Towards a Practical Differentially Private Collaborative Phone Blacklisting System

Daniele Ucci, Roberto Perdisci, Jaewoo Lee, Mustaque Ahamad

ACSAC'20 - December 7-11, Virtual Event

Work partially funded by the National Science Foundation (NSF) under grants:

- 1514035
- 1514052
- 1943046

CIS SAPIENZA

RESEARCH CENTER FOR CYBER INTELLIGENCE AND INFORMATION SECURITY

Phone spam and robocalls

- Successful attack vector
- Effective scam delivery tool
- Supports well-coordinating fraudulent campaigns

Federal agencies, smartphone vendors and companies have proposed different solutions, respectively:

- systems for blocking robocalls
- spam blocking functionalities
- third party mobile apps

Most spam blocking apps rely on caller ID blacklisting, but this poses *serious privacy risks* to users

Phone spam and robocalls

We aim to build a practical phone blacklisting system that leverages *differential privacy* mechanisms to collaboratively learn effective anti-spam phone blacklists while providing strong privacy guarantees.

In particular:

- we rely on a state-of-the-art *local differential privacy* (LDP) protocol
- we envision an app that will report unknown caller IDs from which the user received a phone call
- the server is able to identify, from reports, *heavy hitter caller IDs* that are highly likely associated with spamming activities, minimizing the risk of the server learning any sensitive information

Differential privacy

- Gives formal guarantees on the responses returned by arbitrary queries to a database D [1]
- Considering
  - all databases D_1 and D_2, differing on at most one tuple
  - a randomized function K, implementing a mechanism for answering queries against D_1 and D_2
  - R as the response returned by K
- Then

$$e^{-\varepsilon} \leq \underbrace{\frac{\Pr[K(D_1) = R]}{\Pr[K(D_2) = R]}}_{\substack{\text{Attacker gain by querying} \\ D_1 \text{ over } D_2}} \leq e^{\varepsilon}$$

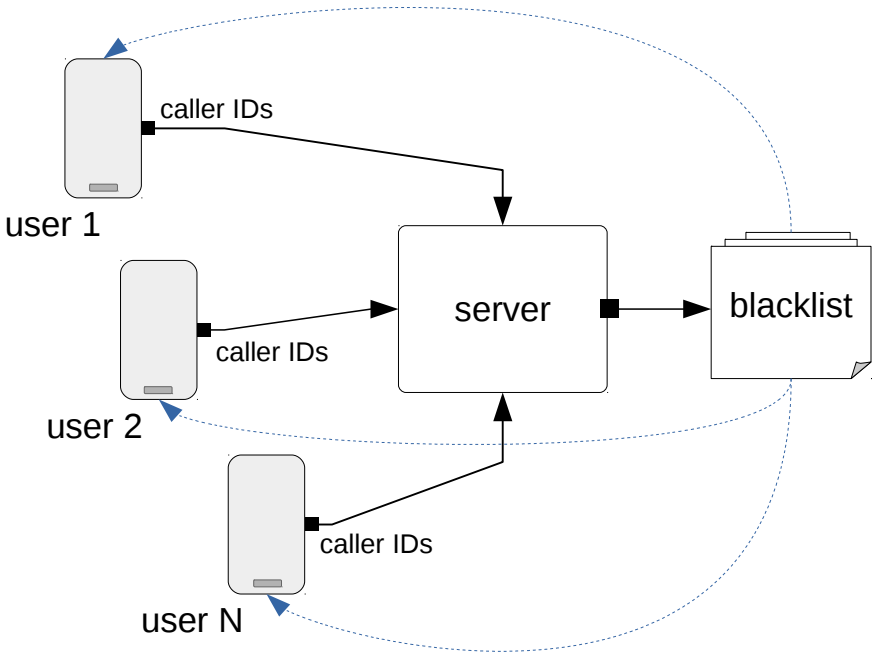
ε is a privacy parameter ranging from 0 to 1

[1] Cynthia Dwork. "Differential privacy". In: *in ICALP*. Springer, 2006, pp. 1–12.

