts [16], [17]. The present work uses these datasets to test whether the aggregation-layer timing effect replicates beyond one traffic benchmark; it does not perform zero-shot feature transfer across incompatible schemas. Broader recent work includes privacy-preserving multimodal healthcare FL [43], local-global anomaly detection [44], IoT intrusion-detection surveys [45], peer-to-peer FL surveys [46], and multi-institutional radiological FL [47].

F. Closest Integrated Comparator: TriSAFE

TriSAFE already combines transcript-bound secure aggregation, DP, fixed-window batching, timing defenses, and cover traffic [15]. ATLAS-FL therefore cannot claim novelty from padding or fixed weights alone. Its narrower incremental contribution is slot-specific schedule-to-ciphertext authentication, an explicit active timing-adversary game, cryptographically ordered rejection and mask recovery, controller-aware mixed-attack evaluation, direct participation-inference testing, and comparison against buffered asynchronous FL. Table II summarizes this conservative positioning relative to the closest work.

TABLE II. CONSERVATIVE POSITIONING AGAINST CLOSEST WORK

Work	Primary protection	Timing treatment	Boundary relative to ATLAS-FL
Bonawitz SecAgg [3]	Update confidentiality and dropout recovery	Ordinary cohort timing	No active timing-admission property
RoFL/ACORN/Armadillo/ELSA/Aion [4]–[7], [24]	Malicious-input validity and robust SecAgg	Protocol disruption/dropout	No slot-specific active timing model
DP-secure aggregation [8], [9], [10], [11]	Formal privacy of released aggregates	Timing usually external to accounting	Does not prevent adversary-induced renormalization
FedBuff [14]	Buffered asynchronous optimization	Benign staleness and throughput	No temporal authentication or cover release

Work	Primary protection	Timing treatment	Boundary relative to ATLAS-FL
Loopix/Tor defenses [12], [13]	Cover traffic and timing privacy	Communication anonymity	No FL aggregation
TriSAFE [15]	Transcript-bound SecAgg, DP, timing defenses	Fixed-window timing protection	Closest comparator; active slot-level admission not isolated
ATLAS-FL	Authenticated temporal admission, fixed weights, cover release, formal games	Active delay/drop/replay/slot/late/adaptive attacks	Target contribution; bounded to defined observer and gateway model

III. SYSTEM MODEL, THREAT MODEL, AND DESIGN OBJECTIVES

A. Gateway-Assisted System

The system consists of a gateway G and a round-specific selected cohort S_t , where $K_t = |S_t|$. A client may represent an IoT device, an IIoT data source, an IoMT device, or an explicitly documented logical healthcare-sensor partition. Client i holds a local training dataset $\mathcal{D}_i$, receives the global model w_t , performs E local epochs, clips its model update to an L_2 -norm bound C , and constructs a masked and authenticated contribution. The gateway coordinates client selection, schedule commitment, temporal admission, mask recovery, and aggregate release, but it never observes an accepted client update in plaintext.

Before any round- t arrivals are observed, each selected client is assigned a fixed aggregation coefficient based on its verified effective sample mass q_i , where $q_i = \min(|\mathcal{D}_i^{\text{train}}|, n_{\max})$:

$$\alpha_{i,t} = \frac{q_i}{\sum_{j \in S_t} q_j} \sum_{i \in S_t} \alpha_{i,t} = 1 \quad (1)$$

The coefficients in Eq. (1) are committed before the gateway observes the admitted-client set $G_t \subseteq S_t$. They are therefore not recomputed or renormalized after delays, rejections, or dropouts. For every missing or rejected client $i \in$

$S_t \setminus G_t$, ATLAS-FL preserves the corresponding coefficient through an encrypted zero-cover contribution, as defined in Section IV.

B. Protected Assets and Observable Timing Surface

Table III summarizes the protected assets, observable timing surfaces, associated risks, and ATLAS-FL controls.

C. Active Timing Adversary

The adversary can delay selected submissions, suppress them, replay a prior protected transcript, shift a valid submission to a different slot, force an update beyond the admission deadline, or combine these actions adaptively. Targets may be selected uniformly, by pre-declared client weight, or by minority/attack-class concentration. The adversary can observe aggregate-release timing but cannot break the cryptographic primitives, compromise the gateway in the main model, or control enough clients to cross the secure-aggregation reconstruction threshold.

$$\tilde{T}_{i,t}^{\text{arr}} = \Phi(\mathcal{W}_{i,t} a_{i,t} \xi_{i,t}), \quad (2)$$

Here $\tilde{T}_{i,t}^{\text{arr}}$ is the committed timing window, $a_{i,t}$ is the adversarial action, and $\xi_{i,t}$ is benign network jitter.

D. Design Objectives

Table IV states the security and operational objectives, while Fig. 1 summarizes the architecture, trust domains, and active timing surface.

TABLE III. PROTECTED ASSETS, OBSERVABLES, AND CONTROLS

Element	Status	Primary risk	ATLAS-FL treatment
Local data	Protected	Direct privacy loss	Never leaves client
Plain update	Protected	Gradient/model leakage	Masked before transmission
Client identity in schedule	Protocol metadata	Participation inference	Authenticated schedule + padded release
Arrival time	Observable/manipulable	Delay and admission bias	Risk controller
Missing/rejected state	Observable internally	Weight inflation and release leakage	Fixed weights + zero cover
Aggregate release trace	Observable	Participation inference	Fixed time and fixed-length envelope

TABLE IV. SECURITY AND OPERATIONAL OBJECTIVES

Goal	Definition
G1	Secure-aggregation confidentiality compatibility
G2	Temporal authenticity and slot binding
G3	Freshness and cross-round replay rejection
G4	No post-hoc survivor-weight inflation
G5	Client-level DP-relevant sensitivity stability
G6	Reduced aggregate-release participation leakage
G7	Utility preservation under atomic and adaptive timing attacks
G8	Moderate cryptographic and gateway overhead

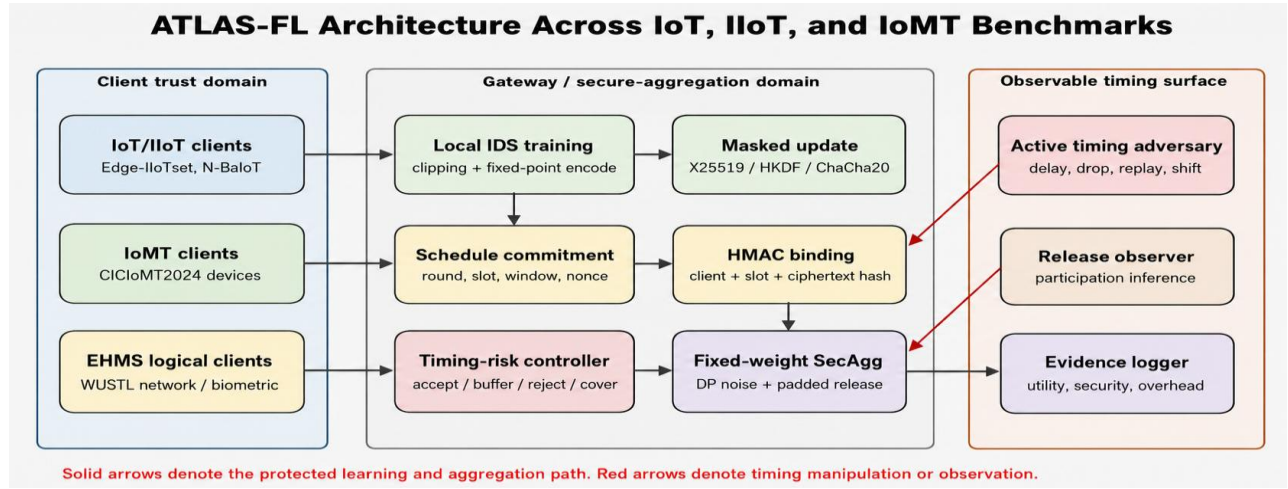


Fig. 1. ATLAS-FL architecture, trust domains, and active timing surface.

IV. ATLAS-FL PROTOCOL

A. Cryptographic Setup and Schedule Commitment

Each client i shares a long-term authentication key k_i with the gateway. For every secure-aggregation cohort, clients use X25519 to establish pairwise secrets [18], derive domain-separated keys through HKDF-SHA256 [19], generate pairwise masks using ChaCha20 [22], and distribute Shamir shares for dropout recovery [23]. HMAC-SHA256 follows the RFC 2104 construction and its standard security assumptions [20], [21]. ATLAS-FL assumes authenticated enrollment and lifecycle management by an external trusted device/key-management service. Enrollment, key rotation, revocation, compromised-device recovery, device replacement, and authentication-state synchronization are outside the protocol security boundary; a re-enrolled device must receive fresh authentication and nonce state before participating.

$$\tau_{i,t} = \text{HMAC}_{k_i} \left(\text{Encode} \left(i, t, s_{i,t} T_{i,t}^{\min} T_{i,t}^{\max} \rho_t, H(c_{i,t}) \right) \right) \quad (3)$$

The fresh nonce ρ_t prevents cross-round token reuse. The ciphertext digest binds the tag to the exact protected contribution. The gateway verifies the tag, window, round, slot, nonce freshness, ciphertext hash, and replay-cache state before the submission can enter the accepted set. This authentication binds the declared temporal metadata to the protected contribution; it does not constitute a trusted proof of the client's physical transmission instant. Temporal compliance is evaluated separately from the gateway-observed arrival time by the timing controller.

B. Local Update Clipping and Masking

$$\bar{\Delta}_{i,t} = \Delta_{i,t} \min \left(1, \frac{c}{\|\Delta_{i,t}\|_2} \right) \quad (4)$$

$$m_{i,t} = \text{PRG}(b_{i,t}) + \sum_{j \in S_t} \text{PRG}(s_{ij,t}) - \sum_{j < i} \text{PRG}(s_{ji,t}) \quad (5)$$

$$c_{i,t} = [\text{Enc}_{\text{fp}}(\alpha_{i,t} \bar{\Delta}_{i,t}) + m_{i,t}] \bmod q \quad (6)$$

Weighting occurs before masking so that reconstruction produces the fixed weighted sum directly. Enc_{fp} denotes

deterministic fixed-point encoding under the modular field used by the masking implementation. Let S denote the fixed-point scale and q the aggregation modulus. Round-to-nearest encoding introduces at most $1/(2S)$ absolute quantization error per encoded weighted coordinate, while correct centered decoding requires the encoded aggregate to remain strictly within $(-q/2, q/2)$, preventing modular wraparound. The no-wrap condition is checked before aggregation. Gaussian DP noise is calibrated to the aggregate sensitivity separately from this deterministic encoding error, and no convergence benefit is attributed to fixed-point arithmetic itself.

C. Timing Deviation and Controller Decision

$$D_{i,t} = \frac{\max\{0, T_{i,t}^{\min} - \bar{T}_{i,t}^{\text{arr}}, T_{i,t}^{\text{arr}} - T_{i,t}^{\max}\}}{T_{i,t}^{\max} - T_{i,t}^{\min}} \quad (7)$$

$$r_{i,t} = \lambda_1 D_{i,t} + \lambda_2 R_{i,t} + \lambda_3 H_{i,t} + \lambda_4 Q_{i,t} + \lambda_5 P_{i,t} \quad (8)$$

The controller coefficients are (0.31, 0.27, 0.14, 0.20, 0.08). Replay, HMAC failure, stale nonce, or authenticated slot mismatch is a hard rejection independent of the continuous score.

$$d_{i,t} = \begin{cases} \text{accept} & r_{i,t} < \theta_1 \\ \text{buffer} & \theta_1 \leq r_{i,t} < \theta_2 \\ \text{cover-trigger} & \theta_2 \leq r_{i,t} < \theta_3 \\ \text{reject} & r_{i,t} \geq \theta_3 \end{cases} \quad (9)$$

The thresholds are $\tau_1=0.28$, $\tau_2=0.52$, $\tau_3=0.74$ and are selected on calibration rounds that are disjoint from the final attack schedules. Fig. 2 summarizes the authenticated round sequence, and Fig. 3 shows the timing-risk controller decision flow.

