Authenticating Aggregate Queries over Set-Valued Data with Confidentiality

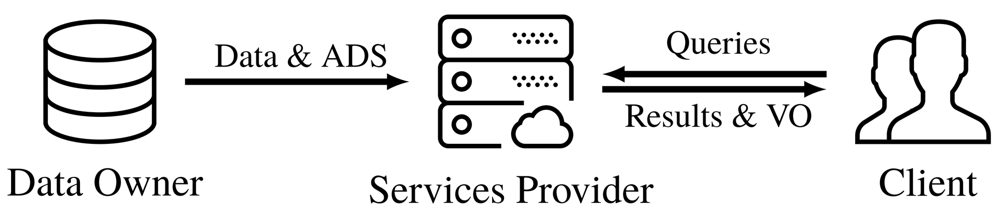
Cheng Xu[1], Qian Chen[1], Haibo Hu[2], Jianliang Xu[1], and Xiaojun Hei[3]

[1]Hong Kong Baptist University, Hong Kong [2]Hong Kong Polytechnic University, Hong Kong
[3]Huazhong University of Science and Technology, Wuhan, China

{chengxu, qchen, xujl}@comp.hkbu.edu.hk haibo.hu@polyu.edu.hk heixj@hust.edu.cn

Problem Statement

- **Outsourced Aggregate Query 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-C130R'}
- **Q2**: Count the participants who carry the gene 'R-G1886S'.
  Answer: 4
- **Q3**: Find the most frequent genes with supports ≥ 3 in ZIPs 20***.
  Answer: {'P-P12A', 'R-G1886S'}

PID	Zip	Mut-Genes
P1	95014	A-C130R, P-I696M
P2	20482	H-C282Y, P-P12A, R-G1886S
P3	95014	A-C130R, U-G71R, W-R611H
P4	01720	A-V2050L, H-C282Y, M-R52C, U-G71R
P5	20134	A-C130R, P-P12A, R-G1886S, S-E366K
P6	17868	C-R102G, R-G1886S
P7	55410	C-R102G, C-Q1334H, S-E288V
P8	20852	C-R102G, P-P12A, R-G1886S, K-T220M

Set-Valued Genome Dataset

BM Accumulator

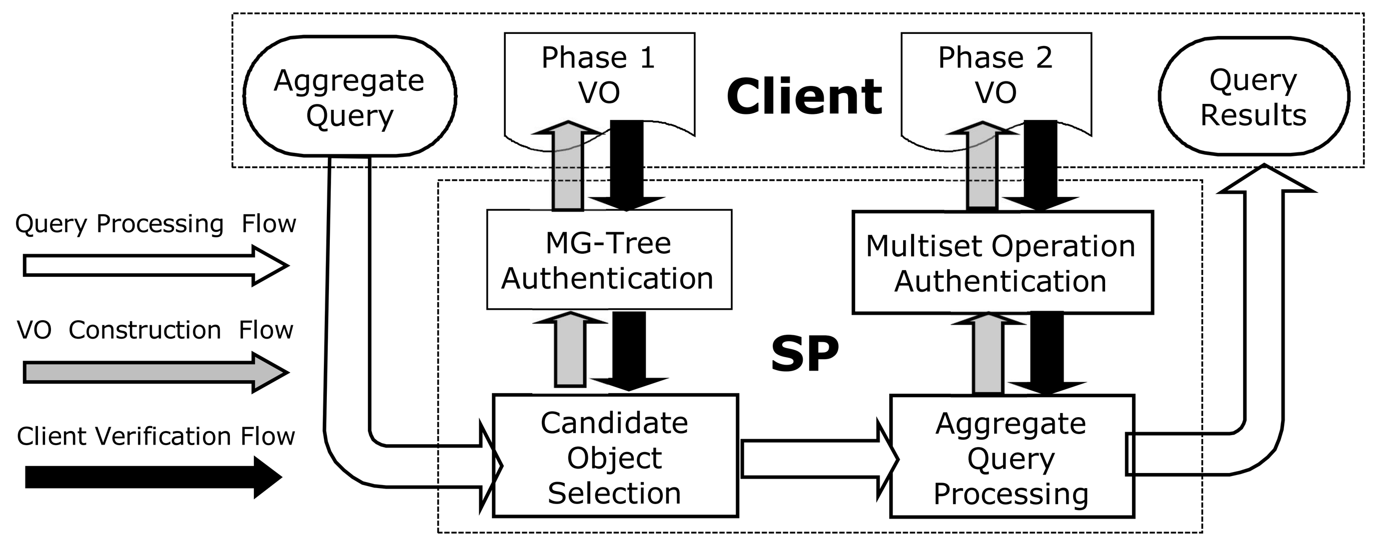
- To present a multiset $X = \{x_1, x_2, \ldots, x_m\}$, where g is a group generator and s is a **private** value of **DO**
$$acc(X) = g^{P(X)} = g^{\prod_{x_i \in X}(x_i+s)}$$
- e.g. $X_1 = \{(1,2),(2,1)\}$, $acc(X_1) = g^{(1+s)^2(2+s)}$.
- **SP** can prepare an $acc(\cdot)$ value by giving $g^s, g^{s^2}, \ldots$
- e.g. $acc(X_1) = g^{s^3+4s^2+5s+2} = g^{s^3} \cdot (g^{s^2})^4 \cdot (g^s)^5 \cdot g^2$.
- Randomized BM Accumulator:
$$acc(X) = g^{P(X)\cdot r_X} = g^{r_X \prod_{x_i \in X}(x_i+s)}$$