Local differential privacy

- Even the data curator is *untrusted*
- Data sent to the curator are previously randomized by the users and satisfy ε-differential privacy
- In LDP, the privacy parameter ε is larger than in DP protocols
- In literature, ε is referred to as *privacy budget*

A privacy-preserving collaborative blacklisting system



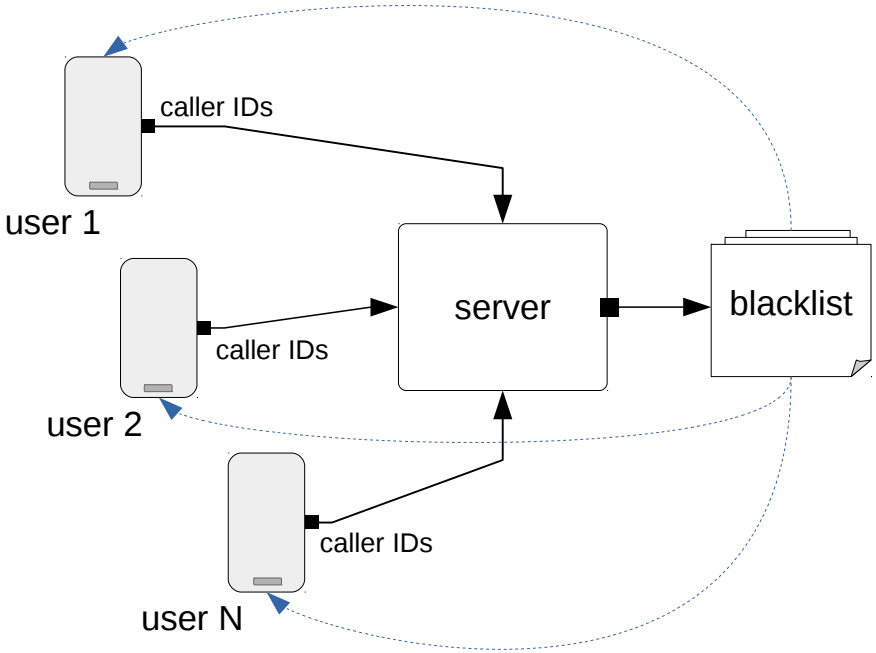
On behalf of the user, the app:

- buffers unknown caller IDs for reporting
- checks unknown caller IDs against a blacklist
- reports unknown caller IDs to the server

The server receives *reports* delivered by client apps through a LDP mechanism.

The server may be compromised –or subpoenaed– allowing an adversary to access users' reports

A privacy-preserving collaborative blacklisting system



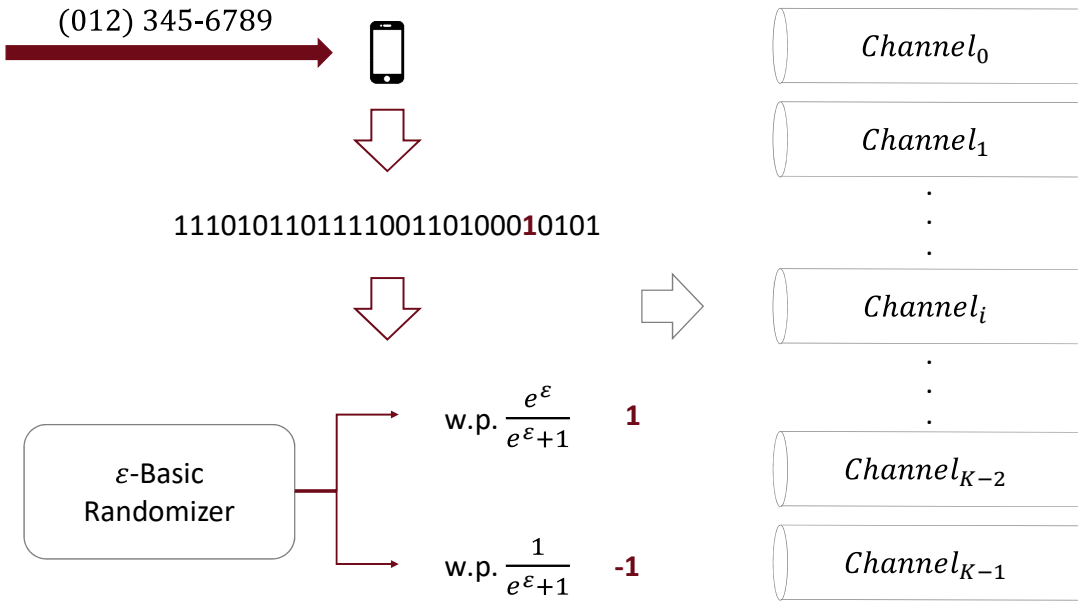
The server:

- gathers user *private reports*
- identifies spamming activities by detecting *heavy hitter* caller IDs
- redistributes newly computed blacklist to clients through the app

Heavy hitter detection is built upon a state-of-the-art protocol proposed by Bassily and Smith[2]

[2] Raef Bassily and Adam Smith. "Local, private, efficient protocols for succinct histograms". In: *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. ACM. 2015, pp. 127–135.

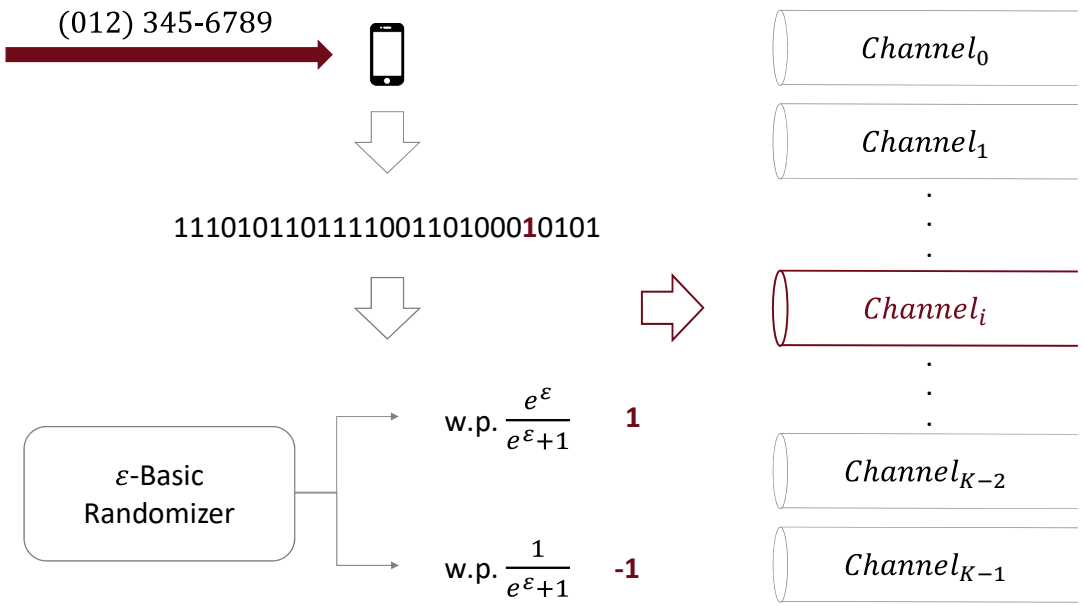
SH protocol overview



Each client app:

- collects calls received from *unknown* phone numbers
- randomly transforms one of these numbers into its binary representation b
- randomly selects one of its bits b_i
- applies an LDP algorithm to b_i