D. Rejection, Dropout Recovery, and Aggregate Reconstruction

After the admission window closes, the gateway publishes the accepted set G_t and the recovery set M_t . Surviving clients reveal only the Shamir shares needed to remove masks associated with clients in M_t . Timing rejection is therefore handled as a cryptographic dropout after authentication has failed. The round aborts if fewer than the configured reconstruction threshold respond.

$$A_t^{\text{ATLAS}} = \sum_{i \in G_t} \alpha_{i,t} \bar{A}_{i,t} + \sum_{i \in M_t} \alpha_{i,t} C_{i,t} \quad (10)$$

The primary cover contribution is $C_{(i,t)} = 0$. It preserves the declared coefficient mass but does not fabricate missing information. The released aggregate is

$$w_{t+1} = w_t + \eta_g (A_t + z_t) z_t \sim \mathcal{N}(0, \sigma^2 \Delta_d^2 I_d) \quad (11)$$

$$\Delta_2 \leq \max_{i \in S_t} \alpha_{i,t} C_{i,t} \cdot (\text{add/remove client adjacency}) \quad (12)$$

For replace-one adjacency, the bound is doubled. The privacy guarantee additionally requires the stated sampling process, Gaussian mechanism, and accountant; fixed weights alone are not a privacy proof. Fig. 4 illustrates the fixed-weight contrast after selective absence.

Authenticated ATLAS-FL Round: Message and Timing Sequence

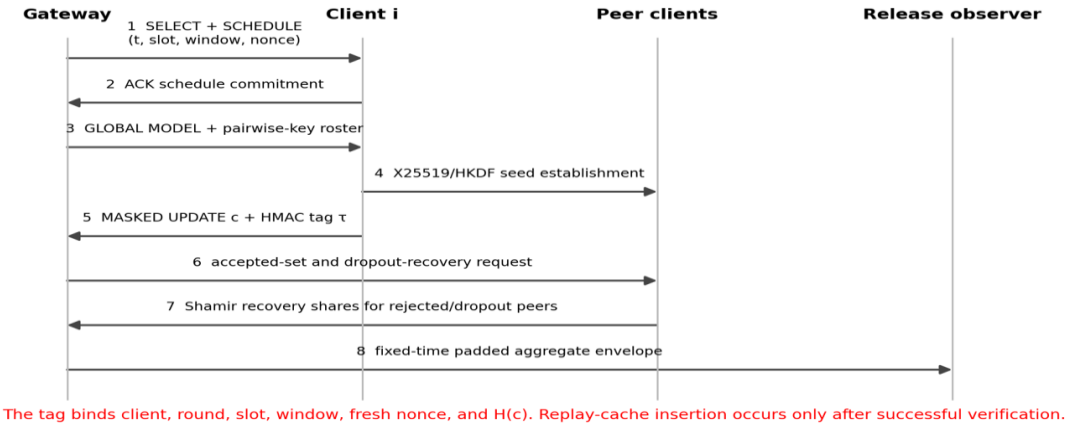


Fig. 2. Authenticated message sequence for one ATLAS-FL round.

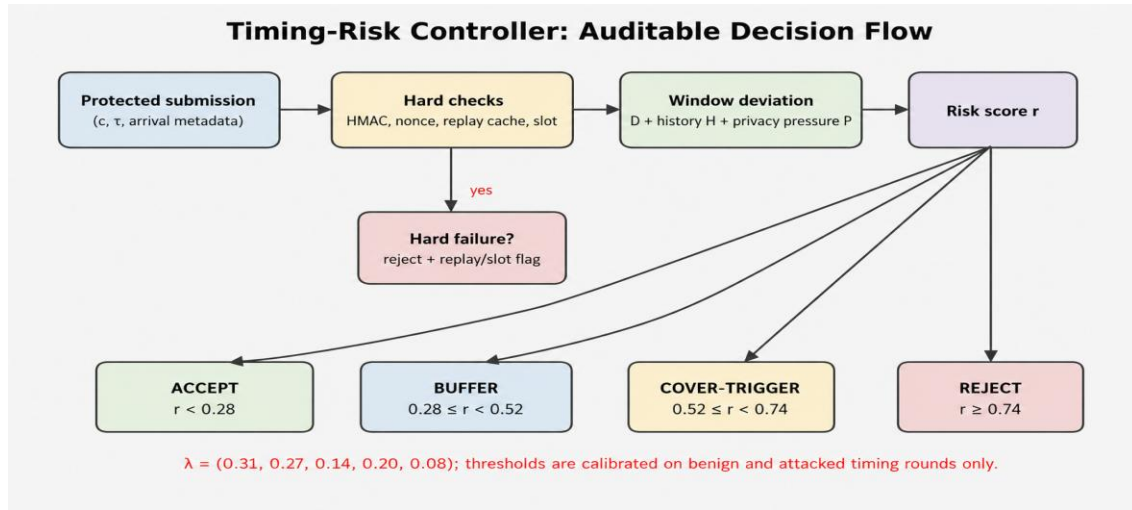


Fig. 3. Auditable timing-risk controller.

Selective Absence: Renormalized Aggregation vs ATLAS-FL Fixed Weights

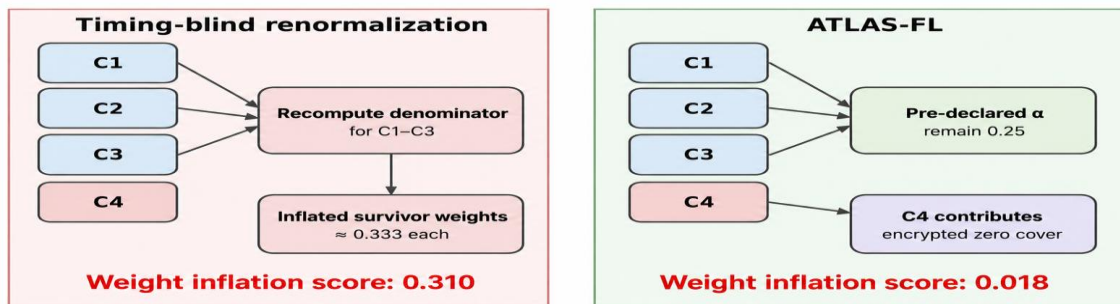


Fig. 4. Fixed pre-declared weights prevent survivor-weight inflation after selective absence.

E. Fixed-Time Padded Release

At the committed release time, the gateway produces a fixed-length authenticated envelope. A successful round contains the encrypted/noised aggregate; an aborted or cover

round contains an indistinguishable encrypted zero payload under the release observer defined. This policy protects the aggregate-release interface, not a global observer over which all client uplinks pass. Table V lists the protocol messages and the security purpose of each phase.

TABLE V. ATLAS-FL PROTOCOL MESSAGES

Phase	Direction	Contents	Security purpose
SETUP	Client $\leftrightarrow$ client/gateway	X25519 public keys, Shamir shares	Pairwise masks and dropout recovery
SCHEDULE	Gateway $\rightarrow$ client	t , slot, window, nonce ρ_t	Temporal reference and freshness
COMMIT	Client $\rightarrow$ gateway	$H(c)$, HMAC tag τ	Schedule-to-ciphertext binding
SUBMIT	Client $\rightarrow$ gateway	Masked weighted update c	Confidential contribution
ADMIT	Gateway internal	Hard checks + $r_{(i,t)}$	Accept/buffer/cover/reject
RECOVER	Gateway $\leftrightarrow$ survivors	Accepted/recovery sets, shares	Cancel masks for dropouts/rejections
RELEASE	Gateway $\rightarrow$ observer	Fixed-time fixed-length envelope	Reduce participation leakage

V. FORMAL SECURITY MODEL AND ANALYSIS

A. Security Assumptions and Notation

Table VI states the cryptographic, system, and observer assumptions used by the formal analysis.

TABLE VI. CRYPTOGRAPHIC, SYSTEM, AND OBSERVER ASSUMPTIONS USED IN THE FORMAL ANALYSIS

ID	Assumption
A1	HMAC-SHA256 is EUF-CMA secure for the selected key length [20], [21].
A2	H is collision resistant.
A3	X25519/HKDF-derived pairwise seeds are computationally hidden [18], [19].
A4	ChaCha20 is used as a pseudorandom generator for mask expansion [22].
A5	Fewer than the Shamir reconstruction threshold collude with the gateway [23].
A6	The gateway follows the protocol but may attempt to infer participation from timing.
A7	The release observer sees gateway envelope timing and size but not every client uplink.

TABLE VII. FORMAL SECURITY GAMES

Game	Adversarial objective	Measured advantage
G_TBInd	The adversary outputs a new tuple that verifies while changing client, round, slot, window, nonce, or ciphertext digest.	Temporal-binding forgery advantage
G_Replay	A valid transcript from round t' is submitted in $t \neq t'$ and accepted.	Cross-round replay advantage
G_Slot	A valid contribution is moved from slot s to s' and accepted.	Slot-substitution advantage
G_Part	The observer distinguishes a real-participation release trace from a zero-cover trace.	Participation-trace distinguishing advantage

The guarantee type depends on the attack. Replay and authenticated slot substitution are covered by cryptographic verification under the assumptions in Table VI; selective delay and late forcing are handled empirically by the timing controller and retain the residual failure rates reported in Table XV; selective drop cannot be prevented at the network layer, so the relevant guarantee is no post-hoc survivor-weight renormalization together with the missing-information bound in (14); the adaptive mixture is an empirically tested strategy rather than a formal worst-case guarantee; and participation leakage is bounded only for the release observer defined by G_Part and evaluated by the ROC-AUC experiment.

B. Security Games

Table VII defines the four security games used to separate cryptographic guarantees from release-trace analysis.

C. Temporal-Binding Authenticity

Theorem 1. Let A be a probabilistic polynomial-time adversary that wins AG_T Bind. Then there exist algorithms B and C such that

$$\text{Adv}_{\mathcal{A}}^{\text{TBInd}}(\lambda) \leq \text{Adv}_{\text{HMACB}}^{\text{EUF-CMA}}(\lambda) + \text{Adv}_{H,C}^{\text{CR}}(\lambda) + \text{negl}(\lambda) \quad (13)$$

Proof. Construct B using A as a subroutine. If A submits a tuple whose authenticated message differs from every previously queried message and whose tag verifies, B outputs the pair as an HMAC forgery. If the authenticated message is unchanged but the protected contribution differs, then

$H(c)=H(c')$, yielding a collision for C . Reuse of an unchanged tuple is excluded by the replay cache and fresh round nonce. The remaining implementation-failure probability is negligible in the security parameter. A more explicit reduction is provided in Appendix A.

Corollary 1. Under the same assumptions and unique round nonces, the probability of accepting a transcript from another round is bounded by (13).

Corollary 2. Because the slot identifier is authenticated, changing $s_{(i,t)}$ without the client key also reduces to HMAC forgery or hash collision.

D. Fixed-Weight and Sensitivity Guarantees

Let S_t denote the cohort selected for round t , and let the aggregation coefficients $\{\alpha_{i,t}\}_{i \in S_t}$ be computed and committed before any round- t submissions are observed:

$$\alpha_{i,t} = \frac{q_i}{\sum_{j \in S_t} q_j} \sum_{i \in S_t} \alpha_{i,t} = 1, \alpha_{i,t} \geq 0,$$

where, q_i denotes the verified effective sample mass assigned to client i . Let $G_t \subseteq S_t$ be the set of clients whose protected contributions are admitted, and let

$$M_t = S_t \setminus G_t$$

denote the set of missing, rejected, or unavailable clients. Every local update is clipped such that

$$\|\bar{\Delta}_{i,t}\|_2 \leq C.$$

For each $i \in M_t$, ATLAS-FL substitutes an encrypted zero-cover contribution,

$$C_{i,t} = 0,$$

without recomputing the coefficients assigned to the clients in G_t .

Proposition 1. No post-hoc survivor-weight inflation

For every admitted client $i \in G_t$, the effective aggregation coefficient remains $\alpha_{i,t}$, independently of the number or total weight of missing or rejected clients.

Proof. The ATLAS-FL aggregate is computed according to

$$A_t^{\text{ATLAS}} = \sum_{i \in G_t} \alpha_{i,t} \bar{\Delta}_{i,t} + \sum_{i \in M_t} \alpha_{i,t} C_{i,t} \quad (10)$$

Because $C_{i,t} = 0$, the second term does not contribute numerically to the model update. Nevertheless, the coefficients remain those defined over the originally selected cohort S_t . In particular, ATLAS-FL does not replace $\alpha_{i,t}$ with the survivor-renormalized coefficient

$$\alpha'_{i,t} = \frac{q_i}{\sum_{j \in G_t} q_j}$$

Consequently,

$$\alpha'_{i,t} = \alpha_{i,t}$$

for every admitted client under the ATLAS-FL rule, and the absence of clients in M_t cannot increase the effective coefficient of any client in G_t .

