

Private Frequency Estimation Via Residue Number Systems

Héber Hwang Arcolezi

Inria, Grenoble, France
heber.hwang-arcolezi@inria.fr

Abstract

We present Modular Subset Selection (MSS), a new algorithm for locally differentially private (LDP) frequency estimation. Given a universe of size k and n users, our ε -LDP mechanism encodes each input via a Residue Number System (RNS) over ℓ pairwise-coprime moduli $m_0, \dots, m_{\ell-1}$, and reports a randomly chosen index $j \in [\ell]$ along with the perturbed residue using the statistically optimal Subset Selection (SS). This design reduces the user communication cost from $\Theta(\omega \log_2(k/\omega))$ bits required by standard SS (with $\omega \approx k/(e^\varepsilon + 1)$) down to $\lceil \log_2 \ell \rceil + \lceil \log_2 m_j \rceil$ bits, where $m_j < k$. Server-side decoding runs in $\Theta(n + rk\ell)$ time, where r is the number of LSMR iterations. In practice, with well-conditioned moduli (*i.e.*, constant r and $\ell = \Theta(\log k)$), this becomes $\Theta(n + k \log k)$. We prove that MSS achieves worst-case MSE within a constant factor of state-of-the-art protocols such as SS and Projective Geometry Response (PGR), while avoiding the algebraic prerequisites and dynamic-programming decoder required by PGR. Empirically, MSS matches the estimation accuracy of SS, PGR, and RAPPOR across realistic (k, ε) settings, while offering faster decoding than PGR and shorter user messages than SS. Lastly, by sampling from multiple moduli and reporting only a single perturbed residue, MSS achieves the lowest reconstruction-attack success rate among all evaluated LDP protocols.

Code — <https://github.com/hharcolezi/private-frequency-oracle-rns>

1 Introduction

Today’s *federated* applications span billions of devices, such as keyboard prediction by Apple (Apple Differential Privacy Team 2017) and Gboard (Sun et al. 2024), and telemetry systems in Google Chrome (Erlingsson, Pihur, and Korolova 2014) and Microsoft operating systems (Ding, Kulkarni, and Yekhanin 2017), all of which must learn from data that never leaves the user’s device in raw form. The prevailing formalism is the *local model* of differential privacy (LDP) (Kasiviswanathan et al. 2011; Duchi, Jordan, and Wainwright 2013): each user applies a randomizer \mathcal{M} to their datum $x \in \mathcal{X}$ and sends only the noisy message $Y = \mathcal{M}(x)$ to an

untrusted aggregator. A mechanism $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ is ε -LDP if for every measurable $S \subseteq \mathcal{Y}$ and every $x, x' \in \mathcal{X}$,

$$\Pr[\mathcal{M}(x) \in S] \leq e^\varepsilon \Pr[\mathcal{M}(x') \in S].$$

Under LDP *no single report* can distinguish two inputs by a factor larger than e^ε . Once the locally obfuscated reports arrive, the server aims to perform global tasks such as statistical estimation or model training. Four main factors determine the practicality of any local-DP protocol:

- (i) **Utility**: the accuracy with which the server can complete its task.
- (ii) **Communication**: the number of bits each user must transmit per report.
- (iii) **Server runtime**: the time and memory required for server-side decoding.
- (iv) **Attackability**: the probability that an adversary correctly recovers an individual’s input from a single report.

Together, these four dimensions form a *multi-constraint regime*: in large-scale telemetry and federated analytics, client bandwidth, server compute, statistical accuracy, and privacy risk may each become the dominant constraint depending on the deployment.

Problem Statement. This work addresses this *multi-constraint* challenge when designing efficient mechanisms for *federated-analytics* deployments that require locally differentially private *frequency estimation* over a finite domain $[k] = \{0, \dots, k-1\}$. In this setting, each user holds a private input x_i and the goal is for an untrusted server to recover an accurate estimate of the population histogram $\mathbf{f} \in \mathbb{R}^k$, where $f_v = \frac{\#\{i: x_i=v\}}{n}$. After collecting n randomized reports $\{Y_i\}_{i=1}^n$, the server computes an estimate $\hat{\mathbf{f}}$ aiming to minimize its distance from \mathbf{f} under some norm $\|\mathbf{f} - \hat{\mathbf{f}}\|$. In line with prior literature (Feldman et al. 2022; Kairouz, Bonawitz, and Ramage 2016; Wang et al. 2017; Acharya, Sun, and Zhang 2019), we quantify estimation error using the expected ℓ_2 norm, and focus on the mean squared error (MSE) metric: $\text{MSE} = \frac{1}{k} \mathbb{E}[\|\hat{\mathbf{f}} - \mathbf{f}\|_2^2]$.

In addition to utility, another fundamental concern in the local DP model is *attackability*: the ability of a Bayesian adversary to reconstruct a user’s true input x from a single obfuscated message Y (Emre Gursoy et al. 2022; Arcolezi

and Gambs 2025). This threat, commonly referred to as a *Data Reconstruction Attack (DRA)* in the AI and ML communities (Geiping et al. 2020; Hayes, Balle, and Mahloujifar 2023; Guerra-Balboa, Sauer, and Strufe 2024), is quantified as the probability that an adversary with full knowledge of the protocol and prior distribution correctly guesses x given Y . Protocols that minimize estimation error while keeping reconstruction rate low provide stronger privacy in practice.

Related work. Table 1 summarizes the trade-offs across utility, communication, computation, and attackability of state-of-the-art LDP frequency estimation protocols. Classical `RandomizedResponse` (Warner 1965; Kairouz, Bonawitz, and Ramage 2016) minimizes per-user bandwidth (one $\lceil \log_2 k \rceil$ -bit symbol) but suffers an $\Theta(k/e^\epsilon)$ gap to the information-theoretic MSE bound and yields the highest single-message reconstruction success rate. The `SubsetSelection` (SS) mechanism (Wang et al. 2016) attains the optimal worst-case MSE by returning a random subset containing the true value. However, this comes with $\Theta(\omega \log_2(k/\omega))$ bits of communication per user and high server cost. Bit-vector schemes like `RAPPOR` (Erlingsson, Pihur, and Korolova 2014) and `OptimalUnaryEncoding` (OUE) (Wang et al. 2017) reach the same optimal bound by perturbing k -length binary encodings, but this increases both message size ($O(k)$) and server time ($O(nk)$). Most recently, the coding-based `ProjectiveGeometryResponse` protocol (Feldman et al. 2022) demonstrates that algebraic structure can enable near-optimal utility with reduced communication cost as $\lceil \log_2 k \rceil$. However, its deployment remains nontrivial: PGR requires the domain size to match a projective geometry constraint, relies on finite fields of size near e^ϵ , and uses dynamic programming for decoding.

