# Privacy-Preserving Sensor Cloud

Hung Dang, Yun Long Chong, Francois Brun, Ee-Chien Chang
*School of Computing*
*National University of Singapore*

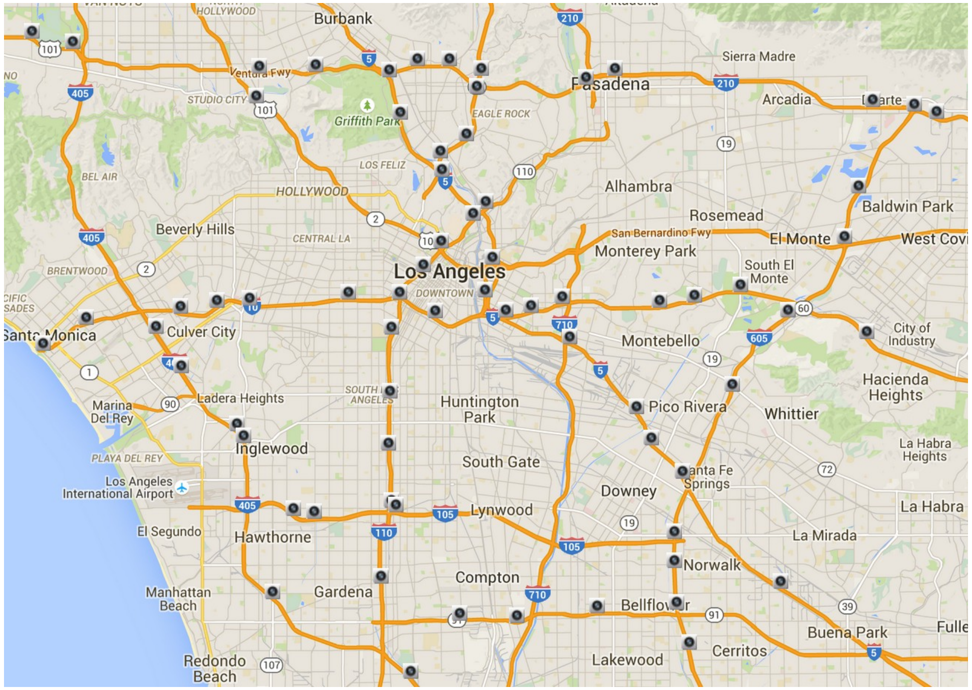
# Motivation

- The ubiquity of time series/multimedia data.
- Privacy concerns.
- The needs of sharings and/or collaboration.

# Application Scenario*

Sensor Cloud:

- Sensors are spatially arranged.

