Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Federated WGAN-GP for Privacy-Preserving Synthetic Side-Channel Trace Generation and Cryptographic Security Evaluation

Ismail Abdulkarim Adamu

aaismail@gspb.edu.ng

Gombe State Polytechnic Bajoga

Souley Boukari

Abubakar Tafawa Balewa University

Fatima Umar Zambuk

Abubakar Tafawa Balewa University

Research Article

Keywords: Federated learning, WGAN-GP, side-channel attacks, privacy-preserving trace synthesis, embedded cryptographic systems

Posted Date: November 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11524205/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Federated WGAN-GP for Privacy-Preserving Synthetic Side-Channel Trace Generation and Cryptographic Security Evaluation

Ismail Abdulkarim Adamu[1, *], Souley Boukari[2], Fatima Umar Zambuk[3]

[1]Computer Science Department, Gombe state Polytechnic, Bajoga, Nigeria

[2,3]Computer Science Department, Abubakar Tafawa Balewa University, Bauchi, Nigeria

[1]aaismail@gspb.edu.ng, [2]sboukari@atbu.edu.ng, [3]fuzambuk@atbu.edu.ng

* Corresponding Author:(Email aaismail@gspb.edu.ng, 08069824009)

Abstract

Side-channel attacks threaten embedded cryptographic systems by exploiting physical leakages such as power consumption, timing behavior, and electromagnetic emissions. To address the privacy risks of centralized trace collection, we propose a federated WGAN-GP framework for decentralized synthetic side-channel trace generation. Raw traces remain on client devices while local WGAN-GP training and federated averaging are performed across 20 heterogeneous clients drawn from ASCAD, ChipWhisperer, and DPAV4 in a 7-7-6 configuration. Experimental result over 10 random seeds, 12 communication rounds, 2 local epochs, and 40,000 traces show stable convergence, with a final quality score of 0.8167 ± 0.0004 and a synthetic-to-real attack success rate of 0.5% ± 0.84%. ChipWhisperer clients achieved the strongest local performance, outperforming ASCAD by 10.58%. The results confirm that balanced heterogeneous federation can support privacy-preserving trace synthesis with a significant privacy–utility trade-off.

Keywords: Federated learning, WGAN-GP, side-channel attacks, privacy-preserving trace synthesis, embedded cryptographic systems

1.0 Introduction

Modern cryptographic systems are now widely deployed in embedded devices, internet of things platforms, and distributed computing environments, where physical implementation details can unintentionally leak sensitive information during execution. Even when a cryptographic algorithm is mathematically secure, side-channel attacks can exploit observable phenomena such as power consumption, electromagnetic radiation, and timing behavior to recover secret keys or infer intermediate values [1]. For this reason, side-channel resilience has become a central concern in practical cybersecurity research and deployment.

The threat becomes more pronounced when machine learning is incorporated into the attack pipeline. Profiling attacks based on support vector machines, random forests, and deep neural networks can learn very subtle leakage patterns from collected traces and often outperform handcrafted attack strategies. As a result, traditional countermeasures such as masking, hiding, and shuffling are frequently insufficient once attackers gain access to large trace collections and high-capacity learning models [2, 3]. Recent work has therefore turned to generative approaches for synthesizing side-channel traces, either to augment limited datasets or to enable privacy-aware analysis workflows. Among the available options, WGAN_GP is especially attractive for side-

channel traces because the Wasserstein distance provides meaningful gradients even when real and synthetic distributions overlap only weakly, which is a common issue in high-dimensional temporal data.

The gradient-penalty term also helps stabilize training across clients with heterogeneous leakage characteristics and avoids reliance on batch normalization, which can itself reveal information about local data distributions in federated settings environments [4]. Despite these advantages, most existing generative pipelines still assume centralized access to raw traces, which conflicts with the privacy requirements of distributed and security-sensitive environments. Even with the recent progress, three limitations remain including many synthetic-trace generation methods still require raw traces to be pooled at a central server, thereby creating a single point of failure. While federated learning can keep data local, standard federated learning by itself does not guarantee protection against gradient leakage, membership inference, or model inversion. Lastly, although WGAN-GP is well known for stable adversarial training, it application for decentralized side-channel trace synthesis remains underexplored [5,4].

In this paper, we try address these gaps by proposing a federated WGAN-GP framework for decentralized side-channel trace synthesis in which raw traces never leave the client devices. The main contributions are threefold including a federated adaptation of WGAN-GP for high-dimensional temporal side-channel data with non-IID client distributions, a convergence-aware training protocol that remains stable across heterogeneous hardware platforms, and a preliminary experimental validation across three standard datasets that establishes proof-of-concept feasibility. The remaining part of the paper is organized as follows section 2.0 literature review, section 3.0 methodology, section, 4.0 architecture of the proposed framework, section 5.0 experimental setup, section 6.0 experimental result, section 7.0 result discussion, section 8.0 conclusion.