Our Contributions. We propose `ModularSubsetSelection` (MSS), a novel single-message ϵ -LDP protocol that tackles the accuracy-bandwidth-computation-attackability four-way trade-off through a modular “*divide & conquer*” design based on *Residue Number System* (RNS) (Szabó and Tanaka 1967). Each input $x \in [k]$ is first mapped to a short RNS vector $(x \bmod m_0, \dots, x \bmod m_{\ell-1})$ using a set of pairwise-coprime moduli $(m_0, \dots, m_{\ell-1})$; by the Chinese Remainder Theorem (CRT), $\prod_{j=0}^{\ell-1} m_j \geq k$ ensures that the mapping is injective over $[k]$. Instead of transmitting the full residue vector, each user samples *one* block index uniformly at random and perturbs its coordinate with `SubsetSelection` at privacy level ϵ . This “*divide*” step reduces the message alphabet from k to at most $\max_j m_j < k$, so the report fits into $\lceil \log_2 \ell \rceil + \lceil \log_2 m_j \rceil$ bits. *On the user side, this requires nontrivial CRT-based design choices to maintain injectivity, full rank, and a favorable ℓ - m_j trade-off.*

On the server side, MSS “*conquers*” the estimation error via a variance-weighted least-squares solver on a sparse design matrix A_w . For well-conditioned moduli, the total decoding cost is $O(n + k\ell)$ (empirically $O(n + k \log k)$). Theoretically, the worst-case mean-squared error satisfies $\text{MSE}_{\text{MSS}} \leq \kappa \text{MSE}_{\text{SS}}$, $\kappa = \text{cond}(A_w)$, and our modulus search keeps $\kappa \leq 10$. *The server-side challenges include controlling κ to guarantee low MSE, designing a*

variance-optimal unbiased decoder, and selecting moduli that balance accuracy and computational cost. In practice, the empirical ratio $\text{MSE}_{\text{MSS}}/\text{MSE}_{\text{SS}}$ never exceeded $\kappa \approx 1.3$ across all (k, ϵ) we tested, indicating only a small constant-factor overhead. Lastly, MSS reduces a Bayesian attacker’s single-report reconstruction success by increasing uncertainty over the domain, outperforming `SubsetSelection` and `RandomizedResponse` in our experiments.

Comparison with `ProjectiveGeometryResponse`.

PGR (Feldman et al. 2022) attains the information-theoretic variance bound of SS but at the cost of finite-field arithmetic, rigid domain constraints, and a dynamic-programming decoder with $O(n + k e^\epsilon \log k)$ states. MSS eliminates these algebraic prerequisites: it accepts *arbitrary* k and ϵ , relies solely on integer arithmetic, and replaces combinatorial decoding by a single sparse least-squares solve amenable to out-of-core and parallel settings. Empirically, MSS matches or approximates the utility loss of PGR (and SS) while requiring much less server-side runtime; moreover, the tunable parameter ℓ lets practitioners navigate the full communication-accuracy spectrum, a flexibility unavailable in PGR. Therefore, MSS offers a lower-complexity, more adaptable alternative without sacrificing practical accuracy.

2 Preliminaries

Our MSS combines ideas from number theory with tools from linear algebra. We review the background below.

Definition 1 (Residue Number System (RNS) (Szabó and Tanaka 1967)). *Let $\mathcal{X} = \{0, \dots, k - 1\}$ be the finite input domain. Given a set of pairwise-coprime integers $\{m_0, \dots, m_{\ell-1}\}$, called moduli, the Residue Number System represents each $x \in \mathcal{X}$ by its residues modulo each m_j :*

$$\mathbf{r}(x) = (x \bmod m_0, \dots, x \bmod m_{\ell-1}).$$

By the Chinese Remainder Theorem, if $\prod_{j=0}^{\ell-1} m_j \geq k$, this representation is injective and fully encodes the domain.

Weighted least-squares estimators. Consider noisy linear measurements $y = Ax + \epsilon$, where $A \in \mathbb{R}^{M \times k}$ is a design matrix and ϵ has row-wise variances $(w_1^{-1}, \dots, w_M^{-1})$. The *generalised least-squares* (GLS) estimator solves

$$\hat{x} = \arg \min_z \|W^{1/2}(Az - y)\|_2^2 = (A^\top W A + \lambda I)^{-1} A^\top W y,$$

with weight matrix $W = \text{diag}(w_1, \dots, w_M)$ and optional ridge parameter $\lambda \geq 0$ (Hastie et al. 2009).

Spectral condition number. For any real matrix B let $\sigma_{\max}(B)$ and $\sigma_{\min}(B)$ denote its largest and smallest singular values. The *spectral condition number* is

$$\kappa = \text{cond}(B) = \frac{\sigma_{\max}(B)}{\sigma_{\min}(B)} \in [1, \infty).$$

Smaller values imply greater numerical stability. In our analysis, we write $\kappa = \text{cond}(A_w)$ for the weighted design matrix $A_w = W^{1/2}A$.

LDP frequency-oracle	Communication (bits)	MSE (worst-case)	Server decoding time	Attackability (DRA)
RandomizedResponse (GRR) (Kairouz, Bonawitz, and Ramage 2016)	$\lceil \log_2 k \rceil$	$\frac{e^\varepsilon + k - 2}{n(e^\varepsilon - 1)^2}$	$O(n + k)$	$\frac{e^\varepsilon}{e^\varepsilon + k - 1}$
RAPPOR (Erlingsson, Pihur, and Korolova 2014; Wang et al. 2017)	k	$\frac{4e^\varepsilon}{n(e^\varepsilon - 1)^2}$	$O(nk)$	$\frac{1}{k} \left[e^{\varepsilon/2} - \frac{e^{(k-1)\varepsilon/2} (e^{\varepsilon/2} - 1)}{(e^{\varepsilon/2} + 1)^{k-1}} \right]$
SubsetSelection (SS) (Wang et al. 2016)	$\lceil \log_2 \binom{k}{\omega} \rceil$	$\frac{4e^\varepsilon}{n(e^\varepsilon - 1)^2}$	$O(n\omega + k)$	$\frac{e^\varepsilon}{\omega e^\varepsilon + k - \omega}$
ProjectiveGeometryResponse (PGR) (Feldman et al. 2022)	$\lceil \log_2 k \rceil$	$\frac{4e^\varepsilon}{n(e^\varepsilon - 1)^2}$	$O(n + k e^\varepsilon \log k)$	$\frac{e^\varepsilon}{K + (e^\varepsilon - 1)c_{\text{set}}}$
ModularSubsetSelection (this work)	$\lceil \log_2 \ell \rceil + \lceil \sum_{j=0}^{\ell-1} \lceil \log_2 \binom{m_j}{\omega_j} \rceil \rceil$	$\frac{4K e^\varepsilon}{n(e^\varepsilon - 1)^2}$	$O(n + k\ell + \sum_{j=0}^{\ell-1} m_j)$	$\frac{1}{\ell k} \sum_{j=0}^{\ell-1} \frac{m_j \cdot e^\varepsilon}{\omega_j \cdot e^\varepsilon + m_j - \omega_j}$

Table 1: Comparison of single-message LDP frequency-estimation schemes. Communication is the number of bits per user. MSE is the worst-case mean-squared error of the unbiased estimator; server time is leading-order in users n and domain size k . DRA is the Bayesian single-message attacker success rate. $\omega = \lfloor k/(e^\varepsilon + 1) \rfloor$ is the SS subset size, and ω_j is the analogous size for modulus m_j in MSS. For PGR, the DRA expression shown applies when the domain equals the natural projective size $K = (q^t - 1)/(q - 1)$; the exact DRA for truncated domains ($k < K$) is provided in Appendix C of (Arcolezi 2025).