Bilinear Pairing

Let $\mathbb{G}, \mathbb{G}_T$ be two groups. A pairing is a map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, which satisfies:

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

Privacy-Preserving Authentication Framework



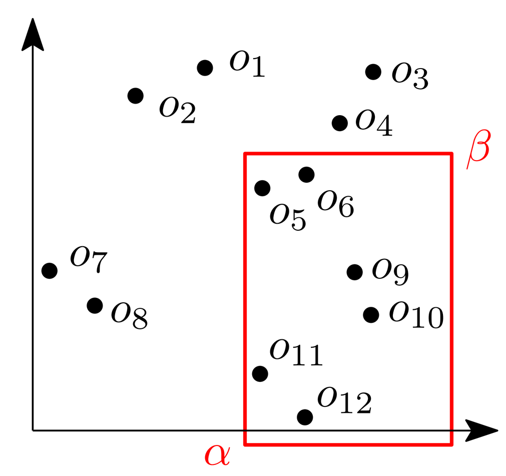
Authentication Protocols on Multiset Operations

- **subset** $sub(X_1, X_2)$ returns acc value of $X_1 - X_2$.
  - **SP** computes $acc(X_1 - X_2)^* = g^{r_{X_1}/r_{X_2} \prod_{x \in (X_1 - X_2)}(x+s)}$.
  - **Client** verifies $e(acc(X_2), acc(X_1 - X_2)^*) \stackrel{?}{=} e(acc(X_1), g)$.
- **sum** $sum(\{X_1, \ldots, X_n\})$ returns acc value of $S = \uplus_{i=1}^n X_i$.
  - Similar to **subset**, process recursively.
- **empty** $empty(\{X_1, \ldots, X_n\})$ returns whether $\cap_{i=1}^n X_i = \emptyset$.
  - **Extended Euclidean Algorithm** $\cap\{X_i\} = \emptyset \Rightarrow \exists Q_i$ s.t. $\sum_{i=1}^n Q_i \cdot P(X_i) = 1$.
- **union** $union(\{X_1, \ldots, X_n\})$ returns acc value of $U = \cup_{i=1}^n X_i$.
  - **Deflation checking**: $\widehat{X}_1 \subseteq U \wedge \widehat{X}_2 \subseteq U \wedge \cdots \widehat{X}_n \subseteq U$.
  - **Inflation checking**: $(U - \widehat{X}_1) \cap (U - \widehat{X}_2) \cap \cdots (U - \widehat{X}_n) = \emptyset$.
- **times** $times(X, t)$ returns acc value of $t \cdot X$.
  - Similar to **sum**, optimized using shift and add.