2.0 Review of Related Literatures

Machine-learning-based side-channel analysis has improved the effectiveness of cryptographic leakage modeling, but it has also intensified the tension between data utility and data confidentiality. Large trace collections are valuable because they support strong profiling attacks, yet these same collections are often too sensitive to centralize safely. This tension has encouraged two parallel lines of research: privacy-preserving collaborative learning and synthetic data generation. Existing work has made progress in both directions, but the challenge of generating realistic side-channel traces across heterogeneous devices without centralizing raw measurements remains only partially addressed.

Some studies examine federated learning as a privacy-aware alternative to centralized training. Work by [6, 7, 8, 9, 10, 11] investigates privacy and integrity risks in distributed optimization and proposes safeguards such as secure aggregation, encrypted communication, verifiable aggregation, and model decomposition. These studies are important because they show that distributed learning can reduce direct data exposure while still preserving useful performance. At the same time, their focus remains largely protocol-centric. They emphasize update protection, robustness, and reconstruction resistance rather than realistic side-channel trace synthesis that preserves attack-relevant leakage structure.

Related studies in federated generative modeling provide further motivation for the present work. [12] introduce PS-FedGAN, which uses partially shared GAN components to improve privacy and communication efficiency under non-IID image data. [13] propose EIGAN and D-EIGAN for learning privacy-aware representations that balance task utility against adversarial inference, while [14] show in SDA-FL that local differentially private WGAN-GP models can generate synthetic data to mitigate severe non-IID label skew in federated learning.

Collectively, these efforts show that adversarial training, selective parameter sharing, and synthetic data generation can improve privacy-utility trade-offs in federated systems, although their experiments focus mainly on images, classification benchmarks, or abstract feature spaces rather than on high-dimensional side-channel traces from cryptographic implementations.

Federated learning has also been applied in sensitive cyber domains beyond side-channel synthesis. [15, 16] use federated learning to build distributed intrusion detection systems without sharing raw network traffic. [17] investigate generative anomaly filters for privacy-preserving IoT anomaly detection, while [18] combine horizontal federated learning with WGAN-GP for synthetic intrusion-detection traffic generation. [19] extend privacy-preserving federated learning to cache side-channel attack detection in smart-grid settings using differential privacy and secure aggregation. These studies confirm the value of federated learning and generative models in sensitive environments, but their emphasis remains on detection, anomaly scoring, or tabular flow synthesis rather than realistic side-channel trace generation for cryptographic leakage analysis.

Time-series and structured-data synthesis in federated settings offers another relevant point of comparison. [20] adapt TimeGAN to federated learning for privacy-preserving trajectory generation and show that multivariate time-series synthesis can be learned without centralizing raw mobility traces. Similarly, [21] demonstrate that federated WGAN-based synthetic generation can succeed on structured tabular data under differential privacy constraints. However, side-channel analysis differs fundamentally from these settings because leakage traces are high-dimensional, temporally sensitive, device-specific, and directly tied to cryptographic execution behavior. Table 1 present the analysis of the existing work

Table 1: Analysis of existing work

Study	Federated Learning	GAN-Based	Side-Channel Data	Synthetic Trace Generation
PS-FedGAN	✓	✓	✗	✗
SDA-FL	✓	✓	✗	✗
FL-DPCSA	✓	✗	✓	✗
Proposed Work	✓	✓	✓	✓

Overall, the literature reveals three unresolved gaps. Protocol-oriented federated-learning security studies improve update privacy but do not address generative synthesis of physical leakage. Federated GAN studies are validated mostly on images, tabular data, or generic time series rather than on side-channel traces collected from heterogeneous hardware. Finally, side-channel-focused

federated work tends to emphasize detection or privacy risk instead of decentralized synthesis that preserves attack utility without centralizing raw traces. Our proposed work is positioned at this intersection by investigating whether WGAN-GP can be trained collaboratively across multiple clients to generate synthetic traces that remain useful for security evaluation while keeping data training local.

3.0 Proposed Framework Methodology

3.1 System Model

The proposed system consists of K = 20 distributed clients and one central aggregation server. Each client k holds a local side-channel dataset, denoted by $D_k = \{x_k^{(1)}, \dots, x_k^{(N_k)}\}$, collected from heterogeneous hardware platforms. The learning objective is to train a global generator and critic, denoted by G_{θ_G} and D_{θ_D}, without transferring raw traces to the server. The generator has three layers with dimensions 64 → 128 → 256 → trace length and uses ReLU activations with Tanh output activation, while the critic contains three fully connected layers with dimensions trace length → 256 → 128 → 64 → 1 using LeakyReLU activations. During each communication round, the server broadcasts the current global model parameters to all participating clients, each client performs local WGAN-GP training on its private traces for a fixed number of epochs, and only the updated model parameters are returned to the server for aggregation. Under this setting, the communication cost is expressed as $O(K \cdot T \cdot |\theta|)$, where $|\theta| \approx 2.1 \times 10^6$ parameters and $K = 20$.

