Problem Statement

- Outsourced Aggregate Queries Services Model
  - Three parties: data owner, service provider and client.
  - Aggregate queries on \( set-valued \) data.

- Challenges:
  - Privacy Clients cannot know the feature’s origin.
  - Integrity Clients can verify the result correctness.
  - Efficiency: Minimize communication and verification overhead.

Aggregate Queries Example on PGP Data

- Q1: Most common gene in Cupertino, CA (Zip: 95014).
  - Answer: ‘A-C130IR’

- Q2: Count the participants who carry the gene ‘R-G1886S’.
  - Answer: 4

- Q3: Find the most frequent genes with supports \( \geq 3 \) in ZIPS 20***.
  - Answer: ‘P-P12A’, ‘R-G1886S’

Authentication Algorithms on Aggregate Queries

- Sum/Count Query sums or counts the multiplicities of the queried feature in all selected objects.
  - Inflation checking: \( R \subseteq S \).
  - Deflation checking: \( (S \setminus R) \cap R = \emptyset \).

- Max/Top-k/FFQ Query returns features with the highest/top-k/above-threshold multiplicity.
  - Inflation checking: \( R \subseteq S \).
  - Deflation checking: \( (S \setminus R) \cap R = \emptyset \).
  - Completeness checking: \( (S \setminus R) \subseteq U \setminus (U \setminus R) \).

Bilinear Pairing

Let \( G, G_f \) be two groups. A pairing is a map \( e: G \times G \rightarrow G_f \), which satisfies:

- **Bilinearity** \( e(P, Q) = e(P, Q)^{\alpha} \).
- **Non-degeneracy** \( e(g, g) \neq 1 \).
- **Computability** Given \( P \) and \( Q \), it is easy to compute \( e(P, Q) \).

Privacy-Preserving Authentication Framework

Authentication Protocols on Multiset Operations

- \( \text{subset}(X_1, X_2) \) returns \( acc \) value of \( X_1 \rightarrow X_2 \).
  - \( SP \) computes \( acc(X_1 \rightarrow X_2) = g^{\delta_{X_1/X_2}} \prod_{X_1 \rightarrow X_2} \).
  - \( Client \) verifies \( e(acc(X_1, X_2)) = e(acc(X_1), g) \).
  - \( sum \) returns \( acc \) value of \( S = \sum_{Q} P(X) \).
    - Similar to \( \text{subset} \) process recursively.

- \( \text{empty}(X_1, X_2) \) returns whether \( \cap_{X_1/X_2} X = \emptyset \).
  - \( \text{Extended Euclidean Algorithm} \) \( \cap(X_1) = \emptyset \Rightarrow \exists Q \cdot \sum_{X_1} P(X_1) = 1 \).
  - \( \text{union}(X_1, X_2) \) returns \( acc \) value of \( U = \cup_{X_1} X_1 \).
    - Deflation checking: \( \bar{X} \subseteq X \land \bar{X} \subseteq U \land \bar{X} \subseteq U \).
    - Inflation checking: \( (U \setminus \bar{X}) \cap (U \setminus X_1) \subseteq (U \setminus \bar{X}) \).
    - \( times \) returns \( acc \) value of \( X \cdot Y \).
      - Similar to \( \text{sum} \), optimized using shift and add.

Example of Aggregate Queries

- \( S = \{(a, 6), (b, 1), (c, 4), (d, 3), (e, 2)\}, U = \{(a, 1), (b, 1), (c, 1), (d, 1), (e, 1)\} \).

- **Sum Query**
  - \( R = \{(a, 6)\} \).
    - Inflation checking: \( \{(a, 6)\} \subseteq \{(a, 6), (b, 1), (c, 4), (d, 3), (e, 2)\} \).
    - Deflation checking: \( \{(b, 1), (c, 4), (d, 3), (e, 2)\} \subseteq \{(a, 6)\} \).

- **Max Query**
  - \( R = \{(a, 6)\}, \bar{R} = \{(a, 1)\} \).
    - Inflation checking: \( \{(a, 6)\} \subseteq \{(a, 6), (b, 1), (c, 4), (d, 3), (e, 2)\} \).
    - Deflation checking: \( \{(b, 1), (c, 4), (d, 3), (e, 2)\} \subseteq \{(a, 6)\} \).
    - Completeness checking: \( \{(b, 1), (c, 4), (d, 3), (e, 2)\} \subseteq \{(a, 6), (b, 6), (c, 6), (d, 6), (e, 6)\} \).

Performance Evaluation