Iterative solution. When A is large and sparse (for large domain sizes k), we solve the GLS normal equations with the Lanczos-based LSMR algorithm (Fong and Saunders 2011), whose cost is $O(r \text{nnz}(A))$ where r is the iteration count to convergence and “nnz” is a shorthand for the number of non-zero entries in a matrix.

3 Modular Subset Selection

Following the *divide & conquer* view from Section 1, Section 3.1 covers the user-side (*divide*) mechanism, and Section 3.2 the server-side (*conquer*) estimation.

3.1 User-Side (“Divide”) Obfuscation

ModularSubsetSelection is a single-message ε -LDP mechanism for frequency estimation over the domain $\mathcal{X} = \{0, \dots, k-1\}$. Building on the residue number system (Definition 1), each input x is represented by its ℓ residues modulo a set of pairwise-coprime moduli. Rather than perturbing all residues with a split privacy budget, MSS samples and reports only a *single* coordinate $J \in [\ell]$, using the full privacy budget ε for that coordinate. This modular sampling aligns with established LDP approaches for multidimensional data (Wang et al. 2017; Arcolezi et al. 2023) and yields strong privacy, low communication cost, and efficient server-side recovery.

Concretely, each user holding a private value x proceeds by selecting one modulus m_J uniformly at random, computing the residue $r = x \bmod m_J$, and applying SubsetSelection with privacy level ε over the domain $[m_J]$. The resulting report consists of the block index J and a noisy subset $Z \subseteq [m_J]$ of fixed size ω_J . The full procedure is given in Algorithm 1.

Theorem 1 (Privacy of MSS). *ModularSubsetSelection in Algorithm 1 satisfies ε -local differential privacy.*

Proof. Fix any $x, x' \in \mathcal{X}$ and any possible output (j, Z) . The output of ModularSubsetSelection consists of two components: (i) a uniformly sampled block index $J \in [\ell]$, and (ii) a perturbed residue set $Z \subseteq [m_j]$ of fixed size ω_j generated by SubsetSelection.

Algorithm 1 USERSIDEMSS($x, \mathbf{m}, \varepsilon$)

Require: Private input $x \in \mathcal{X}$, moduli \mathbf{m} , privacy budget ε
Ensure: Noisy report (J, Z)

- 1: $\ell \leftarrow |\mathbf{m}|$
- 2: Draw $J \sim \text{Uniform}([\ell])$ \triangleright Sample modulus index
- 3: Set $p_J \leftarrow \frac{\omega_J e^\varepsilon}{\omega_J e^\varepsilon + m_J - \omega_J}$, where $\omega_j = \lfloor \frac{m_j}{e^\varepsilon + 1} \rfloor$
- 4: Compute $r \leftarrow x \bmod m_J$
- 5: Draw $\zeta \sim \text{Uniform}([0, 1])$
- 6: **if** $\zeta < p_J$ **then**
- 7: $Z \leftarrow \{r\} \cup$ random sample of $(\omega_J - 1)$ elements from $[m_J] \setminus \{r\}$
- 8: **else**
- 9: $Z \leftarrow$ random sample of ω_J elements from $[m_J] \setminus \{r\}$
- 10: **end if**
- 11: **return** (J, Z)

By construction,

$$\Pr[\text{MSS}(x) = (j, Z)] = \Pr[J = j] \cdot \Pr[Z \mid J = j, x].$$

Since J is independent of x and uniform over $[\ell]$, it contributes no privacy loss. It suffices to show that for fixed j , the randomizer SS_{m_j} applied to $x \bmod m_j$ satisfies ε -LDP.

Let $r = x \bmod m_j$ and $r' = x' \bmod m_j$. From the SS mechanism definition (Wang et al. 2016), we have:

$$\frac{\Pr[Z \mid r]}{\Pr[Z \mid r']} \leq e^\varepsilon, \quad \forall Z \subseteq [m_j], |Z| = \omega_j.$$

Hence,

$$\frac{\Pr[\text{MSS}(x) = (j, Z)]}{\Pr[\text{MSS}(x') = (j, Z)]} = \frac{1/\ell \cdot \Pr[Z \mid r]}{1/\ell \cdot \Pr[Z \mid r']} \leq 1 \cdot e^\varepsilon = e^\varepsilon.$$

Finally, since post-processing does not affect privacy, ModularSubsetSelection satisfies ε -LDP. \square

3.2 Server-Side (“Conquer”) Estimation

Upon receiving the users’ reports $y = (J, Z)$, the server’s goal is to estimate the empirical distribution $\mathbf{f} \in \mathbb{R}^k$ over $[k]$. This is done by first debiasing the noisy SS reports, forming a weighted design matrix that leverages the CRT structure, and then solving a regularized least-squares system.

Design Matrix For each block $j \in [\ell]$, define the mapping matrix $A_j \in \{0, 1\}^{m_j \times k}$ such that

$$A_j[r, x] = \mathbf{1}\{x \bmod m_j = r\}.$$

Each row of A_j encodes the indicator vector of domain values that map to residue r under modulus m_j . Stacking all A_j vertically produces the full design matrix:

$$A = \begin{bmatrix} A_0 \\ \vdots \\ A_{\ell-1} \end{bmatrix} \in \{0, 1\}^{T \times k}, \quad T = \sum_{j=0}^{\ell-1} m_j.$$

Variance-Optimal Row Weights To reflect the per-block variance from the SubsetSelection mechanism, we apply optimal variance weights to each row. Let $p_j = \frac{\omega_j e^\varepsilon}{\omega_j e^\varepsilon + m_j - \omega_j}$ and $q_j = \frac{\omega_j e^\varepsilon (\omega_j - 1) + (m_j - \omega_j) \omega_j}{(m_j - 1)(\omega_j e^\varepsilon + m_j - \omega_j)}$ denote the true and false inclusion probabilities for block j . The marginal probability that a random residue appears in J is

$$\pi_j = q_j + \frac{p_j - q_j}{m_j},$$

and for n_j reports using block j , the variance of each SS estimator coordinate is

$$\sigma_j^2 = \frac{\pi_j(1 - \pi_j)}{n_j(p_j - q_j)^2}.$$

Following generalized least squares, we define the square-root weight vector $\mathbf{w}^{1/2} \in \mathbb{R}^T$ such that each entry corresponding to block j is repeated m_j times and equals

$$\sqrt{w_j} = \frac{p_j - q_j}{\sqrt{\pi_j(1 - \pi_j)/n_j}}.$$

Let the diagonal scaling matrix:

$$W^{1/2} = \text{diag}(\underbrace{\sqrt{w_0}, \dots, \sqrt{w_0}}_{m_0}, \dots, \underbrace{\sqrt{w_{\ell-1}}, \dots, \sqrt{w_{\ell-1}}}_{m_{\ell-1}}).$$

The weighted design matrix is then:

$$A_w = W^{1/2} A.$$

Observation Vector Construction For each block j , let $c_j \in \mathbb{R}^{m_j}$ be the vector of counts, where $c_j[a]$ is the number of times residue $a \in [m_j]$ appeared in block j . Let $n_j = \sum_a c_j[a]$, define the empirical probability vector $\bar{y}_j = c_j/n_j$. Following the unbiased SS estimator (Wang et al. 2016), the debiased per-residue estimate is

$$\hat{s}_j = \frac{\bar{y}_j - q_j}{p_j - q_j},$$

and each coordinate of \hat{s}_j has variance σ_j^2 from above.