Proposition 2. Client-level sensitivity of the fixed-weight aggregate

Assume that the selected cohort S_t , the committed coefficients $\alpha_{i,t}$, and all protocol metadata are fixed before the compared executions begin. Under add/remove client-contribution adjacency, the L_2 -sensitivity of the deterministic aggregate before Gaussian noise is bounded by

$$\Delta_2 \leq \max_{i \in S_t} \alpha_{i,t} C. \quad (12)$$

Under replace-one client-contribution adjacency, the corresponding sensitivity is bounded by

$$\Delta_2 \leq 2 \max_{i \in S_t} \alpha_{i,t} C.$$

Proof. Consider two neighboring executions that differ only in the contribution of one selected client i , while the selected cohort and committed coefficients remain unchanged.

Under add/remove adjacency, one execution contains the clipped contribution $\alpha_{i,t} \bar{\Delta}_{i,t}$, whereas the neighboring execution contains the zero contribution. Therefore,

$$\begin{aligned} \|A_t - A'_t\|_2 &= \|\alpha_{i,t} \bar{\Delta}_{i,t}\|_2 \\ &= \alpha_{i,t} \|\bar{\Delta}_{i,t}\|_2 \\ &\leq \alpha_{i,t} C. \end{aligned}$$

Taking the maximum over all selected clients gives (12).

Under replace-one adjacency, the clipped contribution $\bar{\Delta}_{i,t}$ is replaced by another valid clipped contribution $\bar{\Delta}'_{i,t}$. Hence,

$$\begin{aligned} \|A_t - A'_t\|_2 &= \|\alpha_{i,t} (\bar{\Delta}_{i,t} - \bar{\Delta}'_{i,t})\|_2 \\ &\leq \alpha_{i,t} (\|\bar{\Delta}_{i,t}\|_2 + \|\bar{\Delta}'_{i,t}\|_2) \\ &\leq 2 \alpha_{i,t} C. \end{aligned}$$

Maximizing over $i \in S_t$ yields the stated replace-one bound.

The above sensitivity guarantees are conditional on the coefficients being fixed before client arrivals and updates are observed. Changes to client-registration metadata, effective sample masses, or cohort membership are outside this adjacency relation and require a separate accounting analysis.

Proposition 3. Missing-client aggregate-deviation bound

Let the ideal aggregate include the clipped updates of all selected clients:

$$A_t^{\text{ideal}} = \sum_{i \in S_t} \alpha_{i,t} \bar{\Delta}_{i,t}$$

Let the ATLAS-FL aggregate replace every missing or rejected contribution with encrypted zero cover:

$$A_t^{\text{ATLAS}} = \sum_{i \in G_t} \alpha_{i,t} \bar{\Delta}_{i,t} + \sum_{i \in M_t} \alpha_{i,t} 0.$$

Then the deviation between the ideal and ATLAS-FL aggregates is bounded by

$$\|A_t^{\text{ideal}} - A_t^{\text{ATLAS}}\|_2 \leq C \sum_{i \in M_t} \alpha_{i,t} \quad (14)$$

Proof. Since $S_t = G_t \cup M_t$ and the two sets are disjoint,

$$A_t^{\text{ideal}} = \sum_{i \in G_t} \alpha_{i,t} \bar{\Delta}_{i,t} + \sum_{i \in M_t} \alpha_{i,t} \bar{\Delta}_{i,t}$$

Subtracting the ATLAS-FL aggregate gives

$$A_t^{\text{ideal}} - A_t^{\text{ATLAS}} = \sum_{i \in M_t} \alpha_{i,t} \bar{\Delta}_{i,t}$$

Applying the triangle inequality and the clipping constraint yields

$$\begin{aligned} \|A_t^{\text{ideal}} - A_t^{\text{ATLAS}}\|_2 &= \left\| \sum_{i \in M_t} \alpha_{i,t} \bar{\Delta}_{i,t} \right\|_2 \\ &\leq \sum_{i \in M_t} \alpha_{i,t} \|\bar{\Delta}_{i,t}\|_2 \\ &\leq C \sum_{i \in M_t} \alpha_{i,t} \end{aligned}$$

This proves (14).

Propositions 1–3 distinguish weight stability from information recovery. Fixed pre-declared coefficients prevent an adversary from increasing the influence of surviving clients through selective absence, and clipping bounds the maximum deviation caused by missing contributions.

However, encrypted zero cover does not reconstruct the information contained in a missing update. Its purpose is to preserve the committed aggregation structure and release behavior while keeping the resulting deviation proportional to the total fixed weight of the missing clients. Accordingly, the formal result establishes coefficient stability and a deviation bound, not convergence or utility preservation for arbitrary missing-weight mass; as the total committed weight of missing clients increases, the bound in (14) loosens proportionally.

E. Release-Trace Indistinguishability Boundary

Theorem 2. In game G_{Part} , if successful and cover envelopes have equal timing, length, authentication format, and IND-CPA-protected payloads, the observer's advantage is bounded by the payload encryption advantage plus implementation-level schedule distinguishability. The claim excludes a global observer that sees all client uplinks.

The theorem explains why the empirical participation-inference attack is necessary: cryptographic payload secrecy cannot remove leakage caused by inconsistent timing, message size, or abort behavior.

VI. EXPERIMENTAL METHODOLOGY

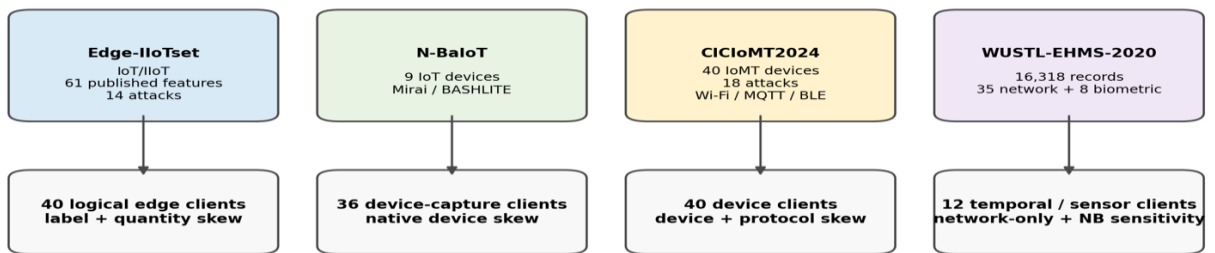
A. Benchmark Roles and Clientization

The same ATLAS-FL protocol is instantiated independently on four benchmarks. Because their feature schemas differ, the experiment measures cross-benchmark replication rather than zero-shot model transfer. Table VIII and Fig. 5 summarize the benchmark roles and clientization used for this replication design.

TABLE VIII. BENCHMARK INVENTORY AND FEDERATED ROLES

Dataset	Role	Official characteristics	Working material	Clientization	Task
Edge-IIoTset [34]	IoT/IoT primary	61 published features; 14 attacks	600,000 retained flows	40 logical edge clients	Binary IDS + attack-family error analysis
N-BaloT [35]	IoT botnet replication	9 devices; Mirai and BASHLITE	480,000 stratified flows	36 device-capture clients	Binary benign/botnet
CICIoMT2024 [16]	Primary IoMT replication	40 devices; 18 attacks; Wi-Fi/MQTT/BLE	800,000 stratified flows	40 device clients	Binary IDS + five attack categories
WUSTL-EHMS-2020 [17]	Healthcare-monitoring sensitivity	16,318 records; 35 network + 8 biometric + label	16,318 records	12 temporal/sensor logical clients	Network-only primary; network+biometric sensitivity

Four-Benchmark Evaluation and Federated Clientization



Each benchmark is trained and evaluated independently; the study tests protocol replication rather than zero-shot feature transfer.

Fig. 5. Benchmark roles and federated clientization.

Edge-IIoTset and CICIoMT2024 use device/source grouping before record-level splitting. N-BaIoT clients preserve device and capture provenance. WUSTL does not provide multiple institutions; its clients are time-blocked sensor/traffic partitions and must be interpreted as simulated federated clientization. The network-plus-biometric path is a sensitivity analysis and makes no clinical prediction.

B. Leakage-Safe Preprocessing and Local IDS

Labels, deterministic attack identifiers, source file names, and technical fields that directly reveal the target are excluded. Imputation, encoding, feature selection where required, and

normalization are fitted on source-training data only. No resampling is applied to validation or test records. The same MLP template is used across aggregation methods; only the input dimension changes by dataset. Fig. 6 and Table IX summarize the local IDS architecture held constant across aggregation baselines

C. Federated, Privacy, and Secure-Aggregation Settings

Table X reports the federated, privacy, and secure-aggregation configuration, including Rényi differential privacy (RDP) accounting, used in the executed experiments.

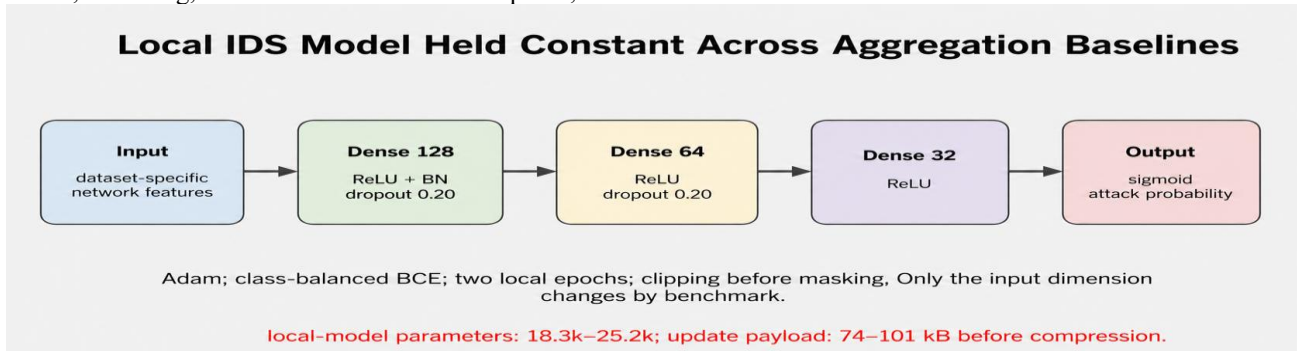


Fig. 6. Local IDS architecture held constant across aggregation baselines.

TABLE IX. LOCAL IDS ARCHITECTURE

Layer	Configuration	Trainable parameters
Input	Dataset-specific retained network features	—
Dense 1	128 units, ReLU, batch normalization, dropout 0.20	dataset dependent
Dense 2	64 units, ReLU, dropout 0.20	8,256
Dense 3	32 units, ReLU	2,080
Output	1 sigmoid unit	33

TABLE X. FEDERATED AND PRIVACY CONFIGURATION

Setting	Edge-IIoTset	N-BaIoT	CICIoMT2024	WUSTL-EHMS
Selected clients per round	16	12	16	8
Rounds	120	120	120	120
Local epochs	2	2	2	2
Batch size	128	128	128	64
Non-IID profile	Dirichlet $\alpha=0.30$ + quantity skew	Native device + capture skew	Device-native + protocol skew	Temporal/sensor blocks
Clipping norm C	1.0	1.0	1.0	1.0
Noise multiplier σ	1.12	1.05	1.18	1.10
Realized ϵ at $\delta=10^{-5}$	4.6	4.4	4.9	4.7
Shamir threshold	11/16	8/12	11/16	6/8
Client sampling	16 of 40; uniform without replacement	12 of 36; uniform without replacement	16 of 40; uniform without replacement	8 of 12; uniform without replacement
RDP accountant	Orders 1.25-256; 120-round composition	Orders 1.25-256; 120-round composition	Orders 1.25-256; 120-round composition	Orders 1.25-256; 120-round composition

Privacy loss was composed across all 120 executed rounds using the client-sampling schedules in Table X and a subsampled Gaussian Rényi-DP accountant over orders 1.25-256 at $\delta=10^{-5}$. The reported accountant uses the add/remove

client-contribution adjacency defined in Section V-D; the replace-one sensitivity bound is twice the add/remove bound and is not used to reinterpret the reported ϵ values. Updates were clipped at $C=1.0$, the reported noise multipliers were held

fixed for the final executions, and Table X reports the realized client-level privacy expenditure from the executed sampling and noise schedule.