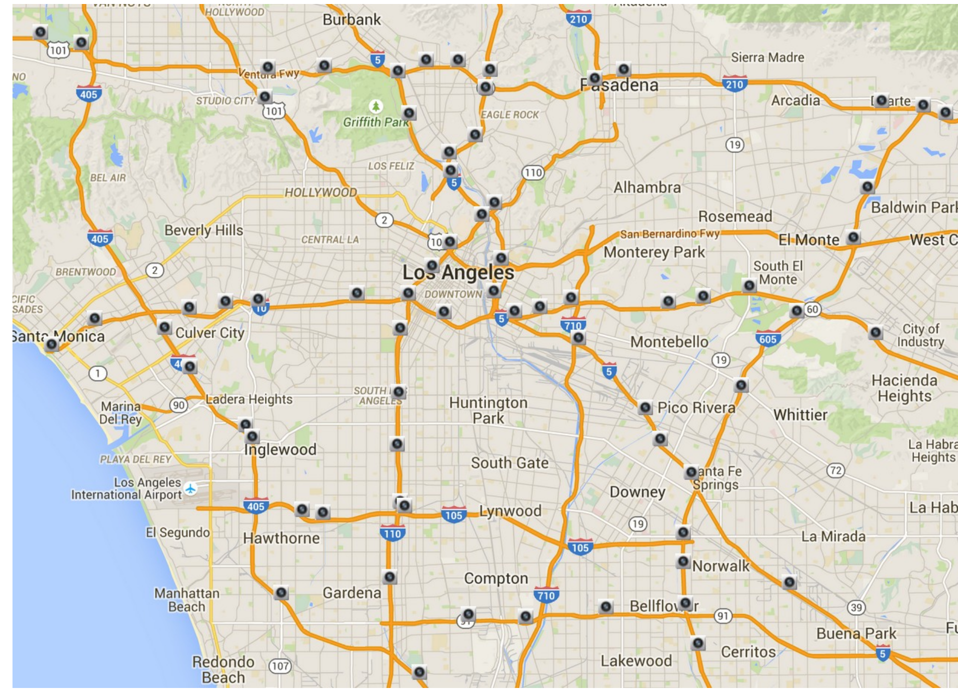


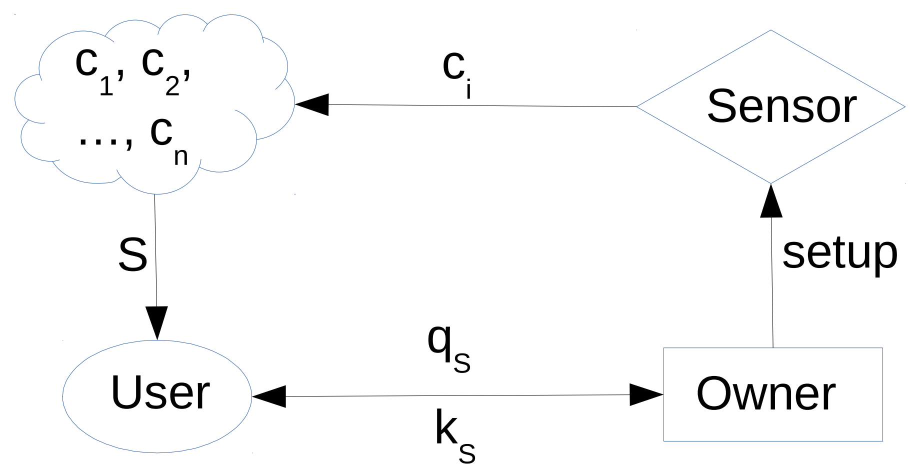

*Our techniques is also applicable to other applications involve multidimensional data.

# Application Scenario*

Sensor Cloud:

- Sensors are spatially arranged.
- Sensors continuously sense, encrypt and stream samples to the cloud.



