FP-Fed

- Fingerprinting remains a common tracking scheme
- Typical FP uses centralized crawls, with limitations
- Use online *federated learning* to detect fingerprinters instead
- Challenges: performance, accuracy, privacy

Issues with crawls

- How do you simulate a human during crawls?
  - Crawlers often visit a limited subset of sites
  - Crawlers don’t log into sites, take real-world actions
  - Crawlers use a limited number of configurations, vantage points
  - Crawlers limited in scale

Differential Privacy

- High-level: take two *neighboring* data sets D, D’
  - Differ by one *record* or one *user’s* contribution
- P(f(D) in S) =~ P(f(D’) in S)
  - Multiplicative difference of e$^{\epsilon}$
- Centralized DP: D is an entire database (trusted server)
  - LDP: D is individual data contribution
  -

Federated Learning

- Gradient descent:
  - Evaluate current model on test set
  - Compute gradient
- Federated gradient descent:
  - Evaluate current model on *local* set
  - Compute local gradient
  - Use DP to compute *average*

Features

- API calls (instrumented browser)
  - Focus on high-entropy APIs
- API call counts
- Specific features based on API usage

Results

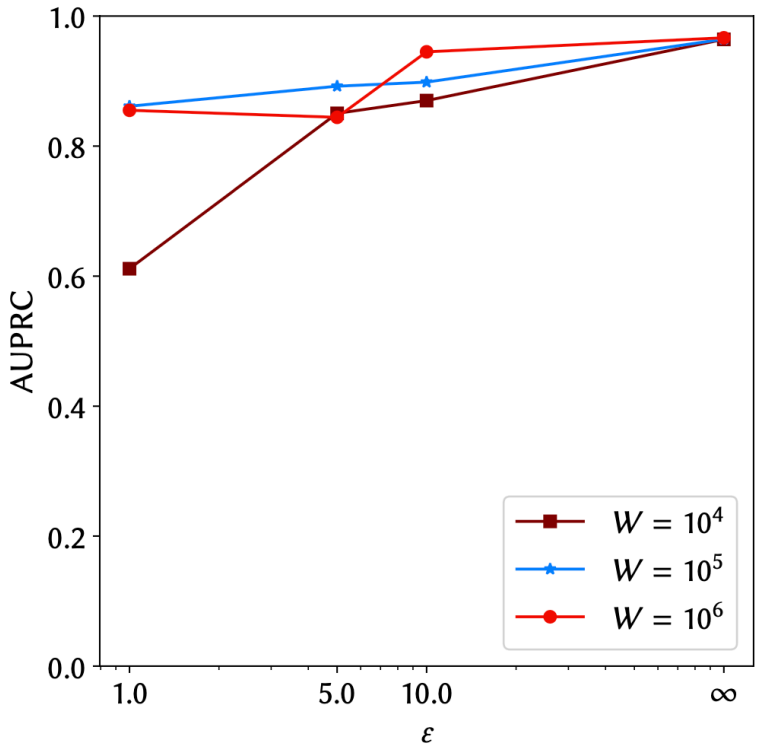

Fig. 4: Model performance (AUPRC) vs. the privacy parameter (ε) for an increasing number of participants (W).

Summary

Positive Points:

- **Innovative Approach:** Combines federated learning and differential privacy for scalable, privacy-preserving fingerprint detection.
- **Thorough Evaluation:** Tested on 18,300 websites with strong performance under various privacy settings.
- **Efficient Design:** Lightweight model with minimal features, enabling practical deployment on user devices.

Areas for Improvement:

- **Real-World Testing:** Needs large-scale deployment to validate scalability and performance with diverse user conditions.
- **Adaptability:** Must address evolving fingerprinting techniques through continuous learning or feature updates.
- **Security Risks:** Requires further protection against data poisoning and adversarial attacks.

Discussion

- Realistic?
  - Trust in server (privacy)
  - Trust in users (robustness)
  - Deployment feasibility
- Can FP scripts adapt to avoid detection?
- Are there potential incentives for participation?
- What’s the right privacy setting (epsilon)?

More discussion / takeaways?

- Anyone really like / really dislike the paper? Why or why not?
- Anything surprising?