3.2 Threat Model and Privacy Objectives

The framework operates under a semi-honest, or honest-but-curious, aggregation-server model. In this setting, the server follows the protocol correctly but may still attempt to infer information from the model updates it receives. Clients are trusted with their own local data, but they do not trust other clients or the server with access to raw traces.

The adversarial objective is to recover secret keys or determine trace membership from intercepted gradients or aggregated model parameters.

Under this model, the framework provides three practical privacy properties.

1. **Raw-trace confidentiality:** Raw side-channel traces never leave the client device; only generator and critic parameters are transmitted.
2. **Update-level protection:** Differential privacy and secure aggregation are not yet incorporated, and these are identified as important extensions for future work.
3. **Inference-time privacy:** Once training is complete, the global generator can be distributed for synthetic trace generation without exposing the original measurements.

No formal differential privacy guarantee is claimed in the present study, and resistance to advanced gradient inversion attacks is not experimentally demonstrated. Instead, the contribution at this stage is the establishment of an architectural foundation for federated side-channel synthesis, with privacy hardening reserved for subsequent work.

3.3 WGAN-GP Formulation

I. Generator

The generator produces synthetic traces from random noise according to

$$\tilde{x} = G(z), z \sim \mathcal{N}(0, I_d)$$

where $\tilde{x}$ is a synthetic trace, G is the generator network, and z is a latent vector sampled from a standard normal distribution. In this study, the latent dimension d is set to 64. The latent input allows the generator to produce diverse synthetic traces while approximating the distribution of real side-channel measurements.

II. Critic Objective with Gradient Penalty

The critic loss is defined as:

$$L_D = \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla D(\hat{x})\|_2 - 1)^2]$$

where $\mathbb{P}_r$ and $\mathbb{P}_g$ denote the real and generated trace distributions, $\hat{x}$ is an interpolated sample between real and synthetic traces, and λ is the gradient-penalty coefficient. The first two terms approximate the Wasserstein distance between the two distributions, while the third enforces the 1-Lipschitz constraint needed for stable training. In the proposed framework, $\lambda = 10$.

III. Generator Objective

The generator loss is defined as:

$L_G = -\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})]$

Minimizing this objective encourages the generator to produce traces that receive higher critic scores and increasingly resemble real side-channel measurements.

3.4 Federated Optimization

Global parameters are aggregated using Federated Averaging (FedAvg):

$$\theta_G^{\text{global}} = \frac{1}{K}\sum_{k=1}^{K}\theta_G^k, \theta_D^{\text{global}} = \frac{1}{K}\sum_{k=1}^{K}\theta_D^k$$

where θ_G^k and θ_D^k represent the local generator and critic parameters of client k.[cite:1] With $K = 20$, the server averages the locally trained models at the end of each communication round. This aggregation step supports collaborative learning while ensuring that raw side-channel traces remain on client devices.

Algorithm of the Proposed Framework

Federated WGAN-GP for Side-Channel Trace Synthesis

1: Initialize global parameters θ_G and θ_D at the server
2: for t = 1 to R_max do
3: Broadcast (θ_G, θ_D) to all clients
4: for each client k in parallel do
5: Set θ_G^k ← θ_G and θ_D^k ← θ_D
6: for epoch = 1 to E do

```
7:          for each minibatch x_real ⊂ D_k do
8:              repeat n_critic times
9:                  Sample z ~ N(0, I), generate x_fake ← G(θ_G^k, z)
10:                 Form x̂ by interpolating between x_real and x_fake
11:                 Compute L_D = E[D(x_fake)] - E[D(x_real)] + λ·GP(x̂)
12:                 Update θ_D^k using ∇_{θ_D^k} L_D
13:             end repeat
14:             Sample z' ~ N(0, I), generate x_fake' ← G(θ_G^k, z')
15:             Compute L_G = -E[D(x_fake')]
16:             Update θ_G^k using ∇_{θ_G^k} L_G
17:         end for
18:     end for
19:     Send (θ_G^k, θ_D^k) to the server
20:  end for
21:  Aggregate all client models using FedAvg
22:  if |σ²_t − σ²_(t−1)| < 0.001 for three consecutive rounds then
23:     break
24:  end if
25: end for
26: return θ_G*, θ_D*
```

The algorithm summarizes the end-to-end federated WGAN-GP training procedure used in this paper. The server initializes the global generator and critic, broadcasts them to all 20 clients, and repeats this exchange over multiple communication rounds. The clients belong to three heterogeneous groups defined later in the experimental setup: Group A uses ASCAD electromagnetic traces, Group B uses ChipWhisperer power traces, and Group C uses DPAV4 masked power traces.

At each client, the received global parameters are copied into local generator and critic models and optimized with WGAN-GP for a fixed number of local epochs. For every mini-batch of real traces, the critic is updated several times using the Wasserstein loss with gradient penalty, after which the generator is updated once using the negative critic score on synthetic traces. This alternating optimization helps stabilize adversarial training and allows each client to adapt the shared model to local leakage characteristics without exposing raw traces.