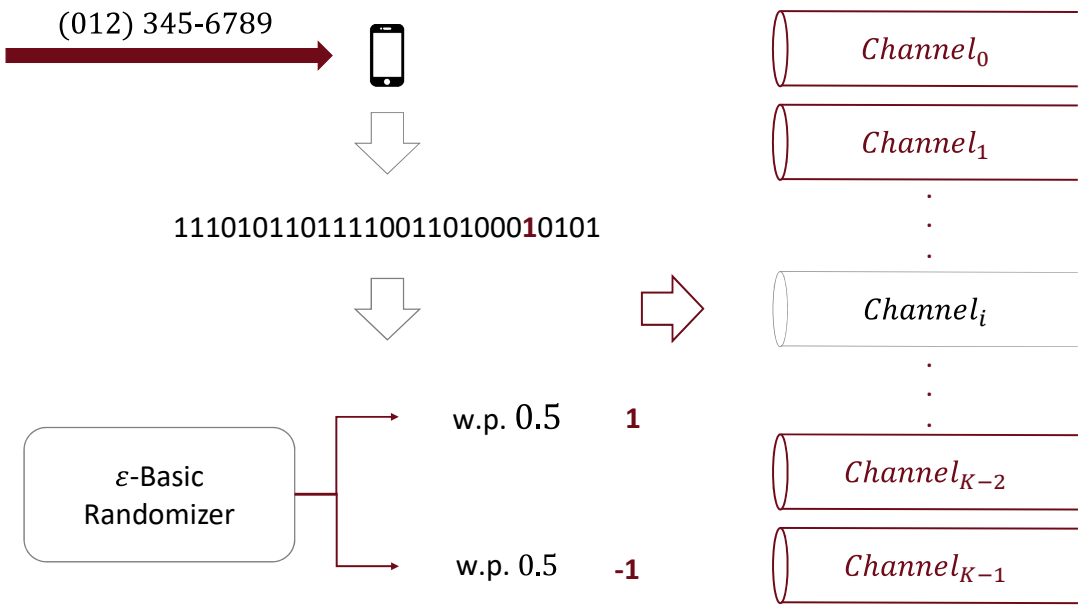
SH protocol overview



Each client app:

- for a specific *channel* i, sends to the server the output of the applied LDP algorithm

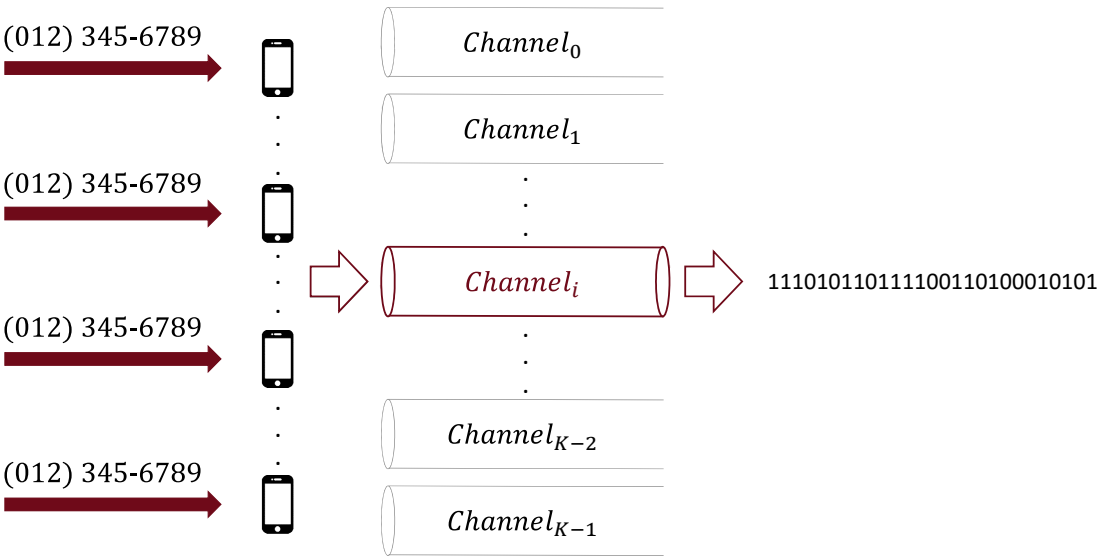
SH protocol overview



Each client app:

- for a specific *channel* i, sends to the server the output of the applied LDP algorithm
- randomly distributes -1s and 1s in all other channels

SH protocol overview



The server:

- properly reconstructs the bit-string representing a possible spammer's number s
- given the number of reports, computes an estimate of s frequency $f(s)$ through an oracle FO
- checks if the computed frequency exceeds a threshold η; if $f(s) > \eta$, then s is a heavy caller