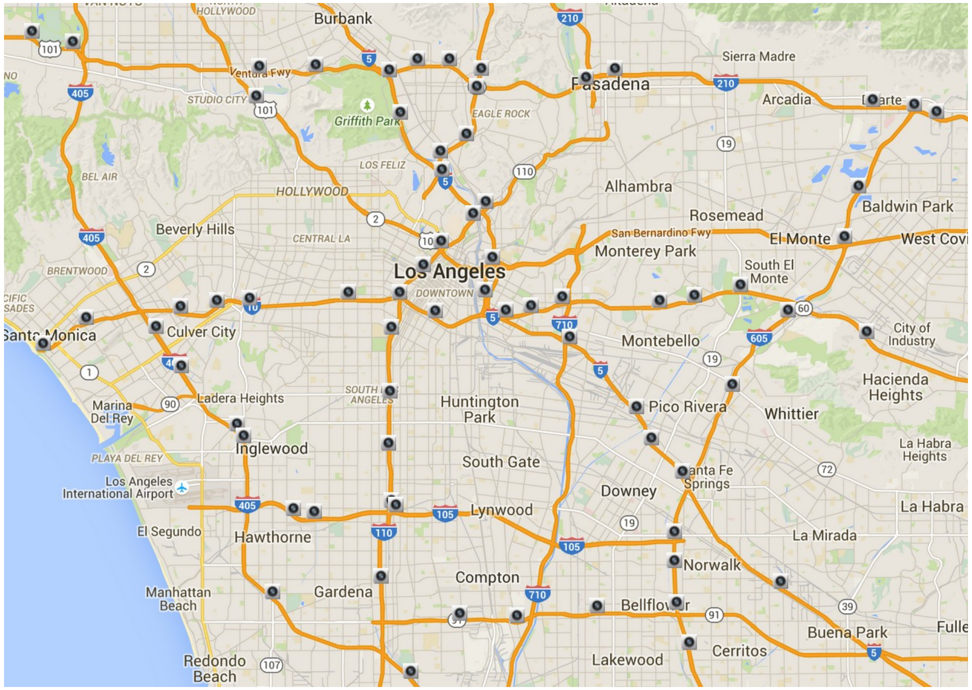



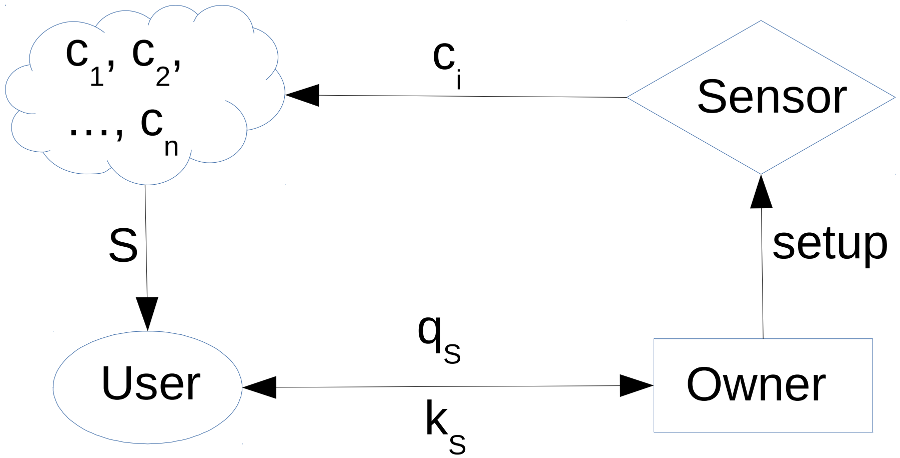

*Our techniques is also applicable to other applications involve multidimensional data.

# Application Scenario*

Sensor Cloud:

- Sensors are spatially arranged.
- Sensors continuously sense, encrypt and stream samples to the cloud.
- Samples are indexed by temporal and spatial meta-information.