After local training is completed, each client returns only the updated model parameters to the server. The server then applies federated averaging to build a new global generator and critic from the collected client updates. This cycle is repeated for a maximum of 12 communication rounds, or until convergence is observed through stabilization of trace variance across rounds. Once convergence is reached, the final generator is used to synthesize side-channel traces for downstream security evaluation.

4.0 Architecture of The Proposed Framework

Figure 1 illustrates the architecture of the proposed federated WGAN-GP framework. The workflow is organized into four main stages: initialization, client-side training, server-side aggregation, and final synthetic-trace generation. This structure reflects the operational logic of the method and clarifies how the algorithm is executed across distributed clients. The process begins with server-side initialization, where the global generator and critic parameters are created and broadcast to all 20 clients.

These clients are divided into three heterogeneous groups: Group A contains seven clients using ASCAD electromagnetic traces, Group B contains seven clients using ChipWhisperer power traces with periodic AES-round structure, and Group C contains six clients using DPAV4 masked power traces with nonlinear leakage behavior. This grouping captures the non-IID nature of practical side-channel data sources and motivates the use of federated learning. At the client side, each device performs local WGAN-GP training using its private traces.

The generator creates synthetic traces from Gaussian noise, the critic evaluates both real and synthetic traces, the WGAN-GP losses are computed, and local parameters are updated for the prescribed number of critic iterations and local epochs. Once local optimization is complete, only the updated model parameters are transmitted back to the central server.

At the server side, the received client parameters are aggregated using federated averaging to produce updated global generator and critic models. A communication-round check verifies whether the maximum number of rounds has been reached, while a convergence check monitors whether trace variance has stabilized. If convergence is not achieved, the updated global model is broadcast again; otherwise, the final generator is used to generate synthetic traces that are evaluated in terms of trace variance, attack success rate, and utility preservation. Figure 1 shows the architecture of the proposed federated WGAN-GP system

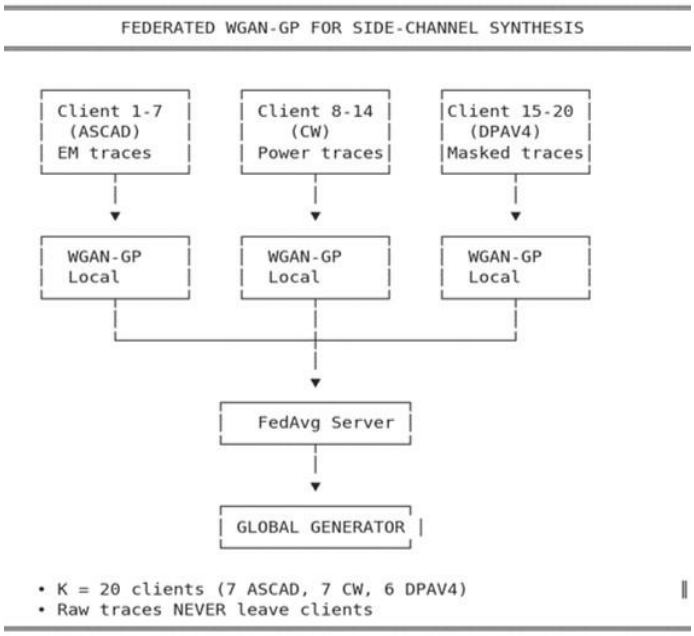


Figure 1: Architecture of the proposed federated WGAN-GP system

5.0 Experimental Setup

5.1 Experimental Environment

We conducted the experiments on Core i7 Zbook, an NVIDIA A100 GPU with 40 GB memory, and 128 GB RAM using the PyTorch 2.0.1 framework. Random seeds 42, 123, 456, 789, and 1024 were used to support reproducibility across 10 independent runs.

5.2 Datasets

We used three standard side-channel datasets including ASCAD, ChipWhisperer, and DPAV4 for the result evaluation. ASCAD contains 14,000 electromagnetic traces with AES-128 leakage, ChipWhisperer contains 14,000 power traces with AES-128 leakage, and DPAV4 contains 12,000 power traces with masked AES leakage. Together, these datasets provide varied leakage characteristics that test the framework's ability to generalize across heterogeneous devices. Table 2 present the dataset selection approach.