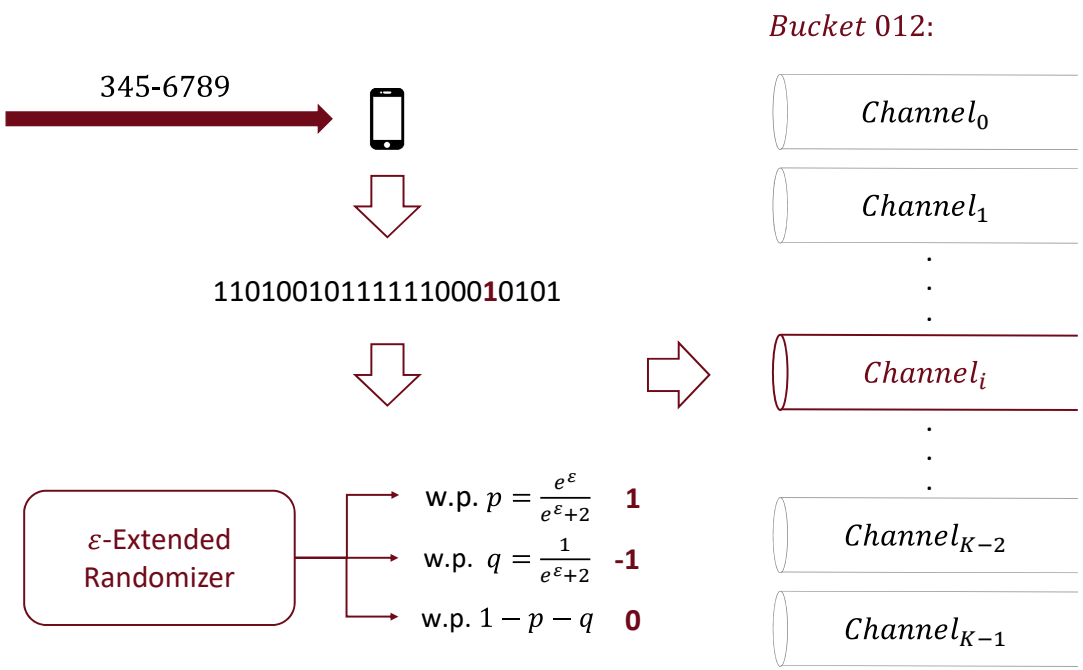
Addressed limitations of the SH protocol

- Sparsity of user reports and high variance introduced by the ϵ-Basic Randomizer potentially impedes the correct reconstruction of spam phone numbers
  - Our blacklisting system should perform well for realistic, limited population sizes (e.g., thousands of users)
  - We design a bucketization mechanism based on the phone number's area code to address the sparsity of user reports
  - We replace the original randomizer with a new ε-Extended Randomizer to reduce variance

- Complexity of the frequency oracle
  - We substitute it with a simpler and logically equivalent oracle[3]

[3] Tianhao Wang et al. "Locally Differentially Private Protocols for Frequency Estimation". In: *26th USENIX Security Symposium (USENIX Security 17)*. 2017.