Stacking all blocks yields

$$\mathbf{s} = \begin{bmatrix} \hat{s}_0 \\ \vdots \\ \hat{s}_{\ell-1} \end{bmatrix}.$$

Since each coordinate has variance σ_j^2 , we apply the standard GLS reweighting, *i.e.*, scaling each entry by the inverse of its noise standard deviation, to obtain the weighted observations

$$\tilde{\mathbf{s}} = \mathbf{w}^{1/2} \odot \mathbf{s}.$$

Least Squares Estimation To estimate the raw frequency vector $\hat{\mathbf{f}} \in \mathbb{R}^k$, we solve:

$$\hat{\mathbf{f}} = \arg \min_{\mathbf{z} \in \mathbb{R}^k} \|A_w \mathbf{z} - \tilde{\mathbf{s}}\|_2^2 + \lambda \|\mathbf{z}\|_2^2,$$

where $\lambda > 0$ is a small ridge regularization parameter (*e.g.*, $1/\varepsilon^2$) introduced for numerical stability. In practice, we solve this system using the LSMR algorithm (Fong and Saunders 2011), a Krylov-subspace method that efficiently handles large sparse matrices and avoids explicit inversion. When $\lambda = 0$ and A_w has full column rank, the solution is:

$$\hat{\mathbf{f}} = (A_w^\top A_w + \lambda I)^{-1} A_w^\top \tilde{\mathbf{s}}, \quad (1)$$

but the computation is performed iteratively without forming dense matrices. This estimator is unbiased in expectation (see Theorem 2) as well as asymptotically (see Corollary 1), and its analytical variance is derived in Section 4.3.

Unbiasedness Analysis An important property of any frequency oracle is whether its estimates are unbiased. For our MSS estimator, we now show that it satisfies exact unbiasedness in the unregularized case ($\lambda = 0$), and is asymptotically unbiased when a small ridge regularization is applied.

Theorem 2 (Exact unbiasedness, $\lambda = 0$). *Let $\mathbf{f} \in \mathbb{R}^k$ be the true input histogram, and suppose the weighted design matrix $A_w \in \mathbb{R}^{T \times k}$ has full column rank. Then the least-squares estimator*

$$\hat{\mathbf{f}} = (A_w^\top A_w)^{-1} A_w^\top \tilde{\mathbf{s}}$$

satisfies

$$\mathbb{E}[\hat{\mathbf{f}}] = \mathbf{f}.$$

Proof. Recall that $\tilde{\mathbf{s}} = W^{1/2} \mathbf{s}$, and from the de-biasing procedure, each entry $s_{j,a}$ satisfies:

$$\mathbb{E}[s_{j,a}] = \sum_{x: x \bmod m_j = a} f_x = (A\mathbf{f})_{j,a}.$$

Stacking over all blocks yields:

$$\mathbb{E}[\tilde{\mathbf{s}}] = W^{1/2} A\mathbf{f} = A_w \mathbf{f}.$$

Taking the expectation of the estimator gives:

$$\mathbb{E}[\hat{\mathbf{f}}] = (A_w^\top A_w)^{-1} A_w^\top \mathbb{E}[\tilde{\mathbf{s}}] = (A_w^\top A_w)^{-1} A_w^\top A_w \mathbf{f} = \mathbf{f}. \quad \square$$

Corollary 1 (Asymptotic Unbiasedness, $\lambda > 0$). *Let $\hat{\mathbf{f}}_\lambda$ be the regularized estimator:*

$$\hat{\mathbf{f}}_\lambda = (A_w^\top A_w + \lambda I)^{-1} A_w^\top \tilde{\mathbf{s}}.$$

Then, as $\lambda \rightarrow 0$, the estimator converges in expectation to the true histogram:

$$\mathbb{E}[\hat{\mathbf{f}}_\lambda] \rightarrow \mathbf{f}.$$

*For practical settings (*e.g.*, $\lambda = 1/\varepsilon^2$), the bias introduced is $O(\lambda)$, which becomes negligible for large ε (weaker privacy). When ε is small (strong privacy), regularization bias may be more significant.*

Remark 1. *In practice, a small ridge term $\lambda > 0$ can be used to improve numerical conditioning and accelerate convergence of iterative solvers. Although this introduces a small bias, the estimator remains practically unbiased.*

4 Analysis of MSS

This section analyzes MSS along the four axes of the multi-constraint regime highlighted in Section 1. We bound its communication and decoding costs, derive closed-form and worst-case MSE expressions in terms of the condition number κ , show how moduli selection controls κ , and quantify its resilience to data reconstruction attacks.

4.1 Communication Cost

Each user sends a pair (J, Z) as defined in Algorithm 1, where $J \in [\ell]$ is the block index and $Z \subseteq [m_j]$ is a noisy subset of size ω_j produced via SS. The number of bits is:

$$\lceil \log_2 \ell \rceil + \frac{1}{\ell} \sum_{j=0}^{\ell-1} \left\lceil \log_2 \binom{m_j}{\omega_j} \right\rceil. \quad (2)$$

This reflects the average-case encoding cost, assuming uniform selection of the block index and optimal enumeration-based encoding over the $\binom{m_j}{\omega_j}$ possible subsets.

4.2 Server-Side Decoding and Aggregation Cost

The total server-side runtime consists of three main phases: collecting residue counts, forming the debiased observation vector, and solving the weighted least-squares problem.

First, a single pass over the n user reports is sufficient to aggregate per-block residue counts and compute normalization factors, requiring $O(n)$ time. Next, debiasing each empirical histogram \tilde{y}_j and applying variance-optimal weights to form the scaled vector $\tilde{\mathbf{y}}$ takes $O(\sum_{j=0}^{\ell-1} m_j)$ operations.

Finally, to recover the histogram $\hat{\mathbf{f}} \in \mathbb{R}^k$, the server solves a regularized least-squares system using the sparse weighted design matrix $A_w \in \mathbb{R}^{T \times k}$, where $T = \sum_j m_j$. As mentioned in Section 3.2, we use LSMR (Fong and Saunders 2011), which takes r iterations. Each iteration performs one matrix-vector multiplication with A_w and A_w^\top , costing $O(k\ell)$ due to the structured sparsity of the CRT design.