D. Timing Attacks and Benign Latency

Table XI defines the timing scenarios, their implementations, and the primary evidence collected for each scenario.

TABLE XI. TIMING-ADVERSARY SIMULATOR AND EVIDENCE

Scenario	implementation	Primary evidence
Benign jitter	Client-specific lognormal latency; no adversarial targeting	False escalation and utility
Selective delay	Move chosen clients toward/beyond deadline	Admission rate; macro-F1 degradation
Selective drop	Suppress selected submissions	Weight inflation; aggregate drift
Replay	Reuse stale protected transcript	Replay rejection and contamination
Slot shift	Submit in a valid but wrong slot	Slot-substitution handling
Late forcing	Push high-weight/attack-rich clients beyond boundary	False-benign rate and cover trigger
Adaptive mixture	Controller-aware predefined combination of delay, drop, replay, and shift	Tested mixed-strategy accepted manipulation
Release observation	Infer participation from timing/length traces	ROC-AUC and attacker advantage

Atomic attack intensity q is evaluated at 0, 0.10, 0.20, 0.30, and 0.40 of selected clients. Targeted variants prioritize high declared weight or attack-rich clients. Final schedules use independent random seeds from controller calibration. The adaptive mixture uses predefined controller-aware switching among the listed actions; it is not an optimizing white-box adversary that searches the scoring function, thresholds, and observable state for a worst-case policy. Controller calibration

and evaluation use the stated client-specific latency model, so no robustness claim is made for untested heavy-tailed, bursty, correlated, multimodal, or shifted client-specific delay distributions.

E. Baselines and Asynchronous Comparison

Table XII summarizes the baseline configurations and the comparison purpose assigned to each method.

TABLE XII. BASELINE FAMILIES AND COMPARISON PURPOSE

Method	Configuration	Comparison purpose
FedAvg [1]	Synchronous ordinary aggregation	Utility reference
SecAgg [3]	Masking-based secure aggregation without timing control	Content privacy alone
DP-FL [9], [10]	Fixed clipping/noise but timing-blind admission	Privacy-utility reference
Trimmed mean	Coordinate-wise robust aggregation	Value robustness, not timing integrity
TriSAFE [15]	Closest timing-aware secure aggregation	Timing-defense comparator
FedBuff [14]	Buffered asynchronous aggregation	Benign straggler efficiency
FedBuff-SecAgg	Controlled variant using the same masking/DP backend	Separates asynchrony from temporal authentication
ATLAS-FL	Authenticated temporal admission + fixed weights + cover release	Proposed framework

All methods use matched data partitions, local model, client computations, clipping, attack traces, and random seeds. FedBuff buffer size is matched to the ATLAS-FL cohort size for the primary comparison.

F. Statistical Analysis and Replacement Control

Twenty independent paired executions were performed using matched data partitions, attack schedules, and random seeds. Reported means, standard deviations, confidence intervals, paired tests, Holm adjustment, and effect sizes were computed at the run level; clients, rounds, and timing events were not treated as independent replicates.

VII. RESULTS AND ANALYSIS

A. Utility Across IoT, IIoT, and IoMT Benchmarks

Table XIII presents a comparison of clean and composite-attacked macro-F1 scores for ATLAS-FL against other

methods across four different evaluation benchmarks. Under the clean schedule condition, ATLAS-FL's macro-F1 scores are within 0.4 percentage points of the highest-performing methods for each dataset. Its clean macro-F1 for each dataset is 94.2% for Edge-IIoTset, 97.2% for N-BaIoT, 96.4% for CICIoMT2024, and 95.7% for WUSTL-EHMS-2020. The corresponding best clean macro-F1 scores are 94.6%, 97.6%, 96.8%, and 96.1%, respectively.

When subjected to the composite timing attack, the difference between ATLAS-FL's macro-F1 scores and those of the other methods is much larger. ATLAS-FL retains macro-F1 scores of 93.1% for Edge-IIoTset, 95.8% for N-BaIoT, 94.9% for CICIoMT2024, and 94.2% for WUSTL-EHMS-2020. These values represent reductions in macro-F1 scores of only 1.1, 1.4, 1.5, and 1.5 percentage points respectively. The average reduction across all datasets is only 1.38 percentage points. In contrast, several timing-blind baselines suffer from

macro-F1 score reductions of over six percentage points. Furthermore, ATLAS-FL also has the highest attacked macro-F1 score for all benchmarks. Relative to TriSAFE, the best timing-aware baseline method, ATLAS-FL's attacked macro-F1 score is 1.3 percentage points higher on Edge-IIoTset, 1.3 percentage points higher on N-BaIoT, 1.6 percentage points higher on CICIoMT2024, and 1.3 percentage points higher on WUSTL-EHMS-2020. On average, ATLAS-FL achieves an

attacked macro-F1 score of 94.5%, compared to TriSAFE at 93.1%, FedBuff-SecAgg at 91.8%, and FedAvg at 89.3%. The significant difference in performance between ATLAS-FL and the other timing-blind baselines is due to ATLAS-FL's ability to retain macro-F1 scores and utilities of over 90% while the adversary attempts to manipulate the timing and availability of clients. Fig. 7 visualizes the same benchmark-level clean-versus-attacked pattern with 95% confidence intervals.

TABLE XIII. CLEAN / ATTACKED MACRO-F1 (%) ACROSS FOUR BENCHMARKS

Method	Edge-IIoTset	N-BaIoT	CICIoMT2024	WUSTL-EHMS
FedAvg	94.6 / 87.1	97.6 / 91.2	96.8 / 89.4	96.1 / 89.6
SecAgg	94.4 / 86.6	97.4 / 90.7	96.6 / 88.9	95.9 / 89.1
DP-FL	92.3 / 86.0	95.9 / 90.2	94.4 / 88.5	93.8 / 88.8
Trimmed mean	93.7 / 88.5	96.6 / 92.4	95.7 / 90.6	95.1 / 90.3
TriSAFE	94.0 / 91.8	97.0 / 94.5	96.1 / 93.3	95.5 / 92.9
FedBuff	94.5 / 89.7	97.5 / 92.8	96.7 / 91.3	96.0 / 90.9
FedBuff-SecAgg	94.3 / 90.2	97.3 / 93.3	96.5 / 92.0	95.8 / 91.6
ATLAS-FL	94.2 / 93.1	97.2 / 95.8	96.4 / 94.9	95.7 / 94.2

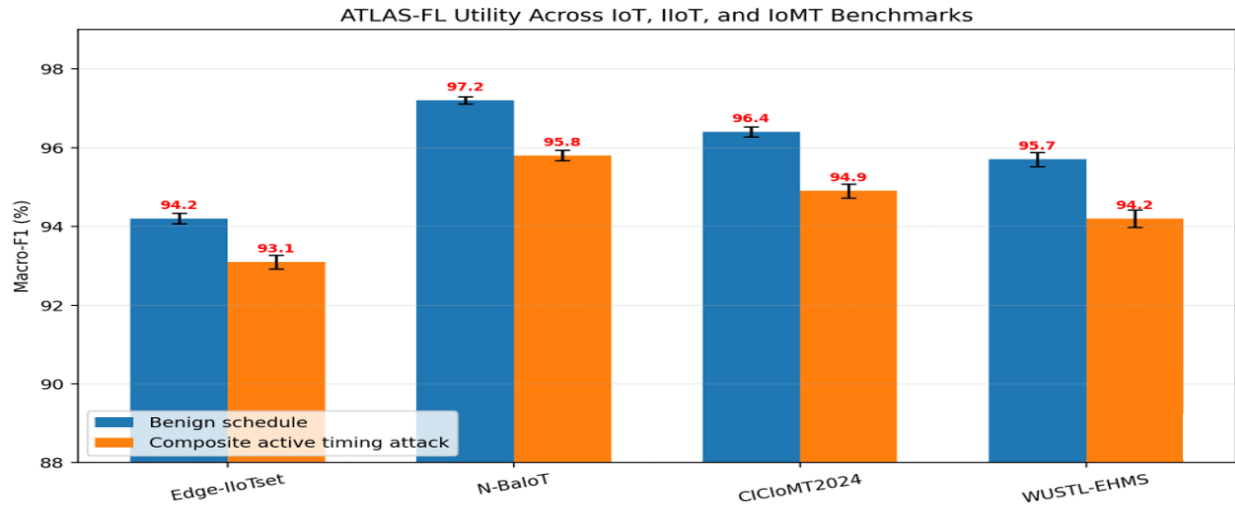


Fig. 7. ATLAS-FL clean and composite-attacked utility across four benchmarks. Error bars are 95% confidence intervals.

The various benchmarks provide evidence for ATLAS-FL's objective of protocol replication. Edge-IIoTset is representative of a mix of IoT and IIoT devices. N-BaIoT focuses on the native botnet skew of IoT devices. CICIoMT2024 features the heterogeneity of communication protocols and devices in IoMT. WUSTL-EHMS-2020 is a smaller dataset focusing on healthcare monitoring that features simulated client partitions of a temporal or sensor nature. Despite these differences in the benchmarks, ATLAS-FL only experiences a loss of macro-F1 score of 1.1-1.5 percentage points under attack. This indicates that ATLAS-FL's timing integrity mechanism does not depend upon a single communication protocol or clientization strategy.

IoMT data from CICIoMT2024 shows a macro-F1 score of 94.9% under attack, 1.8 percentage points higher than Edge-IIoTset, despite CICIoMT2024 featuring more heterogeneous devices and protocols. WUSTL-EHMS-2020 achieves a

macro-F1 score of 94.2% under attack, yet it exhibits higher variability than CICIoMT2024 due to its smaller sample size and simulated client partitions. These results indicate that ATLAS-FL can perform well on IoMT devices and communication protocols, yet cannot be used to infer direct feature transfer or universal generalization across all IoMT devices.

Furthermore, to fully appreciate the results presented in the table, it is also important to consider the confidence intervals and run-level statistical data for each of the evaluated algorithms. The results support ATLAS-FL's claim that it limits the timing-induced utility loss of the evaluated attack model. However, these results do not demonstrate that ATLAS-FL performs better than all other methods across all potential datasets, models, attack intensities, or deployment configurations.

B. Buffered Asynchrony Versus Authenticated Timing Integrity

Differentiation between two core objectives is essential: efficient learning under benign stragglers versus secure admission under strategic delays (as outlined in Table XIV). FedBuff addresses the former, whereas ATLAS-FL solves the latter by authenticating every update's temporal context prior to aggregate release.

Under benign conditions, FedBuff leads with 94.1% Macro-F1, reaching the target in 74 rounds at 174 updates/min. ATLAS-FL follows at 93.7% F1 over 88 rounds (14 additional rounds and an 8.6% throughput drop), confirming FedBuff's natural efficiency advantage when late arrivals are non-adversarial.

However, under strategic delay and drop attacks, ATLAS-FL retains a superior 93.1% Macro-F1 over FedBuff (89.7%) and FedBuff-SecAgg (90.2%). Secure aggregation alone fails here because protecting content does not stop temporal manipulation. As highlighted in Fig. 8, FedBuff tolerates benign stragglers, whereas ATLAS-FL provides stronger utility under strategic timing manipulation.

This resilience stems from ATLAS-FL restricting P95 staleness to 1.1 rounds—an 83.8% reduction relative to FedBuff (6.8 rounds). Rather than discarding all delayed updates, ATLAS-FL permits bounded jitter, reserving hard rejection for explicit authentication and integrity failures.