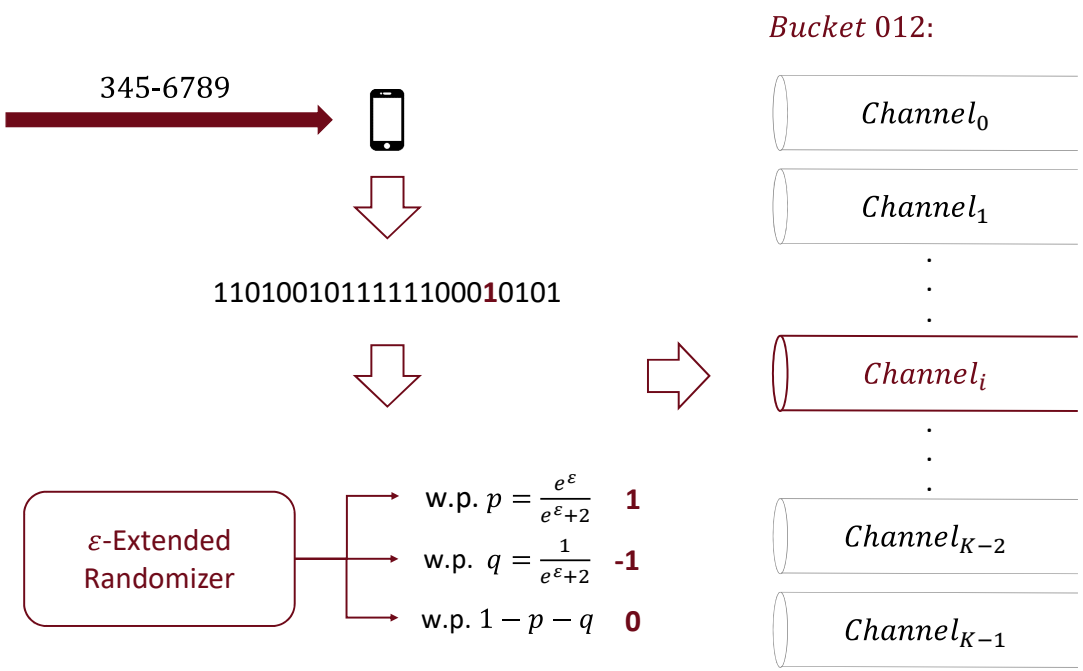
Modified SH protocol overview



In each client app:

- Communication channels are instantiated per bucket
- LDP algorithm is applied just to the *local* number collected from unknown calls
- The ε-Extended Randomizer satisfies ε-LDP

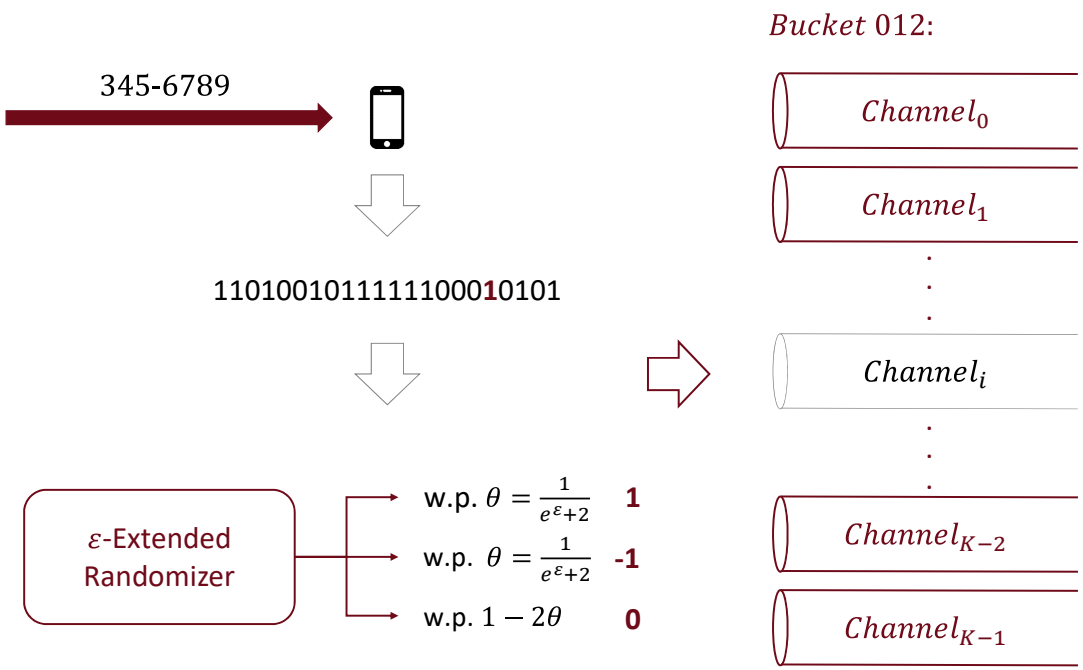
Modified SH protocol overview



As for the SH protocol:

- for a specific *channel* i, the app sends to the server the output of the ε-Extended Randomizer depending on p and q

Modified SH protocol overview



As for the SH protocol:

- for a specific *channel* i, the app sends to the server the output of the ε-Extended Randomizer depending on p and q
- for all other channels, the app sends to the server the output of the ε-Extended Randomizer depending on θ