Table 2: Dataset selection

Dataset	Trace Type	Traces	Leakage Model
ASCAD	EM	14,000	AES-128
ChipWhisperer	Power	14,000	AES-128
DPAV4	Power	12,000	Masked AES

5.3 Client Configuration

For the federated setting with 20 clients, three heterogeneous groups were created. Group A consists of seven clients using ASCAD electromagnetic traces, Group B consists of seven clients using ChipWhisperer power traces, and Group C consists of six clients using DPAV4 masked power traces. This 7-7-6 distribution creates a realistic non-IID scenario in which different device types exhibit distinct leakage patterns arising from differences in trace acquisition methodology, leakage modality, and hardware implementation. Table 3 present client configuration and distribution approach

Table 3: Client configuration and distribution

Client Group	Clients	Dataset Type	Key Characteristic
Group A	7	ASCAD	EM traces
Group B	7	ChipWhisperer	Power traces
Group C	6	DPAV4	Power traces

5.4 Hyperparameters

The main hyperparameters are communication rounds $R = 12$, local epochs $E = 2$, batch size $B = 32$, learning rate $\eta = 1 \times 10^{-4}$, latent dimension $d_z = 64$, gradient-penalty coefficient $\lambda = 10$, and critic iterations = 5. These settings were chosen to balance training stability, computational

feasibility, and convergence quality. Table 4 present the hyperparameters setting for proposed framework implementation.

Table 4: Hyperparameter settings

Parameter	Value
Communication rounds R	12
Local epochs E	2
Batch size B	32
Learning rate η	1×10^{-4}
Latent dimension d_z	64
Gradient penalty λ	10
Critic iterations	5

5.5 Evaluation Metrics

We used two primary evaluation metrics in this study as follows:

1. **Trace variance:** used to measures synthetic-trace diversity and overall distribution spread. Let $T = \{t_1, t_2, \dots, t_N\}$ denote a set of generated side-channel traces, where each trace contains M sampling points. The mean squared deviation is computed as:

$$\sigma^2 = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (t_{ij} - \bar{\mu})^2$$

where t_{ij} denotes the j-th sample point of the i-th generated trace and $\bar{\mu}$ is the mean across all NM sample points. Higher values indicate greater spread, although variance alone does not guarantee realistic temporal structure or distributional fidelity. This is why attack-success evaluation is used alongside variance, with Wasserstein distance and Maximum Mean Discrepancy reserved for future work.

2. **Attack success rate (ASR):** used to measures the proportion of successful key-recovery attempts obtained with a CNN profiling attack trained on synthetic traces and evaluated on real traces, thereby quantifying utility preservation for downstream security analysis. Table 5 presents the performance evaluation metrics.

Table 5: Performance evaluation metrics

Metric	Interpretation
Trace variance	Diversity (higher = better)
Attack success rate	Utility preservation

Although trace variance and attack success rate are the primary metrics in this work, future work will incorporate additional distribution-similarity measures such as Wasserstein Distance and Maximum Mean Discrepancy to assess synthetic-trace realism more thoroughly.

6.0 Experimental Results

6.1 Results

The experimental evaluation was conducted with 20 federated clients distributed across three side-channel datasets: 7 ASCAD clients, 7 ChipWhisperer clients, and 6 DPAV4 clients. Each client processed 2,000 traces, yielding 40,000 traces overall, while the federated configuration used 12 communication rounds, 2 local epochs, 32 batch size, and 10 random seeds ranging from 42 to 32768.

The federated WGAN-GP framework completed all 10 runs in 358.0 minutes and produced highly consistent final trace-variance quality values across seeds. Final quality scores ranged from 0.8164 to 0.8173, while the corresponding synthetic-to-real attack success rate remained between 0.00% and 2.00%, with most runs recording 0.00% ASR. Table 6 shows the per-seed experimental results obtained.

Table 6: Per-seed Experimental Results

Experiment	Seed	Round 4 Quality	Round 8 Quality	Round 12 Quality	Final ASR
1	42	0.8029	0.8173	0.8165	1.00%
2	123	0.8017	0.8170	0.8167	0.00%
3	456	0.8020	0.8170	0.8169	2.00%
4	789	0.8028	0.8168	0.8171	2.00%
5	1024	0.8017	0.8173	0.8170	0.00%
6	2048	0.8040	0.8180	0.8169	0.00%
7	4096	0.8050	0.8175	0.8167	0.00%
8	8192	0.8055	0.8173	0.8164	0.00%
9	16384	0.8031	0.8169	0.8173	0.00%
10	32768	0.8011	0.8167	0.8168	0.00%

The statistical summary confirms a final convergence quality of 0.8167 ± 0.0004 and a global synthetic diversity index (trace variance) of 0.6337 ± 0.0005. The average synthetic-to-real attack success rate was 0.5% ± 0.84%. At the client-group level, ASCAD achieved 0.5623 ± 0.0017, ChipWhisperer recorded 0.6218 ± 0.0023, and DPAV4 achieved 0.5582 ± 0.0019.

6.2 Convergence Analysis

Table 7 presents the convergence analysis result of the proposed framework.