Overall runtime. The total decoding complexity is thus:

$$O\left(n + k\ell + \sum_{j=0}^{\ell-1} m_j\right).$$

In practice, one can select moduli such that $\sum_j m_j = O(k)$ (since each $m_j < k$ and $\ell = \Theta(\log k)$), yielding:

$$O(n + k\ell) \quad \text{with } \ell = \Theta(\log k).$$

Simplified bounds.

- When the number of solver iterations is constant ($r = O(1)$), the runtime becomes: $\Theta(n + k \log k)$.
- In the worst case, when convergence requires $r = \Theta(k)$ iterations, the runtime becomes: $\Theta(n + k^2 \log k)$.

4.3 Closed-form Variance of the Estimator

Let $\mathbf{f} = (f_0, \dots, f_{k-1})^\top \in \mathbb{R}^k$ denote the unknown population histogram over domain $[k]$, satisfying $\sum_{x=0}^{k-1} f_x = 1$. For each modulus m_j (with $j = 0, \dots, \ell - 1$), define the corresponding RNS marginal distribution:

$$g_{j,a} = \sum_{x: x \bmod m_j = a} f_x, \quad \mathbf{g}_j = (g_{j,0}, \dots, g_{j,m_j-1})^\top.$$

Noisy subset selection. Conditioned on a user sampling block $J = j$, the probability that a specific residue $a \in [m_j]$ appears in the subset Z is:

$$\begin{aligned} \pi_{j,a} &= \mathbb{P}[a \in Z \mid J = j] = p_j \cdot g_{j,a} + q_j \cdot (1 - g_{j,a}) \\ &= q_j + (p_j - q_j) \cdot g_{j,a}, \end{aligned}$$

where p_j and q_j are the true and false inclusion probabilities of `SubsetSelection` for block j , respectively. Let $\mathbf{Y}_j \in \{0, 1\}^{m_j}$ be the indicator vector for residues in Z . The server computes:

$$\tilde{\mathbf{y}}_j = \frac{\bar{\mathbf{y}}_j - q_j \mathbf{1}}{p_j - q_j}, \quad \bar{\mathbf{y}}_j = \frac{1}{n_j} \sum_{u=1}^{n_j} \mathbf{Y}_j^{(u)}.$$

The covariance of $\tilde{\mathbf{y}}_j$ is:

$$\Sigma_j(\mathbf{f}, n_j) = \frac{1}{n_j(p_j - q_j)^2} (\text{diag}(\boldsymbol{\pi}_j) - \boldsymbol{\pi}_j \boldsymbol{\pi}_j^\top).$$

Global covariance. Define the global observation vector:

$$\tilde{\mathbf{y}} = \begin{bmatrix} \tilde{\mathbf{y}}_0 \\ \vdots \\ \tilde{\mathbf{y}}_{\ell-1} \end{bmatrix} \in \mathbb{R}^T, \quad \text{with } T = \sum_{j=0}^{\ell-1} m_j.$$

Since each user contributes to exactly one block, these per-block estimators are negatively correlated. For $j \neq j'$, the cross-block covariance becomes:

$$\text{Cov}[\tilde{\mathbf{y}}_j, \tilde{\mathbf{y}}_{j'}] = -\frac{1}{n(p_j - q_j)(p_{j'} - q_{j'})} \boldsymbol{\pi}_j \boldsymbol{\pi}_{j'}^\top.$$

Stacking all components, the full covariance of $\tilde{\mathbf{y}}$ is:

$$\Sigma(\mathbf{f}) = \text{blockdiag}(\mathbb{E}_{n_j}[\Sigma_j(\mathbf{f}, n_j)]) - \frac{\ell}{n^2} \mathbf{u} \mathbf{u}^\top,$$

$$\text{with } \mathbf{u} = \begin{bmatrix} \boldsymbol{\pi}_0 / (p_0 - q_0) \\ \vdots \\ \boldsymbol{\pi}_{\ell-1} / (p_{\ell-1} - q_{\ell-1}) \end{bmatrix}.$$

Variance (i.e., Mean Squared Error – MSE). Since the MSS estimator is unbiased (see Theorem 2), its mean squared error coincides with its variance: $\text{MSE}_{\text{MSS}}(\mathbf{f}) = \text{Var}[\hat{\mathbf{f}}]$. Let $G = A_w^\dagger = (\tilde{A}^\top \tilde{A} + \lambda I)^{-1} \tilde{A}^\top$ be the gain matrix used in the estimator $\hat{\mathbf{f}}$. The variance of MSS is:

$$\text{MSE}_{\text{MSS}}(\hat{\mathbf{f}}) = \frac{1}{k} \text{Tr}(G \Sigma(\mathbf{f}) G^\top). \quad (3)$$

This expression holds for arbitrary distributions \mathbf{f} and reflects the protocol's total estimation risk.

Worst-case bound. Let $\text{MSE}_{\text{SS}}(\varepsilon, n) = \frac{4e^\varepsilon}{n(e^\varepsilon - 1)^2}$ denote the worst-case per-coordinate MSE of a single SS block. Stacking the ℓ blocks, the MSS decoder outputs $\hat{\mathbf{f}} = G\tilde{\mathbf{y}}$ with $G := A_w^\dagger$. The covariance of $\hat{\mathbf{f}}$ is $G\Sigma G^\top$, where Σ is block-diagonal with copies of the SS covariance.

Theorem 3 (Worst-Case MSE of MSS). *For any frequency vector \mathbf{f} and any moduli choice with finite $\kappa = \text{cond}(A_w)$,*

$$\text{MSE}_{\text{MSS}}(\hat{\mathbf{f}}) = \frac{1}{k} \text{Tr}(G\Sigma(\mathbf{f})G^\top) \leq \boxed{\frac{4\kappa e^\varepsilon}{n(e^\varepsilon - 1)^2}}. \quad (4)$$

Proof Sketch. For the unbiased estimator $\hat{\mathbf{f}} = G\tilde{\mathbf{y}}$ one has $\text{MSE} = k^{-1} \text{Tr}(G\Sigma G^\top)$, where Σ is the covariance of $\tilde{\mathbf{y}}$. Trace–Cauchy–Schwarz yields $\text{Tr}(G\Sigma G^\top) \leq \|G\|_2^2 \text{Tr}(\Sigma)$. Since $G = A_w^\dagger$, $\|G\|_2 = \sigma_{\min}^{-1}(A_w) = \kappa/\sigma_{\max}(A_w)$. Every row of A_w contains exactly one entry 1; hence $\sigma_{\max}(A_w) = \sqrt{S}$ with $S = \sum_j w_j$. Cancelling S gives Eq. (4). \square

Bounding κ analytically. Let the ℓ moduli $m_0, \dots, m_{\ell-1}$ be pairwise-coprime primes drawn from the interval $[L, H]$ with $L = \frac{k}{\beta\ell}$ and $H = \frac{\beta k}{\ell}$ where $\beta > 1$. Set

$$T^* = \left\lceil \frac{\ln k}{\ln(k/(\beta\ell))} \right\rceil, \quad \alpha = \frac{w_{\max}}{w_{\min}} \leq \frac{\beta + e^\varepsilon}{1/\beta + e^\varepsilon}. \quad (5)$$

Theorem 4 (Condition-number bound). *With probability 1 over the random prime selection,*

$$\kappa = \text{cond}(A_w) \leq \alpha \frac{\ell + T^*}{\ell - T^*}. \quad (6)$$

Thus, any $\ell \geq \frac{1 + \kappa_{\max}/\alpha}{\kappa_{\max}/\alpha - 1} T^*$ guarantees $\kappa \leq \kappa_{\max}$.