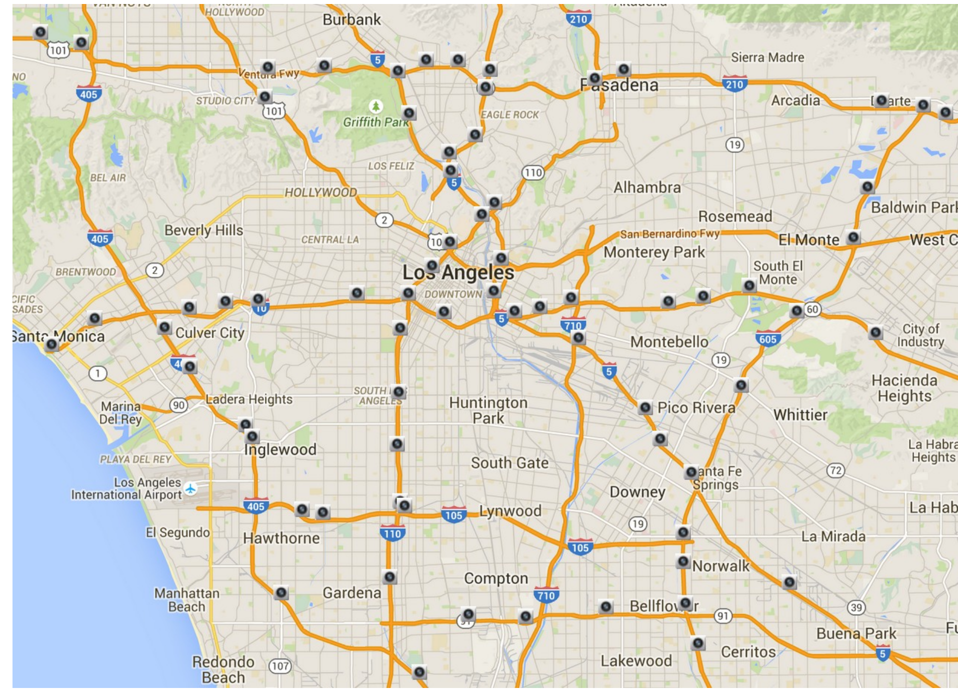



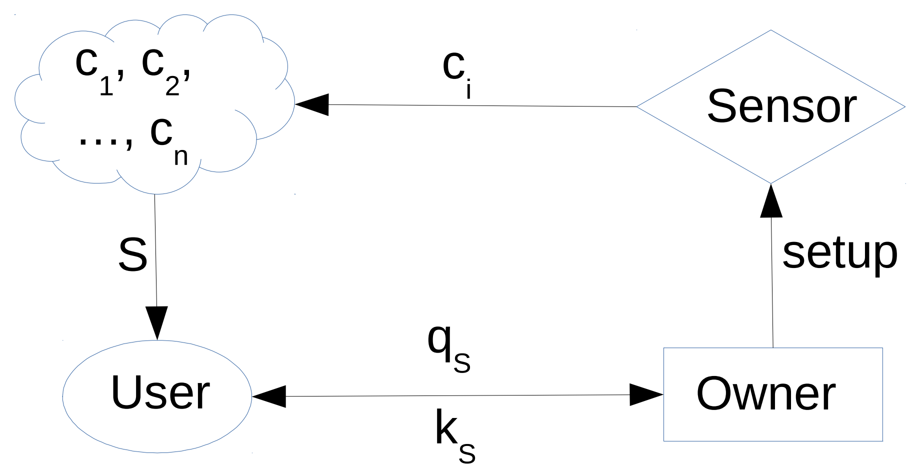

*Our techniques is also applicable to other applications involve multidimensional data.

# Application Scenario*

Sensor Cloud:

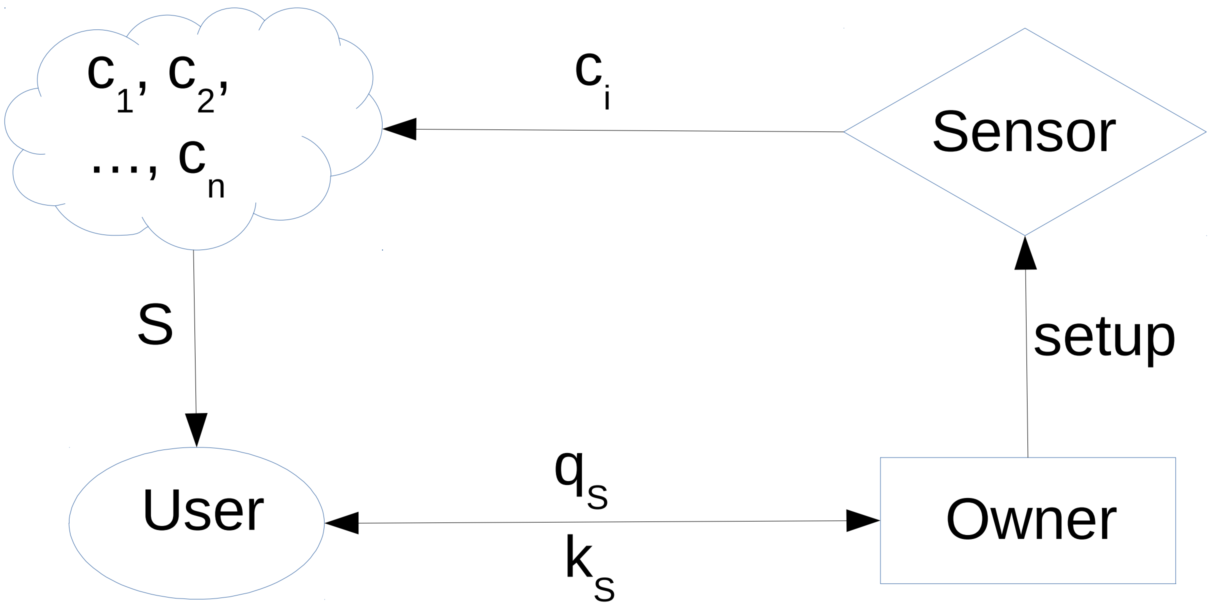
- Sensors are spatially arranged.
- Sensors continuously sense, encrypt and stream samples to the cloud.
- Samples are indexed by temporal and spatial meta-information.
- Sharings is done in query-and-response fashion: a query specifies a desired set of samples, a response grants access to the desired set.




*Our techniques is also applicable to other applications involve multidimensional data.

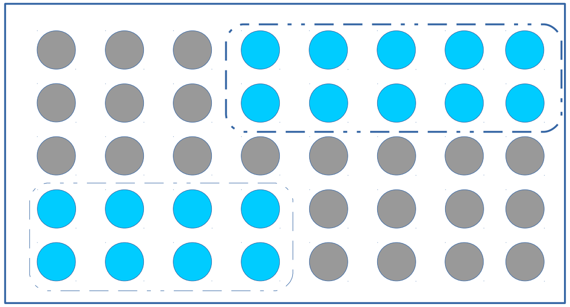
# System model



# Query Types

- Q1 - d-dimensional range query
  - Samples' indices form a d-dimensional.

    e.g.: all samples on street A on date X.