Evaluation

- We evaluated our LDP-based system on real-world user reported call records collected by the U.S. Federal Trade Commission:
  - $471,460$ complaints between Feb. 17th 2016 and Mar. 17th 2016, for a total of 29 days
- In all our experiments:
  - we use area code bucketization, comparing results obtained using the Basic Randomizer to results obtained using our Extended Randomizer
  - we allocate two different privacy budgets to run the heavy hitter detection and the frequency estimation protocol (respectively, $\varepsilon_{\mathsf{HH}} \in \{12, 8.8, 7, 5.6, 4.4\}$ and $\varepsilon_{\mathsf{OLH}} = 3$)
  - we set a parameter to enable the heavy hitter detection phase only for those buckets containing more than a minimum number, τ, of complaints (i.e., $\tau \in \{143, 151, 161, 174, 195\}$)

Each experimental evaluation with a given $\varepsilon_{\mathsf{HH}}$ and $\varepsilon_{\mathsf{OLH}} = 3$ was repeated 10 times and the results averaged

Evaluation

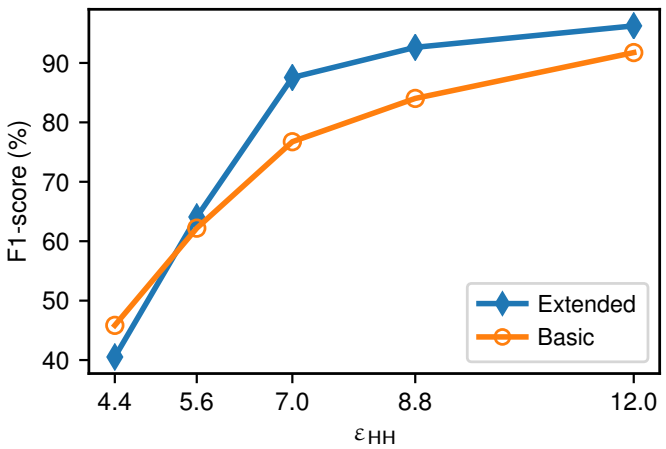
We define:

- **True Heavy Hitters** (THHs) as the set of phone numbers reported from a number of users greater than τ and whose estimated frequencies are greater than τ as well
- **False Heavy Hitters** (FHHs) as the set of phone numbers reported from a number of users lower than τ and whose estimated frequencies are greater than τ
- **Undetected Heavy Hitters** (UHHs) as the set of phone numbers reported from a number of users greater than τ and whose estimated frequencies are lower than τ

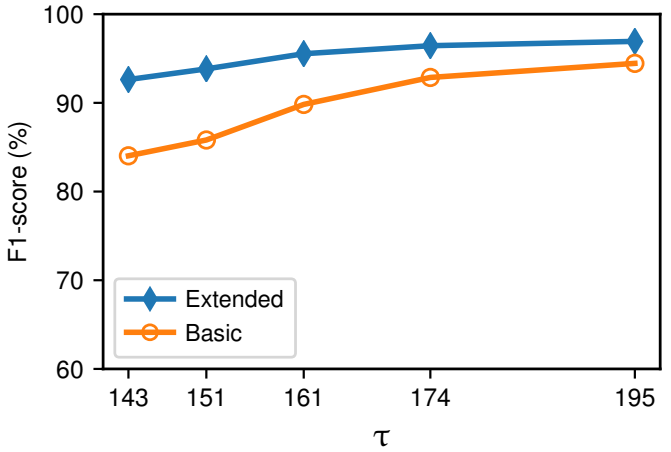
Then:

- **Recall**: $R = THHs/(THHs + UHHs)$
- **Precision**: $P = THHs/(THHs + FHHs)$
- **F1-score**: $F_1 = 2 * (P * R)/(P + R)$