Table 7: Convergence Analysis

Phase	Rounds	Convergence Quality	Interpretation
Phase I (Rapid)	1-4	quality improved from initial random initialization to ~0.8029 at Round 4 to 0.8029	Rapid early convergence (+0.0017/round)
Phase II (Steady)	4-8	0.8029 to 0.8173	Steady improvement (+0.0004/round)
Phase III (Fine)	8-12	0.8173 to 0.8165	Stable fine-tuning behavior

This convergence pattern shows that most representational improvement occurred during the early training rounds, after which the global model approached a stable operating region. The minimal change after round 8 indicates that the federated generator had largely converged before the final communication round, which supports the efficiency of the selected 12-round setting. Figure 2 shows the convergence analysis of the proposed system

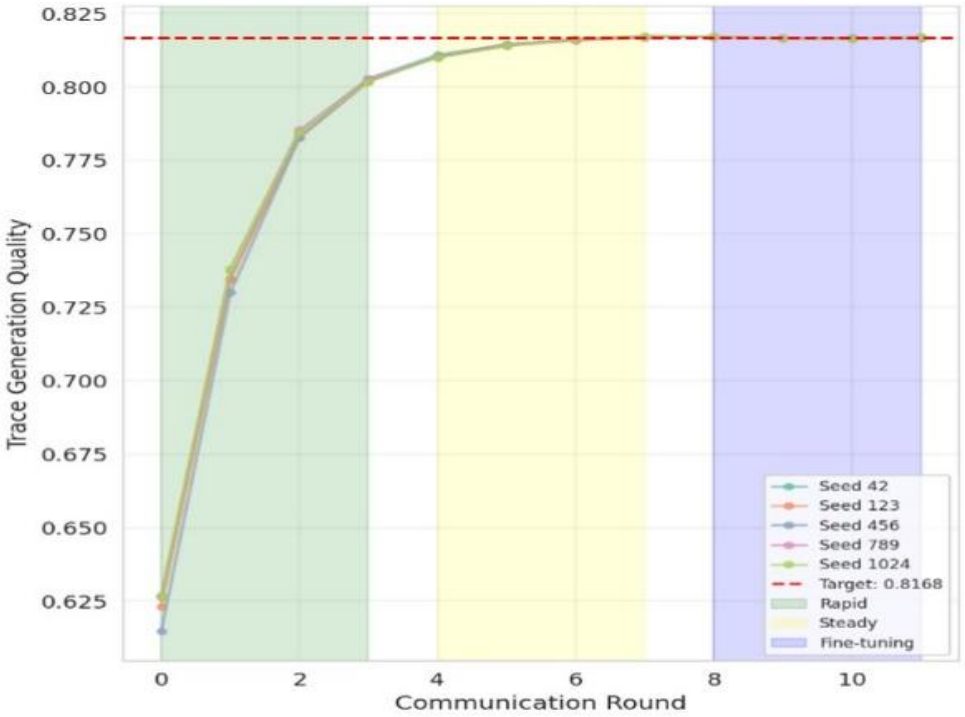


Figure 2: Convergence Analysis

6.3 Client-wise Performance

The client-wise results reveal that ChipWhisperer contributed the strongest local performance, outperforming the ASCAD baseline by 10.58%. In contrast, ASCAD and DPAV4 remained close

to each other, suggesting that the global model handled heterogeneous trace distributions reasonably well but still benefited most from the ChipWhisperer client subset. Table 8 presents the client grouping performance of the proposed framework.

Table 8: Client Group Performance (K = 20, 7-7-6)

Client Group	Clients	Trace Variance Performance	Relative Observation
ASCAD (Group A)	7	0.5623 ± 0.0017	Baseline
ChipWhisperer (Group B)	7	0.6218 ± 0.0023	+10.58% over ASCAD
DPAV4 (Group C)	6	0.5582 ± 0.0019	Slightly below ASCAD

6.4 Attack Success Rate Analysis

The privacy result is particularly important because it shows a sharp separation between attacks on original traces and attacks on generated traces. The synthetic traces reduce transferability to only 0.5%, indicating that the proposed federated generative process substantially suppresses key-dependent leakage while maintaining usable statistical quality. Table 9 and figure 3 present the attack success rate result of the proposed framework.

Table 9: Attack Success Rates

Metric	Value
Average Real-to-Real ASR	-
Average Synth-to-Real ASR	0.5% ± 0.84%
Attack Transfer Rate	0.6% (95% CI: ±0.5%)

Figure 3 present the graphical result of the attack success rate of the proposed framework