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

Example of Aggregate Queries

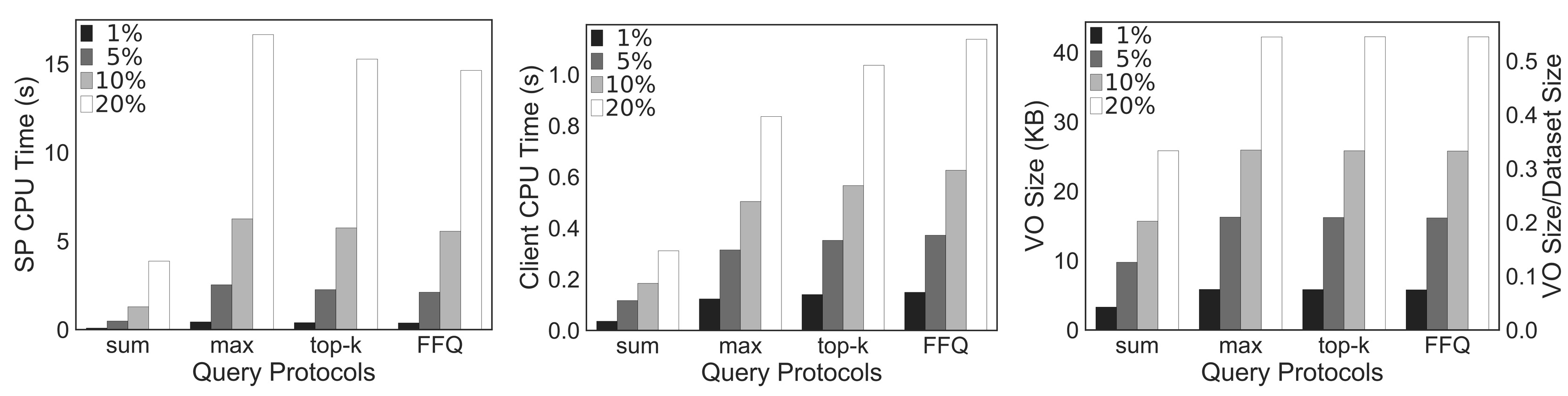


Object	Feature Set
o_5	$X_5 = \{c, d\}$
o_6	$X_6 = \{a, c, c\}$
o_9	$X_9 = \{a, a, d, e\}$
o_{10}	$X_{10} = \{a, d\}$
o_{11}	$X_{11} = \{a, b, e\}$
o_{12}	$X_{12} = \{a, c\}$

(a) Objects (b) Features

- $S = \{(a,6),(b,1),(c,4),(d,3),(e,2)\}$, $U = \{(a,1),(b,1),(c,1),(d,1),(e,1)\}$.
- **Sum Query**
  - $R = \{(a,6)\}$.
  - **Inflation checking**: $\{(a,6)\} \subseteq \{(a,6),(b,1),(c,4),(d,3),(e,2)\}$;
  - **Deflation checking**: $\{(b,1),(c,4),(d,3),(e,2)\} \cap \{(a,6)\} = \emptyset$.
- **Max Query**
  - $R = \{(a,6)\}$, $\widehat{R} = \{(a,1)\}$.
  - **Inflation checking**: $\{(a,6)\} \subseteq \{(a,6),(b,1),(c,4),(d,3),(e,2)\}$;
  - **Deflation checking**: $\{(b,1),(c,4),(d,3),(e,2)\} \cap \{(a,6)\} = \emptyset$.
  - **Completeness checking**: $\{(b,1),(c,4),(d,3),(e,2)\} \subseteq \{(b,6),(c,6),(d,6),(e,6)\}$.

Performance Evaluation



(a) SP CPU Time (b) Client CPU Time (c) VO size

- C. Xu, Q. Chen, H. Hu, J. Xu, and X. Hei, "Authenticating aggregate queries over set-valued data with confidentiality," *IEEE Trans. on Knowl. and Data Eng.*, vol. 30, no. 4, pp. 630–644, Apr. 2018.