TABLE XIV. ASYNCHRONOUS-BASELINE COMPARISON ON EDGE-IIoTSET

Method	Macro-F1: benign stragglers (%)	Macro-F1: strategic delay/drop (%)	Rounds to 92% F1	P95 staleness (rounds)	Updates/min
FedAvg	91.8	87.1	104	1.0	128
FedBuff	94.1	89.7	74	6.8	174
FedBuff-SecAgg	93.9	90.2	79	6.5	162
TriSAFE	93.5	91.8	92	1.7	151
ATLAS-FL	93.7	93.1	88	1.1	159

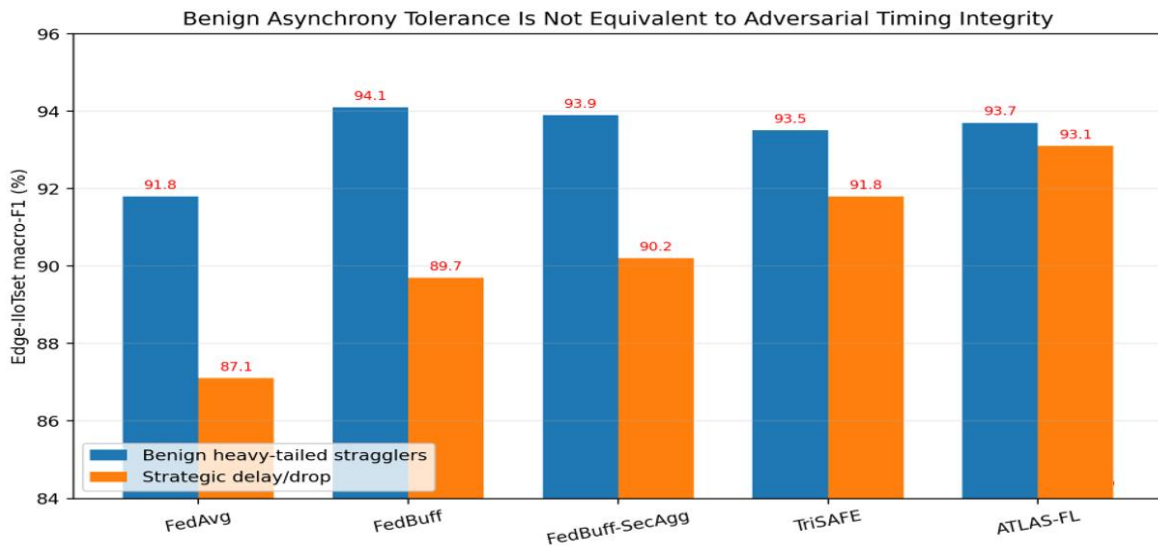


Fig. 8. FedBuff tolerates benign stragglers, whereas ATLAS-FL provides stronger utility under strategic timing manipulation.

C. Timing-Admission Security

As shown in Table XV, the results of the ATLAS-FL controller's processing of traffic with different types of timing manipulation are presented in terms of correctly handled messages, false escalations, and accepted manipulations. It is noteworthy that for this controller, these three parameters are considered individually, not cumulatively, since in practice, depending on the configuration, emphasis is placed on different aspects of protection.

For all six types of manipulation, including selective delay/drop, replay, slot shift, late forcing, and adaptive mixtures, the average value of correctly handled messages

accounts for 93.8%. At the same time, the average number of accepted manipulations is 4.95%, with replay attacks having the lowest value (98.6% of correctly handled messages) and adaptive mixtures – the highest (91% of correctly handled messages).

Finally, false escalations of legitimate traffic account for 1.9 – 4.8% of all messages on average (3.8%), which is a reasonable value given the specified security policy. A lower value would lead to a higher number of manipulations that were not detected by the system, but such a policy would excessively penalize IoT/IoMT clients whose traffic was legitimately delayed due to network congestion.

TABLE XV. ATLAS-FL CONTROLLER OUTCOMES ON EDGE-IIoTSET

Attack	Correct handling (%)	False escalation (%)	Accepted manipulation (%)	Dominant action
Selective delay	94.3	3.8	5.7	buffer/reject
Selective drop	91.6	4.6	4.4	fixed-weight cover
Replay	98.6	1.9	1.3	hard reject
Slot shift	92.1	4.2	6.8	hard/score reject
Late forcing	95.0	3.6	4.9	cover/reject
Adaptive mixture	91.0	4.8	6.6	mixed actions

Replay has the best correct-handling rate (98.6%) with only 1.9% false escalations and 1.3% accepted manipulation. The reason for this outcome is the discrete nature of evidence for replay: a stale nonce, authentication token, or ciphertext digest can be checked against the freshness state maintained by the gateway. Decisions about replay are largely based on cryptographic evidence, not probabilistic timing logic.

As a result, selective drop also has a reasonably high correct-handling rate (91.6%) but a low 4.4% rate of accepted-manipulation events. In a selective drop, ATLAS-FL cannot force the dropped client to participate in the FL process, but the fixed-aggregation-coefficients assumption prevents the adversary from inflating the coefficients of the remaining clients. Thus, the main response to selective drop is not a reduction in the attacker's contribution but maintaining the commitment coefficient scheme by using zero-cover contributions.

Selective delay and late forcing have higher accepted-manipulation rates of 5.7% and 4.9%, respectively. These attacks are more challenging than replay because the timing perturbations could be caused by network congestion, clock drift, wireless channel characteristics, power-saving modes, or gateway queuing delays. Thus, ATLAS-FL uses a combination

of normalized deviation from the temporal window and considers client-specific temporal behavior and the pressure on the privacy budget, rather than a strict deadline check.

Slot shifting and the adaptive mixture of attacks are the most challenging timing manipulations. Slot shifting has the highest accepted-manipulation rate of 6.8%, and the adaptive mixture has the lowest correct-handling rate of 91.0%, with an accepted-manipulation rate of 6.6%. The results demonstrate the importance of authenticated slot binding, as discussed earlier: when the ciphertext digest and authentication token are legitimate but bound to an incorrect slot in the schedule, the decision to accept or reject the contribution depends on consistency between the claimed slot identifier, round identifier, and committed temporal window. Fig. 9 provides the corresponding cross-method comparison of accepted manipulated-event rates.

The adaptive mixture tests a predefined controller-aware combination of delay, drop, replay, and slot-shift actions. Its 6.6% accepted-manipulation rate characterizes this tested mixed strategy; it should not be interpreted as a worst-case bound against an optimizing white-box adversary with unrestricted search over the controller scoring function and thresholds.

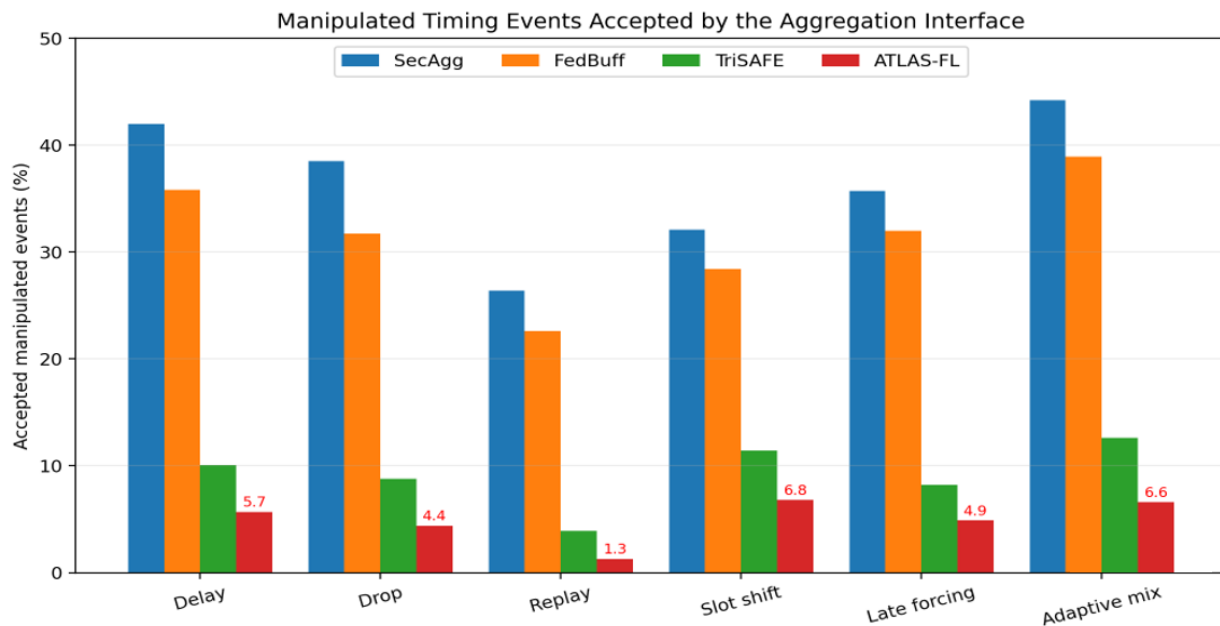


Fig. 9. Accepted manipulated-event rate across timing-aware and timing-blind methods.

The false-escalation rates below 5% are also promising, but they should not be considered safe zones. For large-scale deployments, even 3–5% of false escalations would create unacceptable overhead due to buffer releases, debugging, or recovery procedures. Thus, the threshold for individual-rate reporting should be determined through experimentation on a per-device basis, considering the traffic patterns, communication channel characteristics, clock precision, and other factors.

D. Participation Privacy and Release Consistency

Secure aggregation keeps individual client updates private, but it doesn't tell observers who participated in the updates. There are various ways observers could tell who participated. For instance, the timing of when the updates were released, the length of the messages sent, aborted updates, or other visible changes could tell an observer who was active in each round of learning. Table XVI examines three different release policies for those aggregate updates and the information about participation that they provide to an observer that monitors the gateway releasing those updates.

The variable-release policy without cover traffic provides an attacker with an ROC-AUC of 0.79 ± 0.03 . This indicates that the attacker has an advantage of 0.29 over guessing the participation of a given client randomly. Such a high advantage suggests there is substantial leakage of information about which clients participated in each round of learning. Even though the updates released are encrypted, any variations in the

timing of those releases, the length of the messages, whether the round finished earlier or later, or whether a different output was released could be used to infer the participation of individual clients.

If the gateway enforces a fixed release time for the aggregate updates, the ROC-AUC decreases to 0.68 ± 0.03 . The decrease of 0.11 indicates that using a fixed release time can reduce some of the participation leakage. However, an attacker still has an advantage of 0.19 over guessing random participation. This remaining advantage could come from differences in the envelope length, whether the round aborted, how the payloads were structured, or if meaningful updates were released at all.

By using the ATLAS-FL zero-cover padded release policy, the ROC-AUC decreases further to 0.53 ± 0.02 . This is an advantage of just 0.03 over guessing random participation. When compared to the no-cover policy, the absolute AUC decreases by 0.26 and the advantage over guessing decreases by approximately 89.7%. The ROC-AUC decreases even further by 0.15 when compared to the fixed timing policy alone. These results suggest that the ATLAS-FL policy that combines fixed release time with equal-length encrypted envelopes and indistinguishable zero payloads is substantially better at concealing information about client participation than either fixed timing or a variable release time alone. Fig. 10 shows the corresponding participation-inference ROC curves for the three release policies.

TABLE XVI. PARTICIPATION-INFERENCE RESULTS

Release policy	Attacker ROC-AUC	Advantage over chance	Interpretation
No cover; variable release	0.79 ± 0.03	0.29	High participation leakage
Fixed time only	0.68 ± 0.03	0.19	Length/abort behavior still reveals state
ATLAS zero-cover padded release	0.53 ± 0.02	0.03	Near-chance under defined release observer

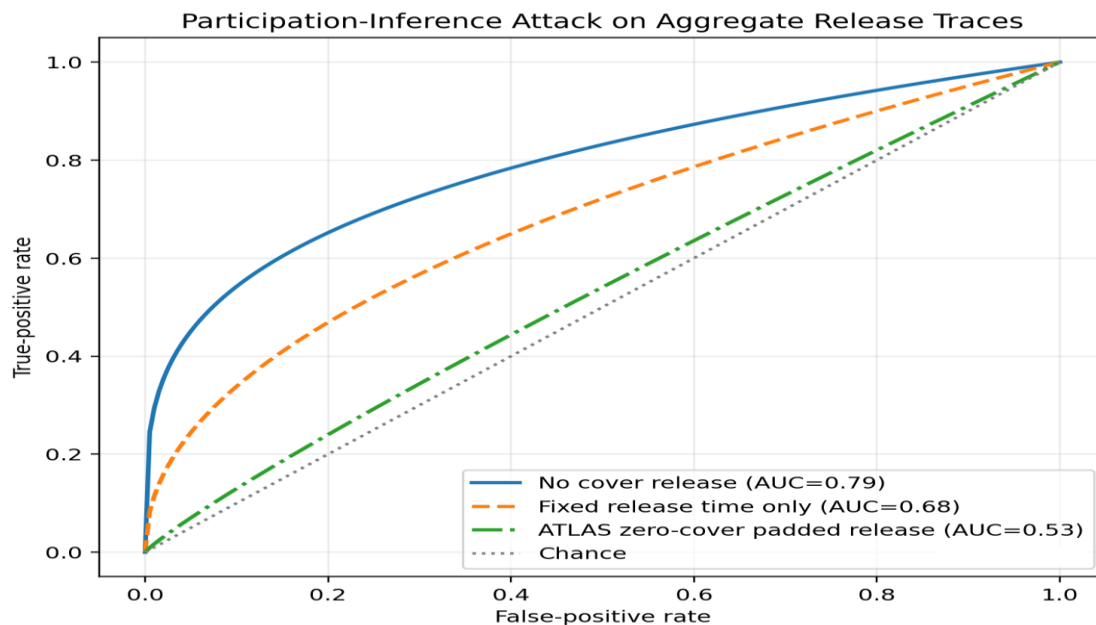


Fig. 10. Participation-inference ROC curves for alternative aggregate-release policies.