Proof Sketch. Because each column of A has exactly one “1” per block, the Gram matrix $G = A^\top W A$ has diagonal entries $S = \sum_j w_j$ and at most T^* off-diagonal collisions per row. Gershgorin discs give $\lambda_{\min}(G) \geq w_{\min}(\ell - T^*)$ and $\lambda_{\max}(G) \leq w_{\max}(\ell + T^*)$, yielding the claim. The bound on T^* follows from the fact that $\prod_{j \in C(x, x')} m_j$ divides $|x - x'| < k$ for any distinct $x, x' \in [k]$. \square

Remark 2 (Conservative bound). *The lower limit $\ell_{\text{theory}} = (1 + \kappa_{\max}/\alpha)/(\kappa_{\max}/\alpha - 1) T^*$ is deliberately conservative. It combines (i) the largest possible weight ratio w_{\max}/w_{\min} (obtained from the extremal moduli in $[L, H]$) with (ii) Gershgorin discs that assume the maximum number T^* of off-diagonal collisions. Both choices over-estimate $\kappa(A_w)$, so the bound is sufficient but generally not tight; smaller values of ℓ frequently satisfy $\kappa \leq \kappa_{\max}$ in practice.*

Optimized Moduli Selection The accuracy-bandwidth trade-off in `ModularSubsetSelection` is dictated by the pairwise-coprime prime moduli $\mathbf{m} = (m_0, \dots, m_{\ell-1})$. A valid tuple of moduli for `ModularSubsetSelection` must:

- (i) **cover the domain:** $\prod_{j=0}^{\ell-1} m_j \geq k$ (CRT property);
- (ii) **ensure full rank:** $\sum_{j=0}^{\ell-1} (m_j - 1) \geq k$; and

- (iii) **yield a small condition number** $\kappa = \text{cond}(A_w) \leq \kappa_{\max}$ so that Eq. (4) guarantees low worst-case MSE.

Because an exhaustive search is intractable, we combine the analytic κ -bound (Theorem 4) with lightweight random sampling. The full pseudocode for Steps 1–3 below is provided in Appendix A of the extended version (Arcolezi 2025).

Step 1: Analytic lower bound for ℓ . Fixing a user-defined target κ_{\max} (we use $\kappa_{\max} = 10$), Theorem 4 yields a *necessary* lower limit $\ell_{\text{theory}} = (1 + \kappa_{\max}/\alpha)/(\kappa_{\max}/\alpha - 1) T^*$. Because this bound is loose (Remark 2), our implementation still *starts the search at $\ell_{\text{theory}} = 2$* and simply discards any candidate that eventually violates $\kappa \leq \kappa_{\max}$.

Step 2: Prime-band sampling. Let the user choose a search width β (default $\beta = 20$). For each $\ell \in \{2, \dots, \ell_{\max}\}$ we draw ℓ distinct primes from $L = k/(\beta\ell)$, $H = (\beta k)/\ell$. If coverage or rank fails, we “bump” random moduli to the next prime until (i)–(ii) hold. Sampling stops as soon as a tuple reaches $\kappa \leq \kappa_{\max}$ or after `#trials`.

Step 3: Moduli selection by exact MSE. For every candidate tuple that satisfies (i)–(iii), we compute the exact MSE in (4) and keep the tuple with the smallest value, thereby selecting the communication-optimal configuration that meets the target condition number.

Deterministic fallback. If no tuple attains κ_{\max} in `#trials`, we fall back to the first ℓ primes $\geq \lceil k^{1/\ell} \rceil$ and deterministically increment them (left to right) until CRT and rank conditions hold; the analytic κ -bound still applies.

4.4 Data Reconstruction Attack on MSS

Following recent work on adversarial analysis of LDP protocols (Emre Gursoy et al. 2022; Arcolezi et al. 2023), we consider a Bayesian attacker who observes a single user report $y = (J, Z)$, knows the full protocol specification, and assumes a uniform prior $\Pr[x] = 1/k$ over the domain.

The adversary aims to infer the true user input x by computing the posterior distribution and selecting the most probable value. The probability of a correct guess, $\Pr[\hat{x} = x]$, defines the per-message *Data Reconstruction Attack* (DRA).

Posterior support. Given an MSS report $y = (j, Z)$, the attacker infers the following posterior support set:

$$\mathcal{S}_{j,Z} = \{x \in [k] \mid x \bmod m_j \in Z\},$$

which includes all domain elements whose residue modulo m_j appears in the subset Z . Its size satisfies

$$|\mathcal{S}_{j,Z}| \leq \omega_j \cdot \left\lceil \frac{k}{m_j} \right\rceil = \omega_j C_j,$$

where $C_j = \lceil k/m_j \rceil$ is the max number of domain values per residue. Assuming no further knowledge, the optimal strategy is to sample uniformly from $\mathcal{S}_{j,Z}$, yielding success rate $1/|\mathcal{S}_{j,Z}|$ when $x \in \mathcal{S}_{j,Z}$.

Upper-bound on the expected DRA. The attacker succeeds only if the true residue is included in Z , which occurs with probability p_j for block j . Conditioned on this, the success rate is $1/(\omega_j C_j)$, where $C_j = \lceil k/m_j \rceil$. To keep the analysis concise, we upper-bound the DRA by assuming the **largest possible** posterior set size $\omega_j C_j$:

$$\text{DRA}_j = \underbrace{p_j}_{\text{truth in } Z} \cdot \underbrace{\frac{1}{|\mathcal{S}_{j,Z}|}}_{\text{Bayes rule}} \leq p_j \frac{1}{\omega_j C_j}.$$

Averaging over the uniformly chosen block index J gives the following closed-form *upper bound* on the DRA:

$$\widehat{\mathbb{E}}[\text{DRA}]_{\text{MSS}} := \frac{1}{\ell} \sum_{j=0}^{\ell-1} \frac{p_j}{\omega_j \lceil k/m_j \rceil} \leq \mathbb{E}[\text{DRA}]_{\text{MSS}}. \quad (7)$$

The equality holds whenever each residue class supports the same number of domain elements (*e.g.*, when $m_j \mid k$), but in general the bound can be slightly loose. A tight expression, together with a complete proof, is provided in Appendix B of the extended version (Arcolezi 2025).