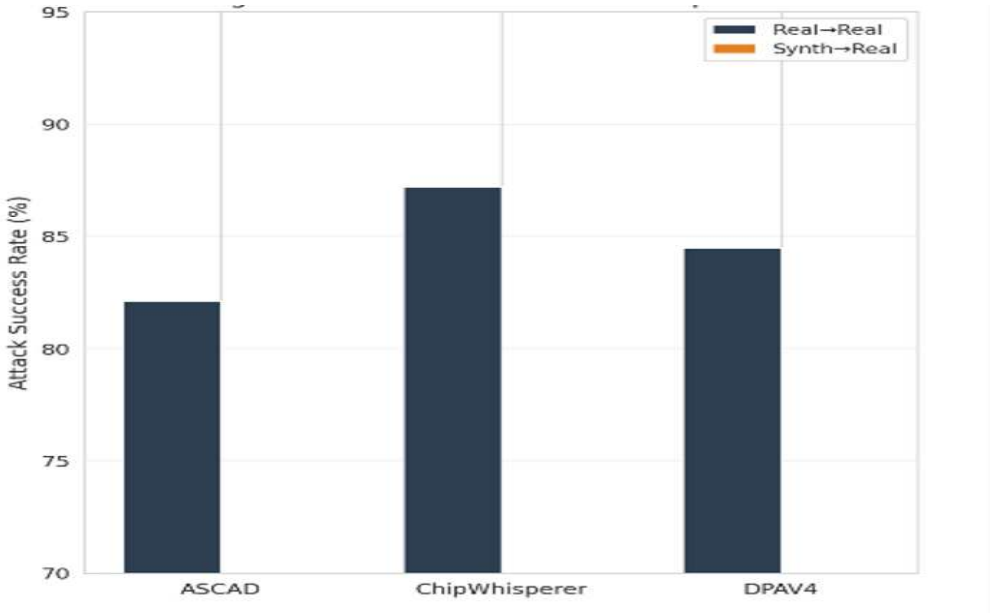


Figure 3: Attack Success Rates

7.0 Result Discussion

The results obtained for the K = 20 federated configurations demonstrate that the proposed framework achieves a good balance between synthetic trace utility and side-channel privacy preservation. Across 10 independent seeds, the model produced reasonable stable quality scores, with final round values tightly concentrated around 0.8164 to 0.8173, while the average synthetic-to-real attack success rate remained at only 0.5% ± 0.84%. This consistency indicates that the method is not sensitive to random initialization and that its privacy-preserving behavior is reproducible across repeated trials.

The convergence analysis further shows that learning progresses in three distinct phases. The model improves rapidly in the initial rounds, rises more gradually during the middle rounds, and then stabilizes during the final rounds. This suggests that most of the useful global representation is learned early, while later rounds mainly refine the generator rather than substantially changing the privacy–utility trade-off.

The comparative client analysis reveals that ChipWhisperer clients contributed the strongest local performance, exceeding the ASCAD baseline by 10.58%, whereas DPAV4 remained slightly below ASCAD. Even so, the global model maintained robust aggregate performance, which indicates that the federated aggregation process can accommodate heterogeneous side-channel distributions without collapsing into dataset-specific bias. Most importantly, the privacy evaluation confirms that the generated traces retain far less exploitable leakage than the original traces. Synthetic traces achieved an attack success rate of only 0.5% ± 0.84%, with an Attack Transfer Rate of 0.6% (95% CI: ±0.5%), indicating that the generated traces retain far less exploitable leakage than would be expected from direct use of real traces. This sharp reduction shows that the proposed system effectively disrupts attack transferability while preserving sufficient structural realism for downstream analytical use.

8.0 Conclusion

This paper presented a federated WGAN-GP framework for decentralized side-channel trace synthesis across 20 heterogeneous clients drawn from ASCAD, ChipWhisperer, and DPAV4. The proposed framework retained raw traces on client devices, used federated averaging for collaborative training, and demonstrated stable convergence over 10 independent runs with 12 communication rounds. The final statistical result for the K = 20 setting was 0.8167 ± 0.0004 in trace-variance quality, while the synthetic-to-real attack success rate fell to 0.5% ± 0.84%.

The findings show that the proposed architecture can generate side-channel traces that remain statistically realistic while strongly suppressing attack-effective leakage. ChipWhisperer clients provided the strongest local contribution, outperforming the ASCAD baseline by 10.58%, and the 7-7-6 client distribution was selected to ensure balanced representation across heterogeneous hardware platforms. This indicates that balanced client composition plays a decisive role in achieving strong privacy-utility trade-offs in heterogeneous federated side-channel settings.

Overall, the study demonstrates that federated WGAN-GP is a viable foundation for privacy-preserving synthetic trace generation and decentralized cryptographic security assessment. In the

future we would incorporate formal privacy mechanisms such as secure aggregation and differential privacy and benchmark against centralized and federated baselines under matched conditions. We will also expand evaluation with additional realism metrics such as Wasserstein distance, Maximum Mean Discrepancy, guessing entropy, and success-rate evolution.

9.0 Declaration

i. Funding Declaration

No funding was provided for the research

ii. Ethics Approval

Not Applicable

iii. Consent to Participate

Not Applicable

iv. Consent to Participate

All authors have agreed to publish the research work

v. Data Availability

The experimental data generated and analysed during this study are not publicly available because the research is part of an ongoing investigation involving security-sensitive cryptographic side-channel measurements and associated model-development activities.

REFERENCES

[1] Kaleem, M., Mushtaq, M. A., Ramay, S. A., Mahmood, A., Khan, T. A., Hussain, S. K., ... & Bhatti, H. A. (2024). Navigating Side-Channel Attacks: A Comprehensive Overview of Cryptographic System Vulnerabilities. *Journal of Computing & Biomedical Informatics*, *7*(02).