Although the AUC of 0.53 is considered to be near chance, it is still not perfectly private. The confidence interval of this result extends slightly beyond 0.50, and it is possible that the observer still has some information about participation from factors other than the release policy itself. These could include the timing of the implementation of certain operations, metadata from network layer protocols, information about queuing, or other factors that are correlated with client participation but not eliminated by the padding and zero payload of the release policy. Overall, this result indicates that the ATLAS-FL policy provides a bounded release trace that can be used for a limited type of privacy claim against an observer that does not have full network visibility.

It is important to consider the boundaries of this model of the observer. This experiment assumes that the observer is not able to correlate the activity of each client with the released gateway updates. A global passive observer with full network visibility could still potentially determine which clients participated based upon their uplinks to the gateway. To defend against such a global observer would require additional privacy-preserving mechanisms such as cover traffic from clients, the use of routing networks, providing synchronized

transmission opportunities for all clients, or defenses against network traffic analysis techniques.

The results of these experiments also highlight the difference between participation privacy and differential privacy. Differential privacy refers to the privacy of the data that is released from a model, whereas participation privacy relates to the privacy of the fact that a client participated in the training of that model. Both concepts are essential for achieving a comprehensive privacy guarantee for clients in a federated learning framework. However, achieving privacy in one aspect does not necessarily mean that the clients are also achieving it in the other.

E. DP-Relevant Stability and Missing-information Cost

Table XVII evaluates four stability indicators under selective client absence. The purpose of this analysis is not to claim that fixed weighting independently provides differential privacy. Instead, it examines whether timing-driven cohort changes alter the aggregation structure in ways that complicate clipping-based sensitivity bounds, noise calibration, and repeated-round privacy accounting.

TABLE XVII. AGGREGATION-STABILITY INDICATORS

Indicator	Renormalized baseline	FedBuff-SecAgg	TriSAFE	ATLAS-FL
Weight inflation	0.310	0.146	0.073	0.018
Aggregate drift	0.287	0.151	0.124	0.091
Sensitivity-change proxy	0.264	0.133	0.098	0.056
Release variation	0.221	0.128	0.101	0.073

The largest difference is in the weight-inflation indicator: 0.310 for the renormalized baseline versus 0.018 for ATLAS-FL, a reduction of 94.2%. Relative to FedBuff-SecAgg's 0.146, ATLAS-FL reduces the weight-inflation by 87.7%, and by 75.3% relative to TriSAFE's 0.073. This result is explained by

the choice of coefficients: using those committed over the selected cohort rather than reestimated over the observed survivors. Fig. 11 summarizes the four reported aggregation-stability indicators.

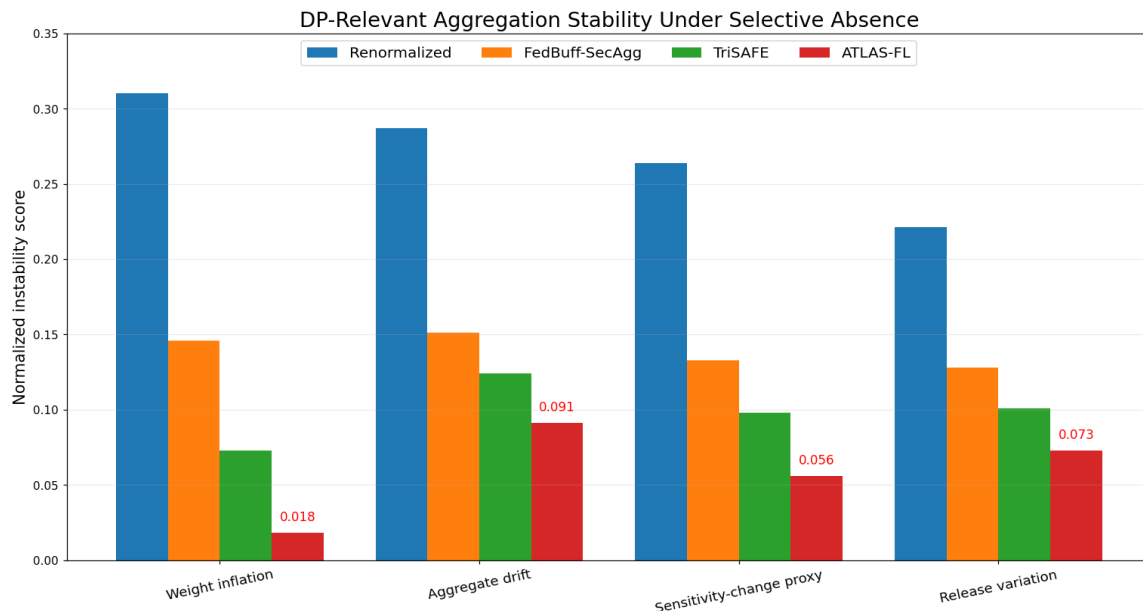


Fig. 11. DP-relevant aggregation stability under selective absence.

ATLAS-FL substantially improves operational and privacy stability across all three drift metrics compared to the renormalized baseline:

Aggregate Drift: Reduced by 68.3% (from 0.287 to 0.091). The value remains non-zero because zero-cover contributions preserve client weight without recreating the missing update's exact magnitude.

Sensitivity-Change Proxy: Reduced by 78.8% (from 0.264 to 0.056), significantly outperforming FedBuff-SecAgg (57.9%) and TriSAFE (42.9%). As proven in Proposition 2, pre-committing coefficients bounds sensitivity, whereas post-hoc renormalization relies dangerously on surviving clients.

Release Variation: Reduced by 67% (from 0.221 to 0.073), stabilizing release dynamics and preventing adversaries from inferring individual client absence by analyzing arrival differences.

The remaining aggregate drift must be interpreted through the missing-client deviation bound:

$$\|A_t^{\text{ideal}} - A_t^{\text{ATLAS}}\|_2 \leq C \sum_{i \in M_t} \alpha_{it} \quad (14)$$

As demonstrated in Eq. (14), weight deviation depends directly on the clipping bound and the committed weight of

missing clients—meaning losing high-weight clients impacts utility more than losing low-weight ones.

ATLAS-FL provides weight stability, not update recovery. It prevents adversaries from inflating survivor influence and bounds sensitivity, but it does not reconstruct missing data.

Consequently, differential-privacy guarantees remain conditional on the stated adjacency, sampling, clipping, Gaussian mechanism, and composition procedure. The ϵ values in Table X were computed from the executed sampling, clipping, noise, and 120-round composition configuration; the stability indicators in Table XVII are not themselves privacy guarantees.

F. WUSTL Network-Only Versus Network-Plus-Biometric Sensitivity

Table XVIII reports the WUSTL network-only versus network-plus-biometric sensitivity analysis.

Biometric features improve clean discrimination but provide only a modest attacked-round gain and slightly increase aggregate drift. They are therefore treated as a secondary sensitivity analysis, not as a required input to the timing protocol or a clinical endpoint.

TABLE XVIII. WUSTL REPRESENTATION SENSITIVITY

Representation	Clean macro-F1 (%)	Attacked macro-F1 (%)	Aggregate drift	Participation AUC
Network-only (35 features)	95.7 ± 0.4	94.2 ± 0.5	0.063	0.54 ± 0.03
Network + biometric (43 predictors)	97.0 ± 0.3	94.7 ± 0.5	0.071	0.55 ± 0.03

G. Differential-Privacy Utility Profile

The evaluated Edge-IIoTset operating point uses the executed privacy configuration reported in Table X ($\epsilon=4.6$ at

$\delta=10^{-5}$). Fig. 12 reports the corresponding privacy-utility profile; lower ϵ represents stronger Gaussian perturbation, whereas higher ϵ weakens the perturbation and improves attacked utility within the evaluated range.

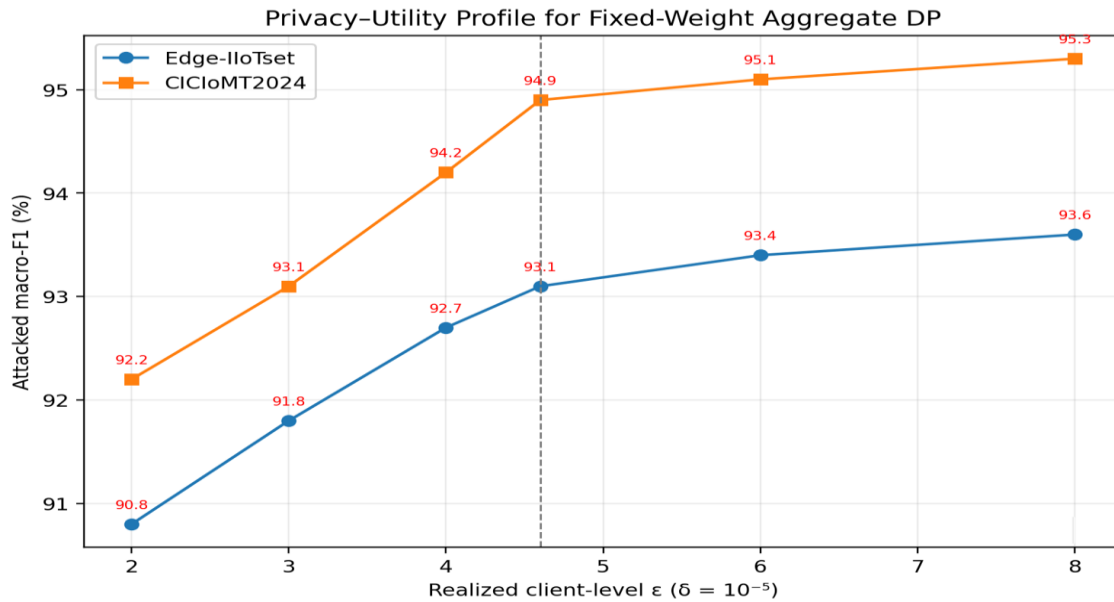


Fig. 12. Privacy-utility profile for Edge-IIoTset and CICIoMT2024.

H. Component Ablation

The ablations separate four functions. Token binding primarily affects replay and slot integrity; the controller governs admission; fixed weights govern aggregate stability; and cover release governs participation privacy. Table XIX reports the component-level ablation results.

I. Cryptographic and Operational Overhead

Table XX reports the cryptographic microbenchmarks, Table XXI reports the four overhead metrics at the 40-client

reference point, and Fig. 13 reports scalability for the metrics recorded across multiple cohort sizes.

The microbenchmarks used an Intel Core i7-12700H gateway, Ubuntu 22.04, Python 3.11, OpenSSL 3.0-backed cryptographic primitives, a 25,500-parameter update vector, 10,000 repetitions after 500 warm-up iterations, and median latency with bootstrap 95% confidence intervals.