5 Experimental Results

In this section, we aim to evaluate the four aspects mentioned in Section 1: (i) **Utility**, (ii) **Communication**, (iii) **Server runtime**, and (iv) **Attackability**. All experiments were run on a Desktop computer with a 3.2 GHz Intel Core i9 processor, 64 GB RAM, and Python 3.11.

Setting. We benchmark our ModularSubsetSelection against state-of-the-art single-message frequency oracles: ProjectiveGeometryResponse, RandomizedResponse, SubsetSelection, and OptimalUnaryEncoding (the optimized RAPPOR variant). Since worst-case MSE is distribution-independent (Table 1), we adopt the synthetic Zipf ($s = 3$) and Spike ($\mathbf{f} = [1, 0, \dots, 0]$) benchmarks from (Feldman et al. 2022), both known to induce high estimation variance. Unless noted otherwise, we fix $n = 10,000$ users, domain $k \in \{1,024, 22,000\}$, privacy budget $\epsilon \in \{0.5, 1.0, \dots, 4.5, 5.0\}$, and average results over 300 independent trials.

Utility comparison. Fig. 1 reports the MSE of each protocol as a function of the privacy budget ϵ , under both Zipf and Spike distributions. Notably, the relative ordering and behavior of all protocols remain consistent across Zipf and Spike distributions, confirming that our conclusions are robust to underlying data characteristics. Among all LDP frequency-oracle protocols, GRR consistently yields the highest error, due to its $\Theta(k/e^\epsilon)$ scaling and lack of structure exploitation. In contrast, OUE, SS, and PGR achieve near-optimal utility across all settings, as all three match the information-theoretic MSE bound for single-message LDP protocols. MSS tracks SS and PGR within $\leq 1.3\times$ throughout, showing that the modular encoding adds only negligible distortion.

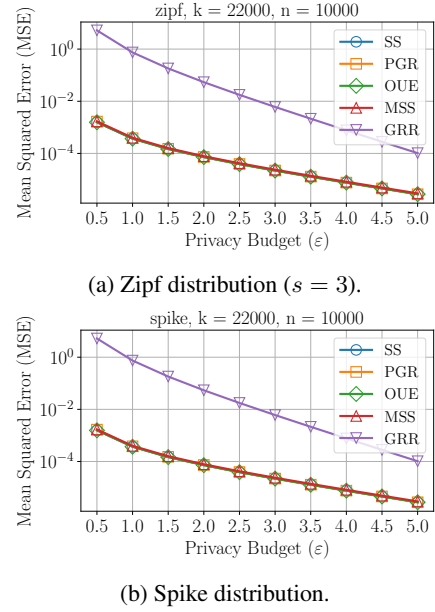


Figure 1: MSE vs. privacy parameter ϵ for $k = 22,000$ and $n = 10,000$, under (a) Zipf and (b) Spike distributions. MSS closely tracks the near-optimal error curves of SS and PGR.

Communication cost. Fig. 2 shows the number of bits each user must transmit under both SS and MSS, as a function of ϵ for $k = 1,024$ and $k = 22,000$. Since MSS relies on a randomized moduli selection process, we report its average and standard deviation over the 300 runs. Across all settings, MSS consistently achieves lower communication cost than SS, up to one-half in high-privacy regimes, while retaining comparable accuracy (see Fig. 1). We omit GRR and PGR from the figure for clarity: their per-report message length is fixed for a given k (both use $O(\log k)$ bits) and is already summarized analytically in Table 1. GRR and PGR therefore form communication-efficient baselines, but as shown in Fig. 1 and Table 2, they pay respectively in much higher MSE (GRR) or substantially higher decoding cost (PGR).

Server runtime. We now compare the server-side runtime of our MSS protocol against the state-of-the-art PGR scheme by Feldman et al. (2022). We run both protocols on the Zipf dataset of size $n = 10,000$ and domain size $k = 22,000$, across several privacy levels. Table 2 reports the average and standard deviation of the server decoding time (in seconds) over 300 trials. MSS consistently outperforms PGR by large margins, achieving decoding speed-ups between $11\times$ and $448\times$. This performance gap stems from their algorithmic differences: MSS solves a sparse weighted least-squares problem, while PGR relies on algebraic decoding over finite fields. We do not plot GRR here, since its server cost is essentially a single histogram pass $O(n + k)$ and thus serves as a trivial lower bound on runtime; however, as Fig. 1 and Fig. 3 show, GRR is not competitive in our multi-constraint regime due to its much worse utility and attackability. The runtime spike at $\epsilon = 4.5$ for PGR likely arises from parameter rounding and structural constraints in

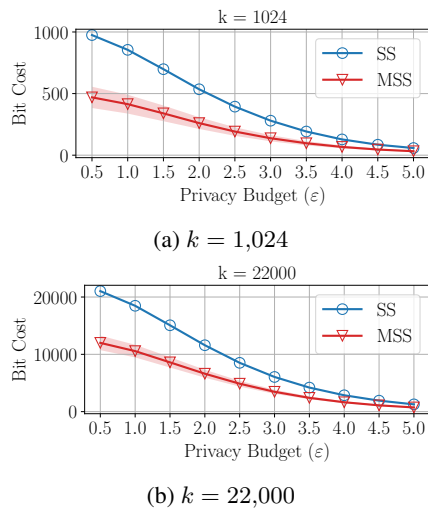


Figure 2: Per-user message length (bits) of SS and MSS as a function of the ϵ , for two domain sizes. MSS consistently requires fewer bits than SS, especially in high privacy regimes.

its projective geometry design.

ϵ	Server-Side Runtime (in seconds)		
	MSS	PGR	MSS Speed-up
2.0	0.160 ± 0.027	2.897 ± 0.220	18.1×
2.5	0.275 ± 0.094	4.019 ± 0.283	14.6×
3.0	0.272 ± 0.086	9.618 ± 0.679	35.4×
3.5	0.162 ± 0.050	1.908 ± 0.138	11.7×
4.0	0.168 ± 0.056	11.461 ± 0.702	68.3×
4.5	0.127 ± 0.047	56.906 ± 3.570	447.8×
5.0	0.152 ± 0.054	3.208 ± 0.198	21.1×

Table 2: Average ± std of server-side runtime (in seconds) for our MSS and PGR, with $k = 22,000$ and $n = 10,000$. MSS is consistently faster than PGR.