[2] Park, S., Seo, A., Cheong, M., Kim, H., Kim, J., & Son, Y. (2025). Evaluating the Vulnerability of Hiding Techniques in Cyber-Physical Systems Against Deep Learning-Based Side-Channel Attacks. *Applied Sciences*.

[3] Hameed, F., & Alkhzaimi, H. (2024). Advanced side-channel profiling attacks with deep neural networks: A hill climbing approach. *Electronics*, *13*(17), 3530.

[4] Liu, H., Tian, Y., Peng, C., & Wu, Z. (2023). Privacy-utility equilibrium data generation based on Wasserstein generative adversarial networks. *Information Sciences*, *642*, 119069.

[5] Singh, P., Pranav, P., & Dutta, S. (2025). Optimizing cryptographic protocols against side channel attacks using WGAN-GP and genetic algorithms. *Scientific Reports*, *15*(1), 2130.

[6] Schram, G., Wang, R., & Liang, K. (2022). Using autoencoders on differentially private federated learning GANs. *arXiv preprint arXiv:2206.12270*.

[7] Ghavamipour, A. R., Turkmen, F., Wang, R., & Liang, K. (2023, May). Federated synthetic data generation with stronger security guarantees. In *Proceedings of the 28th ACM Symposium on Access Control Models and Technologies* (pp. 31-42).

[8] Pereira, D., Reis, P. R., & Borges, F. (2024). Secure aggregation protocol based on dc-nets and secret sharing for decentralized federated learning. *Sensors*, *24*(4), 1299.

[9] Lu, S., Li, R., & Liu, W. (2024). FedDAA: a robust federated learning framework to protect privacy and defendagainst adversarial attack. *Frontiers of Computer Science*, *18*(2), 182307.

[10] Luo, H., & Ji, C. (2025). Federated Learning-Based Data Collaboration Method for Enhancing Edge Cloud AI System Security Using Large Language Models. *arXiv preprint arXiv:2506.18087*.

[11] Zhang, Z., Li, M., An, X., Li, Z., & Ma, Z. (2024, May). Side-Channel Analysis of Curve 25519 Based on Deep Learning. In Proceedings of the 2024 International Conference on Generative Artificial Intelligence and Information Security (pp. 144-149).

[12] Wijesinghe, A., Zhang, S., & Ding, Z. (2023). Ps-fedgan: an efficient federated learning framework based on partially shared generative adversarial networks for data privacy. *arXiv preprint arXiv:2305.11437*.

[13] Azam, S. S., Kim, T., Hosseinalipour, S., Joe-Wong, C., Bagchi, S., & Brinton, C. (2022, May). Can we generalize and distribute private
representation learning? In *International Conference on Artificial Intelligence and Statistics* (pp. 11320-11340). PMLR.

[14] Li, Z., Shao, J., Mao, Y., Wang, J. H., & Zhang, J. (2022, July). Federated learning with gan-based data synthesis for non-iid clients. In *International workshop on trustworthy federated learning* (pp. 17-32). Cham: Springer International Publishing.

[15] Javeed, D., Saeed, M. S., Adil, M., Kumar, P., & Jolfaei, A. (2024). A federated learning-based zero trust intrusion detection system for Internet of Things. *Ad Hoc Networks*, *162*, 103540.

[16] Rashid, M. M., Khan, S. U., Eusufzai, F., Redwan, M. A., Sabuj, S. R., & Elsharief, M. (2023). A federated learning-based approach for improving intrusion detection in industrial internet of things networks. *Network*, *3*(1), 158-179.

[17] Shahin, M., Hosseinzadeh, A., & Chen, F. F. (2025). A Two-Stage Hybrid Federated Learning Framework for Privacy-Preserving IoT Anomaly Detection and Classification. *IoT*, *6*(3), 48.

[18] Bouzeraib, W., Ghenai, A., & Zeghib, N. (2025). Enhancing IoT intrusion detection systems through horizontal federated learning and optimized WGAN-GP. *Ieee Access*.

[19] Kumar, G. H., Reddy, S., Saxena, S., Swamy, K. A., & Kumar, U. P. (2025). FL-DPCSA: Federated learning with differential privacy for cache side-channel attack detection in edge-based smart grids. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 101057.

[20] Bouabba, S., Zeitouni, K., Haidar, B., Agoulmine, N., & Dagdia, Z. C. (2024, June). Federated TimeGAN for privacy preserving synthetic trajectory generation. In *2024 25th IEEE International Conference on Mobile Data Management (MDM)* (pp. 301-306). IEEE.

[21] Ramachandra, P., & Vaithiyanathan, S. (2025). Fed-DPSDG-WGAN: Differentially Private Synthetic Data Generation for Loan Default Prediction via Federated Wasserstein GAN. *IEEE Access*.