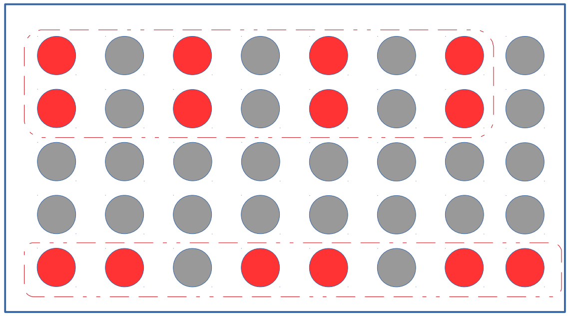


# Query Types

- Q1 - d-dimensional range query
  - Samples' indices form a d-dimensional.

    e.g.: all samples on street A on date X.
- Q2 - Down-sampling query
  - Samples' incides form a down-sampled lattice.

    e.g.: Y samples per each hours on street A on date X.



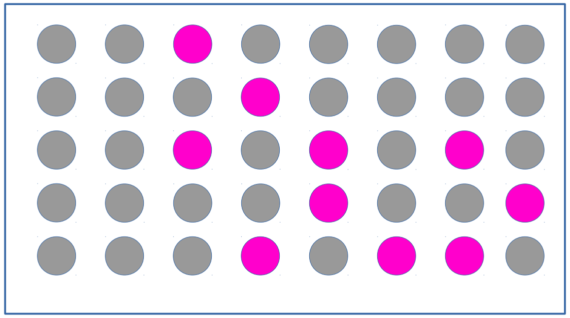
# Query Types

- Q1 - d-dimensional range query
  - Samples' indices form a d-dimensional.

    e.g.: all samples on street A on date X.
- Q2 - Down-sampling query
  - Samples' incides form a down-sampled lattice.

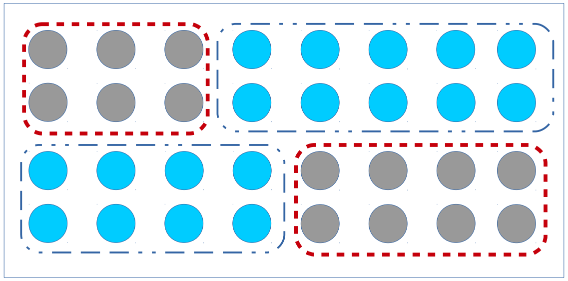
    e.g.: Y samples per each hours on street A on date X.
- Q3 - General query
  - Samples' indices may or may not have any structure.

    e.g.: random set of samples captured on date X.



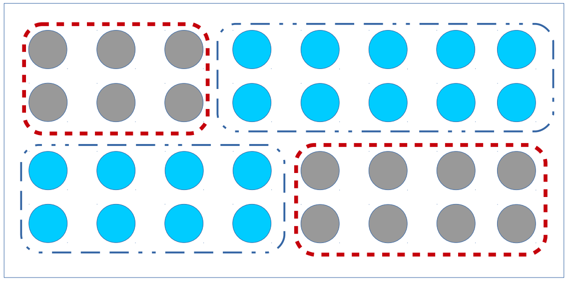
# Problem Definition

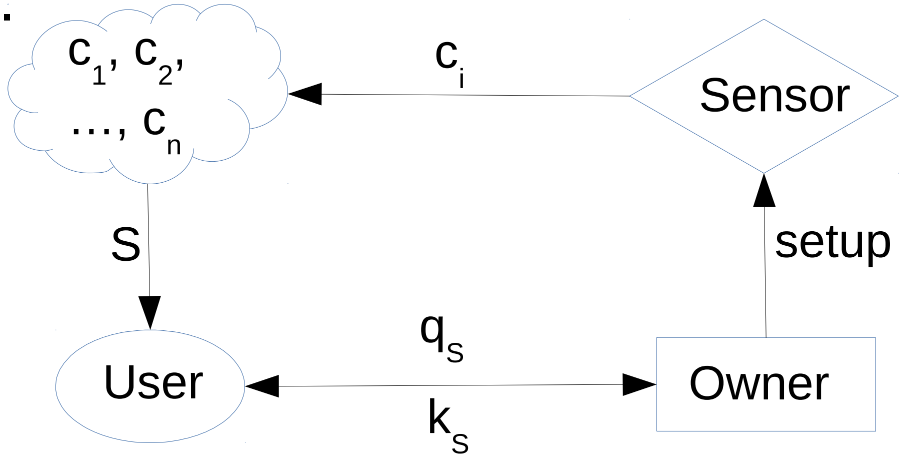
- Security Requirements:
  - Confidentiality of the samples.
  - Collusion resistance.
    - combining multiple aggregated keys could not derive more information than each aggregated key can individually derive
  - Sensors are trusted and independent.