Heavy hitter detection accuracy

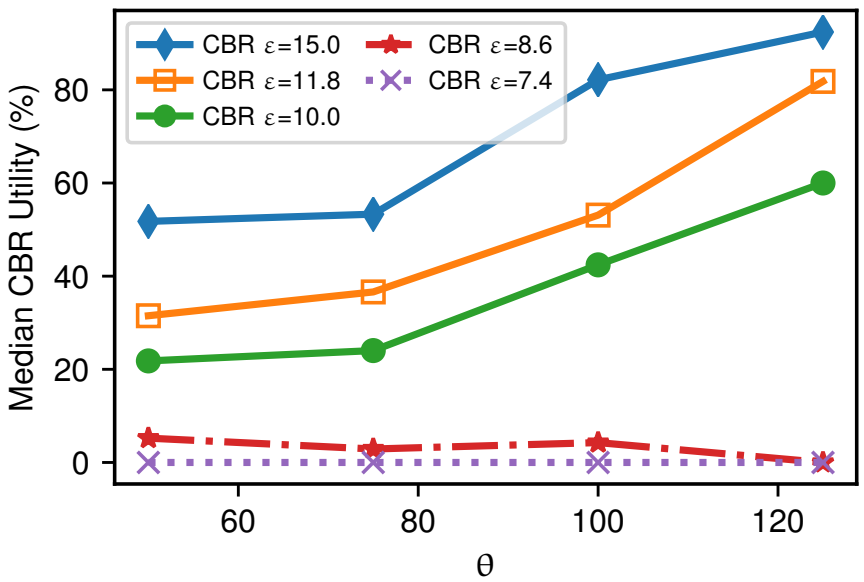


F1-score with parameters: $\varepsilon_{\text{OLH}} = 3$ and $\tau = 143$.



F1-score with parameters: $\varepsilon_{\text{HH}} = 8.8$ and $\varepsilon_{\text{OLH}} = 3$.

Blacklist utility

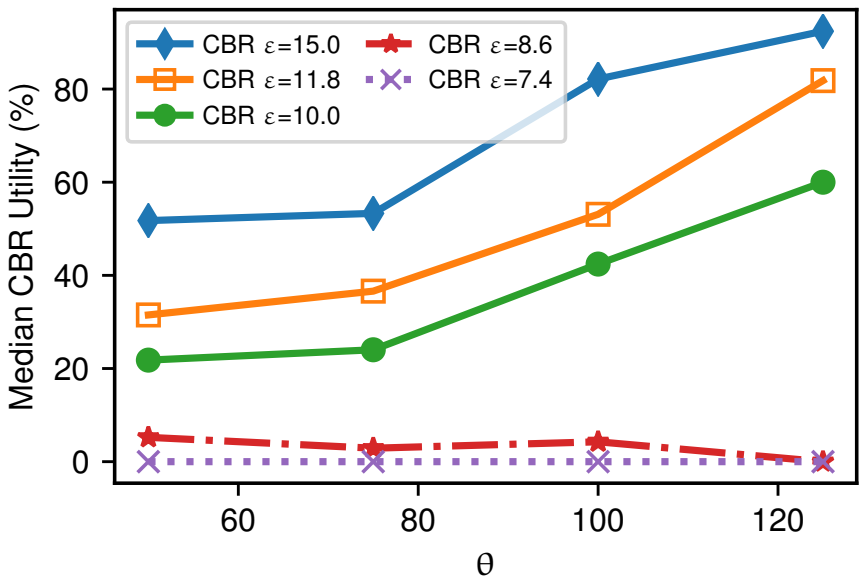


CBR: percentage of calls blocked compared to the baseline.

We compare how a blacklist $\mathbb{B}$ learned over heavy hitters detected using our protocol would fare compared to when no privacy is preserved:

- we use a sliding window mechanism, whereby $\mathbb{B}$ is updated daily over the past week
- we set the same fixed heavy hitter detection threshold $\theta = \tau$ and $\varepsilon_{\text{OLH}} = 3$ for both approaches

Blacklist utility



CBR: percentage of calls blocked compared to the baseline.

We define the Call Blocking Rate (CBR) as:

$$CBR = \frac{N_{\mathbb{B}}}{N_{tot}}$$

- $N_{\mathbb{B}}$ is the number of complaints that would have been blocked by $\mathbb{B}$
- N_{tot} the total number of complaints received up to the previous week

The baseline is computed as the CBR* that can be achieved without applying any privacy-preserving mechanism

Conclusion

We proposed a novel collaborative detection system that learns a list of spam-related phone numbers. Our system

- makes use of local differential privacy to provide clear privacy guarantees
- has been evaluated on real-world user-reported call records collected by the FTC
- is able to learn a phone blacklist in a privacy-preserving way using a reasonable overall privacy budget, maintaining the utility of the learned blacklist