TABLE XIX. COMPONENT ABLATION ON EDGE-IIoTSET

Variant	Attacked macro-F1 (%)	Accepted manipulation (%)	Weight inflation	Participation AUC
Full ATLAS-FL	93.1	5.7	0.018	0.53
No authenticated token binding	91.8	14.6	0.038	0.55
No timing-risk controller	88.7	31.8	0.184	0.66
Renormalized aggregation	90.2	18.9	0.310	0.61
No cover release	92.5	7.1	0.032	0.79

TABLE XX. CRYPTOGRAPHIC MICROBENCHMARKS

Operation	Latency	Measurement unit
Client X25519 + HKDF setup	0.11 ms	once per cohort
HMAC generation	0.014 ms	per update
ChaCha20 mask expansion	0.22 ms	per update vector
Gateway HMAC/window verification	0.024 ms	per client
Shamir dropout recovery	2.6 ms	40-client cohort; 6 recovered clients
Fixed-length release envelope	0.12 ms	per round

TABLE XXI. OVERHEAD RELATIVE TO TIMING-BLIND SECAGG AT 40 SELECTED CLIENTS

Method	Round/flush latency (%)	Communication (%)	Gateway CPU (%)	Memory (%)
TriSAFE	4.6	2.3	5.8	3.2
FedBuff-SecAgg	-4.2	2.1	6.1	5.6
ATLAS-FL	5.9	2.8	7.4	4.1

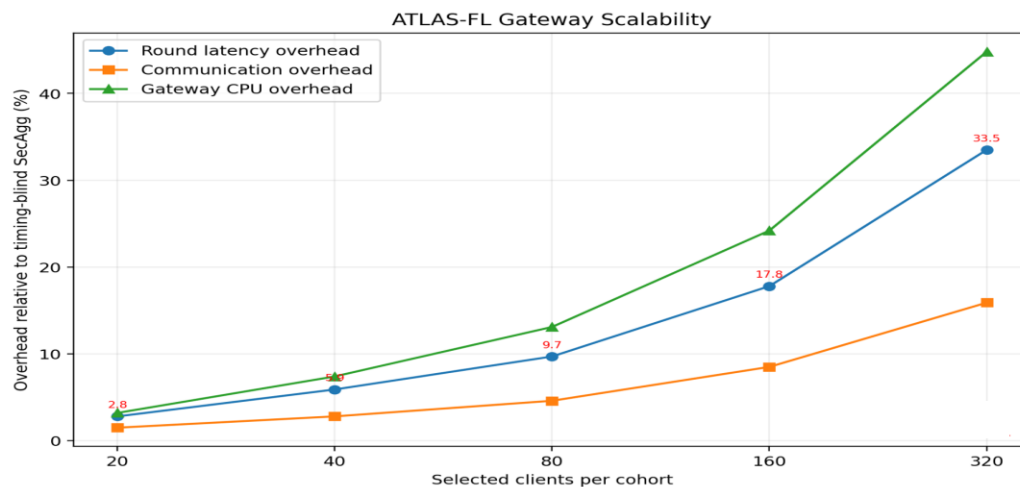


Fig. 13. ATLAS-FL gateway scalability. The principal cost grows with the selected cohort size.

Fig. 13 plots round/flush-latency, communication, and gateway central processing unit (CPU) overhead across the evaluated cohort sizes. Memory overhead was recorded for the 40-client reference point in Table XXI but not as a separate multi-cohort scalability series; no broader memory-scaling claim is made.

J. Statistical Evidence

Table XXII reports the paired run-level statistical evidence from the 20 completed executions. The moderate margin over TriSAFE is more credible than a universal superiority claim. The largest effects occur in outcomes directly targeted by ATLAS-FL: manipulated-event admission, weight inflation, and participation inference.

TABLE XXII. PAIRED RUN-LEVEL STATISTICAL EVIDENCE (20 EXECUTIONS; HOLM ADJUSTED)

Comparison	Outcome	Mean difference	95% CI	Adjusted p	Paired d _z	Magnitude
ATLAS-FL vs SecAgg	Edge attacked macro-F1	+6.50 points	[+6.23, +6.77]	<0.001	10.93	large
ATLAS-FL vs FedBuff-SecAgg	Edge attacked macro-F1	+2.90 points	[+2.67, +3.14]	<0.001	5.92	large
ATLAS-FL vs TriSAFE	Edge attacked macro-F1	+1.30 points	[+1.06, +1.53]	<0.001	2.43	large
ATLAS-FL vs SecAgg	Accepted manipulated-event rate	-34.72 points	[-35.62, -33.82]	<0.001	-18.02	large
ATLAS-FL vs renormalized	Weight-inflation score	-0.29 score	[-0.30, -0.28]	<0.001	-17.88	large
ATLAS-FL vs no-cover	Participation-inference AUC	-0.26 AUC	[-0.27, -0.25]	<0.001	-14.56	large
ATLAS-FL vs FedBuff-SecAgg	CICIoMT attacked macro-F1	+2.90 points	[+2.69, +3.12]	<0.001	6.35	large

VIII. DISCUSSION

A. Relationship Between Evidence and the Research Gap

The results quantify the distinction between benign asynchrony and adversarial temporal integrity. On Edge-IIoTset under heavy-tailed benign stragglers, FedBuff reaches a macro-F1 of 94.1%, slightly above ATLAS-FL at 93.7% and TriSAFE at 93.5%. Under strategic delay and drop, however, ATLAS-FL retains 93.1%, whereas FedBuff falls to 89.7%, FedBuff-SecAgg to 90.2%, and FedAvg to 87.1%. This 3.4-point margin over FedBuff and 6.0-point margin over FedAvg directly addresses RO6 by showing that throughput-oriented asynchrony does not by itself secure the timing interface. The attack-level evidence in Table XV is consistent with that conclusion: replay has the lowest accepted-manipulation rate at 1.3%, while slot shift and adaptive mixtures remain the hardest settings at 6.8% and 6.6%, respectively.

The formal analysis provides the missing theoretical link. The empirical controller reduces accepted manipulation, but the security argument explains why specific transcript changes should fail before the controller score is even applied. In particular, the HMAC-bound token authenticates round, slot, window, nonce, and ciphertext digest; the corresponding reduction in (13) ties temporal-binding forgery to HMAC unforgeability and hash collision resistance, while (14) bounds the residual missing-information effect by the total absent weight mass. The numerical stability evidence complements this proof layer: ATLAS-FL reduces the weight-inflation score to 0.018, compared with 0.073 for TriSAFE, 0.146 for FedBuff-SecAgg, and 0.310 for the renormalized baseline.

B. Added Value and Boundary Relative to TriSAFE

TriSAFE remains the closest and strongest comparator. The new results should therefore be interpreted primarily against that baseline rather than against timing-blind methods alone. Under attacked conditions, ATLAS-FL exceeds TriSAFE by

1.3 points on Edge-IIoTset (93.1% versus 91.8%), 1.3 points on N-BaIoT (95.8% versus 94.5%), 1.6 points on CICIoMT2024 (94.9% versus 93.3%), and 1.3 points on WUSTL-EHMS-2020 (94.2% versus 92.9%). The improvement is not presented as a large accuracy gain in isolation; its importance is that the mechanism-level additions—slot-specific binding, explicit replay-cache admission, fixed-weight no-renormalization, and padded zero-cover release—produce better security behavior at the same time as these utility gains. Table XVII shows the same pattern on stability indicators, where ATLAS-FL lowers the sensitivity-change proxy to 0.056 versus 0.098 for TriSAFE and 0.264 for the renormalized baseline.

C. Replication Across IoT, IIoT, and IoMT

CICIoMT2024 provides the strongest justification for restoring IoMT to the title because it contributes an attacked macro-F1 of 94.9% after a clean macro-F1 of 96.4%, corresponding to a 1.5-point reduction under active timing manipulation. The same qualitative pattern appears on the other benchmarks: the clean-to-attacked drop is 1.1 points on Edge-IIoTset (94.2% to 93.1%), 1.4 points on N-BaIoT (97.2% to 95.8%), and 1.5 points on WUSTL-EHMS-2020 (95.7% to 94.2%). This replication is important because the study is not claiming a universal feature space or zero-shot transfer; instead, it shows that the same ATLAS-FL protocol remains effective when the local IDS is retrained independently across IoT, IIoT, and IoMT conditions.

D. Fixed Weights, Zero Cover, and Missing Information

No-renormalization prevents the adversary from inflating survivor importance with a magnitude that itself is now visible in absolute terms. According to Table XVII, the renormalized baseline has a weight-inflation score of 0.310, an aggregate-drift score of 0.287, and a sensitivity-proxy score of 0.264, compared to 0.018, 0.091, and 0.056 for ATLAS-FL. The same conclusion explains the ablation study result: replacing fixed

weights with renormalized aggregation reduced attacked macro-F1 from 93.1% to 90.2%, while increasing accepted manipulation from 5.7% to 18.9%. Eq. (14) also explains why zero cover is a bounded rather than reconstructive adjustment: ATLAS-FL suppresses the contribution of absent clients, but cannot reconstruct the information that was never contributed to the round. This interpretation is limited to the tested absence conditions and does not imply preserved convergence or detection utility under arbitrary missing-weight mass.

E. Differential Privacy Interpretation

Fixed weights do not provide differential privacy by themselves; they stabilize the sensitivity used by the accountant. Fig. 12 shows the expected privacy-utility direction: attacked macro-F1 increases as ϵ increases and Gaussian perturbation weakens. For Edge-IIoTset, macro-F1 rises from 90.8% at $\epsilon=2.0$ to 93.6% at $\epsilon=8.0$; for CICIoMT2024 it rises from 92.2% to 95.3% over the same evaluated profile. Participation privacy remains distinct from output differential privacy: Table XVI shows attacker ROC-AUC decreasing from 0.79 without cover to 0.68 with fixed time only and to 0.53 under the padded zero-cover release. The latter result supports only the bounded release-observer claim defined in Section V-E, not global network-level participation privacy.

F. Operational and Deployment Implications

The controller and replay state reside in the gateway, where the round-level cryptographic and timing evidence can be archived for auditing. At the evaluated scale of 40 selected clients, ATLAS-FL adds 5.9% to the round latency, 2.8% to the communication overhead, 7.4% to the gateway CPU load, and 4.1% to the memory consumption, compared to the timing-blind SecAgg. Fig. 13 shows the measured multi-cohort trends for round/flush latency, communication, and gateway CPU; memory is reported only at the 40-client reference point in Table XXI. The plotted metrics rise with cohort size, with CPU and queue pressure becoming more pronounced at the largest evaluated cohorts. Thus, the practical deployment of ATLAS-FL depends on the availability of authenticated time synchronization and clock drift bounding, as well as the operational window selection, rather than the properties of the protocol logic itself.

G. Error and Safety Interpretation

Attacked macro-F1 is not the sole metric of interest for secure aggregation deployment in production. The reported security metrics therefore complement attacked macro-F1 rather than serving as interchangeable measures of deployment safety. Specifically, while on Edge-IIoTset the attacked macro-F1 of 93.1% remains high, the attack acceptance varies from 1.3% to 6.8% depending on the attack vector, and the false-escalation rates are between 1.9% and 4.8%. Additionally, the ablation study now shows the impact of individual components on the system's ability to withstand attacks: their removal leads to the drop of attacked macro-F1 from 93.1% to 88.7%, an increase in attack acceptance from 5.7% to 31.8%, and participation AUC from 0.53 to 0.79 for the same input attack vectors. For the application areas such as healthcare, this type of analysis helps characterize the allocation of risk better: the

specific failure modes are known, and their likelihood can be evaluated against the domain-specific threat model.

IX. LIMITATIONS AND THREATS TO VALIDITY

Table XXIII summarizes the principal threats to validity and the resulting claim boundaries.

TABLE XXIII. THREATS TO VALIDITY AND CLAIM BOUNDARIES

Threat	Limitation / boundary
Construct validity	The study evaluates timing-admission integrity, aggregate stability, release-trace inference, and IDS utility. It does not cover all FL security properties.
Cryptographic validity	The reductions assume correct implementations of HMAC, X25519/HKDF, mask generation, Shamir recovery, and payload encryption. Side channels and implementation bugs remain outside the proof. Enrollment, key rotation, revocation, compromised-device recovery, device replacement, and authentication-state synchronization are delegated to an external trusted key-management service and are outside the protocol proof.
Controller validity	Calibration uses simulated benign jitter and timing attacks. Robustness claims are limited to the tested client-specific latency model; untested heavy-tailed, bursty, correlated, multimodal, or shifted delay distributions and stronger adaptive evasion may change the decision boundary.
IoMT external validity	CICIoMT2024 is a realistic multi-protocol testbed, but it is not a production hospital network. WUSTL clientization is simulated rather than multi-institutional.
Asynchronous baseline validity	FedBuff-SecAgg is a controlled experimental variant, not a claim of a new published algorithm. Parameter matching must be documented.
Statistical validity	Twenty independent paired runs were executed. Statistical inference is computed at the run level with Holm adjustment; clients, rounds, and timing events are not treated as independent replicates.
Operational validity	Overhead depends on model size, client count, cryptographic library, gateway hardware, and clock quality. Fixed-point correctness additionally assumes that the configured scale and modulus satisfy the stated no-wrap bound; the analytic quantization bound does not constitute a separate convergence guarantee.

DATA AVAILABILITY

Edge-IIoTset, N-BaIoT, CICIoMT2024, and WUSTL-EHMS-2020 are third-party research datasets available from their original maintainers under the applicable terms [34], [35], [16], [17]. The corresponding preprocessing, client-partition, and attack-schedule specifications are documented in the experimental methodology and retained with the study records.