Attackability. We now evaluate the vulnerability of each LDP protocol to a single-message data reconstruction attack (DRA) (see Section 4.4). Fig. 3 shows the empirical DRA as a function of the privacy budget ϵ , under the Zipf distribution for domain sizes $k = 100$ and $k = 1,024$. MSS consistently achieves the lowest DRA across all ϵ values, confirming its robustness to reconstruction attacks. This is due to its modular randomization strategy, which distributes the probability mass across multiple residue classes, making inference more challenging. In contrast, GRR and SS exhibit higher attackability, especially for small k , as their output space is tightly linked to the input domain. PGR behaves comparably to SS/MSS/OUE at moderate ϵ , but for larger budgets its DRA increases sharply when k is smaller than the projective-domain size $K = (q^t - 1)/(q - 1)$ required by its internal geometry. This truncation mismatch breaks PGR’s combinatorial symmetry and makes certain messages disproportionately informative. A fair comparison using the

non-truncated setting $k = K$ is provided in Appendix D of the extended version (Arcolezi 2025).

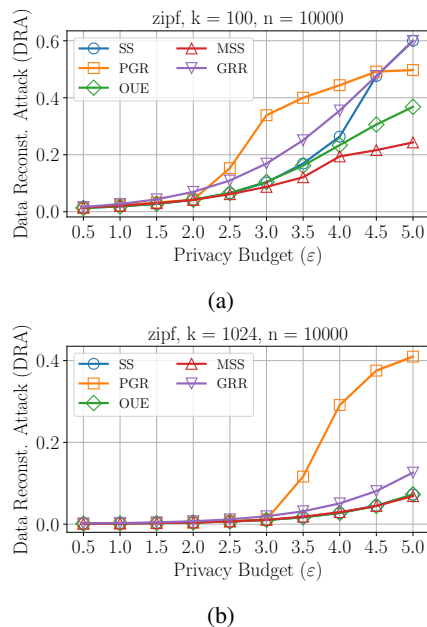


Figure 3: Empirical Data Reconstruction Attack (DRA) of each protocol under the Zipf distribution, evaluated over $n = 10,000$ users. MSS provides the strongest protection across both small and large domains.

Summary. Across all experiments, MSS matches the near-optimal utility of SS, OUE, and PGR, while requiring substantially fewer transmitted bits than SS (Fig. 2) and decoding orders of magnitude faster than PGR (Table 2). At the same time, MSS achieves the lowest empirical attack success rate among all protocols evaluated (Fig. 3), demonstrating strong robustness to single-message reconstruction attacks. Taken together, these results position MSS in an effective operating regime for large-domain LDP frequency estimation, jointly balancing accuracy, communication cost, server-side computation, and attackability.

Ablation studies. Appendix D in (Arcolezi 2025) presents additional experiments under different data distributions, numbers of users, and broader domain sizes, including several ablation studies that further validate our findings.

6 Conclusion

We introduce ModularSubsetSelection (MSS), a simple and powerful LDP-frequency oracle that leverages modular arithmetic to balance privacy, utility, communication, and attackability. Our results show that MSS achieves utility comparable to state-of-the-art protocols like SS and PGR, while significantly reducing communication cost compared to SS, lowering server runtime compared to PGR, and offering stronger protection against data reconstruction attacks. Future work includes extending to other statistical tasks, such as heavy hitters and multidimensional estimation.

Acknowledgments

The author thanks Patricia Guerra-Balboa for her helpful comments on an earlier draft, and the anonymous reviewers for their insightful suggestions. This work has been supported by the French National Research Agency (ANR): “ANR-24-CE23-6239” and “ANR-23-IACL-0006”.

References

- Acharya, J.; Sun, Z.; and Zhang, H. 2019. Hadamard Response: Estimating Distributions Privately, Efficiently, and with Little Communication. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 1120–1129.
- Arcolezi, H. H. 2025. Private Frequency Estimation Via Residue Number Systems. *arXiv preprint arXiv:2511.11569*.
- Arcolezi, H. H.; and Gambs, S. 2025. Revisiting Locally Differentially Private Protocols: Towards Better Trade-offs in Privacy, Utility, and Attack Resistance. *arXiv preprint arXiv:2503.01482*.
- Arcolezi, H. H.; Gambs, S.; Couchot, J.-F.; and Palamidessi, C. 2023. On the Risks of Collecting Multidimensional Data Under Local Differential Privacy. *Proc. VLDB Endow.*, 16(5): 1126–1139.
- Ding, B.; Kulkarni, J.; and Yekhanin, S. 2017. Collecting Telemetry Data Privately. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 3571–3580. Curran Associates, Inc.
- Duchi, J. C.; Jordan, M. I.; and Wainwright, M. J. 2013. Local Privacy and Statistical Minimax Rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 429–438. IEEE.
- Emre Gursoy, M.; Liu, L.; Chow, K.-H.; Truex, S.; and Wei, W. 2022. An Adversarial Approach to Protocol Analysis and Selection in Local Differential Privacy. *IEEE Transactions on Information Forensics and Security*, 17: 1785–1799.
- Erlingsson, U.; Pihur, V.; and Korolova, A. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 1054–1067. New York, NY, USA: ACM.
- Feldman, V.; Nelson, J.; Nguyen, H.; and Talwar, K. 2022. Private frequency estimation via projective geometry. In *Proceedings of the 39th International Conference on Machine Learning*, 6418–6433. PMLR.
- Fong, D. C.-L.; and Saunders, M. 2011. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5): 2950–2971.
- Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33: 16937–16947.
- Guerra-Balboa, P.; Sauer, A.; and Strufe, T. 2024. Analysis and Measurement of Attack Resilience of Differential Privacy. In *Proceedings of the 23rd Workshop on Privacy in the Electronic Society*, WPES '24, 155–171. New York, NY, USA: Association for Computing Machinery. ISBN 9798400712395.
- Hastie, T.; Tibshirani, R.; Friedman, J. H.; and Friedman, J. H. 2009. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.
- Hayes, J.; Balle, B.; and Mahloujifar, S. 2023. Bounding training data reconstruction in dp-sgd. *Advances in neural information processing systems*, 36: 78696–78722.
- Kairouz, P.; Bonawitz, K.; and Ramage, D. 2016. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, 2436–2444. PMLR.
- Kasiviswanathan, S. P.; Lee, H. K.; Nissim, K.; Raskhodnikova, S.; and Smith, A. 2011. What Can We Learn Privately? *SIAM Journal on Computing*, 40(3): 793–826.
- Sun, Z.; Kairouz, P.; Sun, H.; Gascon, A.; and Suresh, A. T. 2024. Private federated discovery of out-of-vocabulary words for gboard. *arXiv preprint arXiv:2404.11607*.
- Szabó, N. S.; and Tanaka, R. I. 1967. *Residue arithmetic and its applications to computer technology*. McGraw-Hill series in information processing and computers. McGraw-Hill.
- Apple Differential Privacy Team. 2017. Learning with privacy at scale. <https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf>.
- Wang, S.; Huang, L.; Wang, P.; Nie, Y.; Xu, H.; Yang, W.; Li, X.-Y.; and Qiao, C. 2016. Mutual information optimally local private discrete distribution estimation. *arXiv preprint arXiv:1607.08025*.
- Wang, T.; Blocki, J.; Li, N.; and Jha, S. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, 729–745. USENIX Association.
- Warner, S. L. 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *Journal of the American Statistical Association*, 60(309): 63–69.