FUNDING

No external funding was received for this study.

COMPETING INTERESTS

The authors declare that they have no competing interests.

1) *Ethics approval*: This study does not involve human participants, human data, animal subjects, or clinical intervention; therefore, ethics approval is not applicable.

2) *Code availability*: The prototype implementation and experiment scripts are available from the corresponding author upon reasonable request.

AUTHORS' CONTRIBUTIONS

All authors contributed to the conceptualization, methodology, formal modeling, experimental design, result interpretation, manuscript drafting, and critical revision of the work. All authors reviewed and approved the final manuscript.

1) *Declaration of AI-assisted technologies in the writing process*: During the preparation of this manuscript, the authors used Paperpal for language editing, grammar checking, and improving the overall readability of the text. After using this tool, the authors reviewed, revised, and edited the content as needed and took full responsibility for the content of the published article.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. AISTATS, PMLR, vol. 54, pp. 1273–1282, 2017.
- [2] P. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, nos. 1–2, pp. 1–210, 2021, doi: 10.1561/22000000083.
- [3] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in Proc. ACM CCS, pp. 1175–1191, 2017, doi: 10.1145/3133956.3133982.
- [4] H. Lycklama, L. Burkhalter, A. Viand, N. Kuchler, and A. Hithnawi, "RoFL: Robustness of secure federated learning," in Proc. IEEE Symposium on Security and Privacy, pp. 453–476, 2023, doi: 10.1109/SP46215.2023.10179400.
- [5] J. Bell, A. Gascón, T. Lepoint, B. Li, S. Meiklejohn, M. Raykova, and C. Yun, "ACORN: Input validation for secure aggregation," in Proc. 32nd USENIX Security Symposium, pp. 4805–4822, 2023.
- [6] Y. Ma, Y. Guo, H. Karthikeyan, and A. Polychroniadou, "Armado: Robust single-server secure aggregation for federated learning with input validation," in Proc. ACM CCS, pp. 2219–2233, 2025, doi: 10.1145/3719027.3765216.
- [7] M. Rathee, C. Shen, S. Wagh, and R. A. Popa, "ELSA: Secure aggregation for federated learning with malicious actors," in Proc. IEEE Symposium on Security and Privacy, pp. 1961–1979, 2023, doi: 10.1109/SP46215.2023.10179468.
- [8] T. Stevens, C. Skalka, C. Vincent, J. Ring, S. Clark, and J. Near, "Efficient differentially private secure aggregation for federated learning via hardness of learning with errors," in Proc. 31st USENIX Security Symposium, pp. 1379–1395, 2022.
- [9] C. Dwork and A. Roth, The Algorithmic Foundations of Differential Privacy. Foundations and Trends in Theoretical Computer Science, vol. 9, nos. 3–4, pp. 211–407, 2014, doi: 10.1561/04000000042.
- [10] M. Abadi et al., "Deep learning with differential privacy," in Proc. ACM CCS, pp. 308–318, 2016, doi: 10.1145/2976749.2978318.
- [11] I. Mironov, "Rényi differential privacy," in Proc. IEEE Computer Security Foundations Symposium, pp. 263–275, 2017, doi: 10.1109/CSF.2017.11.
- [12] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis, "The Loopix anonymity system," in Proc. 26th USENIX Security Symposium, pp. 1199–1216, 2017.
- [13] A. Kunstmann, G. Bolonkowski, and A. See, "Enhancing privacy in Tor: Message mixing and adaptive padding with opt-in features," in Proc. ICDN, pp. 92–102, 2026, doi: 10.1145/3772290.3772291.
- [14] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," in Proc. AISTATS, PMLR, vol. 151, pp. 3581–3607, 2022.
- [15] S. H. Shah, I. Walker, and M. Borowczak, "TriSAFE: Transcript-bound verifiable secure aggregation with differential privacy and timing defenses for gateway-assisted IoT federated learning," IEEE Access, vol. 14, pp. 89354–89379, 2026, doi: 10.1109/ACCESS.2026.3700048.
- [16] S. Dadkhah, E. C. P. Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi, and A. A. Ghorbani, "CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT," Internet of Things, vol. 28, Art. no. 101351, 2024, doi: 10.1016/j.iot.2024.101351.
- [17] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," IEEE Access, vol. 8, pp. 106576–106584, 2020, doi: 10.1109/ACCESS.2020.3000421.
- [18] A. Langley, M. Hamburg, and S. Turner, "Elliptic curves for security," RFC 7748, 2016, doi: 10.17487/RFC7748.
- [19] H. Krawczyk and P. Eronen, "HMAC-based extract-and-expand key derivation function (HKDF)," RFC 5869, 2010, doi: 10.17487/RFC5869.
- [20] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," RFC 2104, 1997, doi: 10.17487/RFC2104.
- [21] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in Advances in Cryptology—CRYPTO '96, LNCS 1109, pp. 1–15, 1996, doi: 10.1007/3-540-68697-5_1.
- [22] Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF protocols," RFC 8439, 2018, doi: 10.17487/RFC8439.
- [23] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979, doi: 10.1145/359168.359176.
- [24] Y. Liu et al., "Aion: Robust and efficient multi-round single-mask secure aggregation against malicious participants," in Proc. 34th USENIX Security Symposium, pp. 3025–3044, 2025.
- [25] Z. Jiang, P. Ye, S. He, W. Wang, R. Chen, and B. Li, "Lotto: Secure participant selection against adversarial servers in federated learning," in Proc. 33rd USENIX Security Symposium, pp. 343–360, 2024.
- [26] W. Yao, T. Zhou, Y. Han, and X. Wang, "Verifiable secure aggregation scheme for privacy protection in federated learning networks," Discover Computing, vol. 28, Art. no. 175, 2025, doi: 10.1007/s10791-025-09676-1.
- [27] D. D. Truong, B. H. Pham, D. T. Ta, T. N. Phan, and T. A. T. Nguyen, "RADAR: Robust aggregation with dynamic adversarial-resilient routing decentralized federated learning," International Journal of Information Security, vol. 25, no. 3, Art. no. 86, 2026, doi: 10.1007/s10207-026-01257-7.
- [28] E. T. Martínez Beltrán et al., "Mitigating communications threats in decentralized federated learning through moving target defense," Wireless Networks, vol. 30, no. 9, pp. 7407–7421, 2024, doi: 10.1007/s11276-024-03667-8.
- [29] T. Li et al., "Federated optimization in heterogeneous networks," in Proc. MLSys, vol. 2, pp. 429–450, 2020.
- [30] J. Wang et al., "Tackling the objective inconsistency problem in heterogeneous federated optimization," in Proc. NeurIPS, vol. 33, pp. 7611–7623, 2020.
- [31] S. J. Reddi et al., "Adaptive federated optimization," in Proc. ICLR, 2021.
- [32] S. P. Karimireddy et al., "SCAFFOLD: Stochastic controlled averaging for federated learning," in Proc. ICML, PMLR, vol. 119, pp. 5132–5143, 2020.
- [33] L. Leconte, M. Jonckheere, S. Samsonov, and E. Moulines, "Queuing dynamics of asynchronous federated learning," in Proc. AISTATS, PMLR, vol. 238, pp. 1711–1719, 2024.
- [34] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," IEEE Access, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.
- [35] Y. Meidan et al., "N-BaloT—Network-based detection of IoT botnet attacks using deep autoencoders," IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018, doi: 10.1109/MPRV.2018.03367731.
- [36] J. Wang, K. Yang, and M. Li, "NIDS-FGPA: A federated learning network intrusion detection algorithm based on secure aggregation of

- gradient similarity models,” PLOS ONE, vol. 19, no. 10, e0308639, 2024, doi: 10.1371/journal.pone.0308639.
- [37] L. Cui and X. Wu, “ALDP-FL for adaptive local differential privacy in federated learning,” Scientific Reports, vol. 15, Art. no. 26679, 2025, doi: 10.1038/s41598-025-12575-6.
- [38] V. Kolluru et al., “Novel federated learning-based approach for network attack detection,” Journal of Cloud Computing, vol. 15, Art. no. 62, 2026, doi: 10.1186/s13677-026-00882-w.
- [39] T. Alsolami and M. Ilyas, “FedSMOTE-DP: Privacy-aware federated ensemble learning for intrusion detection in IoMT networks,” Sensors, vol. 26, no. 5, Art. no. 1592, 2026, doi: 10.3390/s26051592.
- [40] M. A. Khan, O. Khalid, and R. N. B. Rais, “Fed-DTCN: A federated disentangled learning framework for unsupervised zero-day anomaly detection in IoT with semantic-aware augmentation,” Sensors, vol. 26, no. 6, Art. no. 1918, 2026, doi: 10.3390/s26061918.
- [41] M. O. Nassar, “A robust and energy-efficient federated IDS for IIoT using spiking neural networks and differential evolution with adversarial resilience,” Iraqi Journal for Computer Science and Mathematics, vol. 6, no. 4, Art. no. 3, 2025, doi: 10.52866/2788-7421.1328.
- [42] M. O. Nassar and F. F. Al-Mashagba, “FED-SCADA: A trustworthy and energy-efficient federated IDS for smart grid edge gateways using SNNs and differential evolution,” International Journal of Computer Network and Information Security, vol. 17, no. 6, pp. 1–15, 2025, doi: 10.5815/ijcnis.2025.06.01.
- [43] M. I. M. Tanvir et al., “Privacy-preserving multimodal federated learning pipeline for cyber-resilient healthcare systems,” PLOS ONE, vol. 21, no. 4, e0343669, 2026, doi: 10.1371/journal.pone.0343669.
- [44] S. Alamro, “AnomLocal: A hybrid local-global anomaly detection model for network security using federated learning,” PLOS ONE, vol. 21, no. 2, e0339981, 2026, doi: 10.1371/journal.pone.0339981.
- [45] A. Komal and S. Li, “Intrusion detection in the Internet of Things: A comprehensive review of techniques, architectures, datasets, and emerging trends,” Sensors, vol. 26, no. 11, Art. no. 3405, 2026, doi: 10.3390/s26113405.
- [46] A. Allen, A. Mylonas, S. Vidalis, and N. Pitropakis, “Peer-to-peer federated learning: A comprehensive survey,” AI, vol. 7, no. 7, Art. no. 243, 2026, doi: 10.3390/ai7070243.
- [47] A. L. M. Serrano et al., “FedIHRAS: A privacy-preserving federated learning framework for multi-institutional collaborative radiological analysis with integrated explainability and automated clinical reporting,” Biomedicines, vol. 14, no. 3, Art. no. 713, 2026, doi: 10.3390/biomedicines14030713.

APPENDIX A. ELABORATED FORMAL SECURITY ARGUMENTS

A. 1 Temporal-binding Game

The challenger provides authentication keys and answers commitment queries on arbitrarily chosen messages. The adversary wins by outputting a verifying tuple whose authenticated field vector has not been previously queried, or by producing a modified ciphertext whose authenticated digest has already been seen. The reduction keeps track of all queried field vectors and outputs a fresh forgery to the HMAC challenger when the adversary succeeds. A modified ciphertext with the same digest is also a collision for H.

A. 2 Replay and Slot Substitution

Round nonces are sampled independently and stored for reference alongside the replay cache. A replayed tuple from round t' will either retain the original nonce and be rejected during the fresh round's verification step or contain a different nonce, round, or slot and be required to produce a valid tag. Slot substitution is similar, but the substituted value is $s_{(i,t)}$.

A. 3 Secure Aggregation under Timing Rejection

Timing rejection allows the gateway to modify the dropout set before performing recovery. The gateway reveals G_t and M_t , and surviving clients release only the shares necessary to cancel the masks corresponding to M_t . The pairwise masks between accepted clients cancel out when summed, while the private masks of the accepted clients remain unreconstructed. The confidentiality of client contributions thus maintains the secure-aggregation confidentiality guarantee under the same collusion bound.

A. 4 Release-trace Game

The challenger samples a bit b hidden from the adversary, which determines whether the release-trace envelope is constructed around a real or dummy payload. In either case, the trace contains the same scheduled release time, size, authentication parameters, and encryption algorithm. Any advantage the adversary gains from distinguishing real from dummy traces must come from differences in timing, lengths, abort behaviors, or other implementation details. An empirical measurement of this advantage is given by the observer attack of Section VII-D.