# Problem Definition

- Security Requirements:
  - Confidentiality of the samples.
  - Collusion resistance.
    - combining multiple aggregated keys could not derive more information than each aggregated key can individually derive
  - Sensors are trusted and independent.
- Efficiency Requirements:
  - Low computation load.
  - Low communication overhead.
  - Low storage overhead.





# Our Solution

- Each sample is encrypted individually using unique key to avoid collusion attack.

# Our Solution

- Each sample is encrypted individually using unique key to avoid collusion attack.
- Leverage on KAC to ensure:
  - Aggregating any set of keys into one constant size key, attaining low communication overhead.
  - Low storage overhead by constant size ciphertexts.

# Our Solution

- Each sample is encrypted individually using unique key to avoid collusion attack.
- Leverage on KAC to ensure:
  - Aggregating any set of keys into one constant size key, attaining low communication overhead.
  - Low storage overhead by constant size ciphertexts.
- Propose fast reconstruction techniques to reduce the computation load.
  - Achieving orders of magnitude speed-up over original KAC.

# KAC Reconstruction Review

- Reconstructing a ciphertext with index $\mathrm{i} \in S$ using an aggregated key $k_S$ requires:

$$\rho_i = \prod_{j \in S, j \neq i} g_{n+1+i-j}$$

  where all $g_x$ can be drawn from public parametters and $n$ is system capacity.

- This incurs $O(|S|^2)$ group multiplications to reconstruct all samples in $S$.

# A Key Observation

The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

# A Key Observation

The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$

# A Key Observation

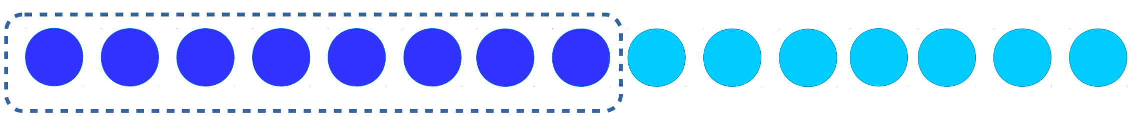
The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$



$\mathrm{X}_1$

# A Key Observation

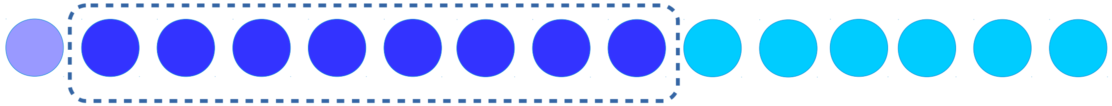
The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$



$\mathrm{X}_2$

# A Key Observation

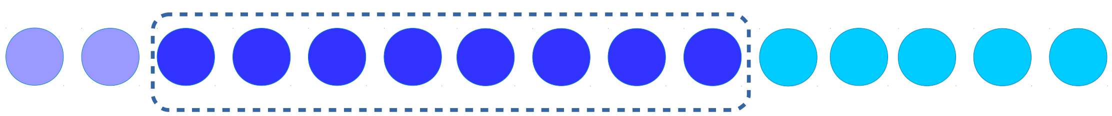
The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$



$\mathrm{X}_3$

# A Key Observation

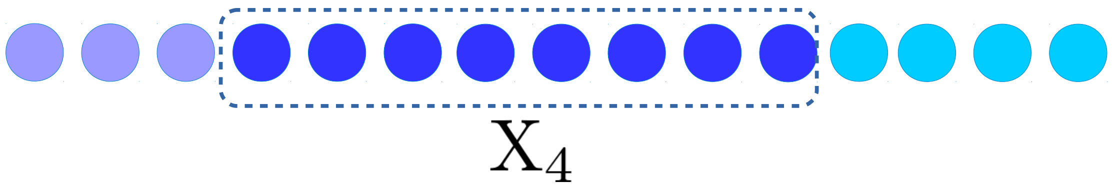
The recurrence relation

$$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\} \text{ where } X_i = \prod_{j=i}^{i+m} p_j$$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$



# A Key Observation

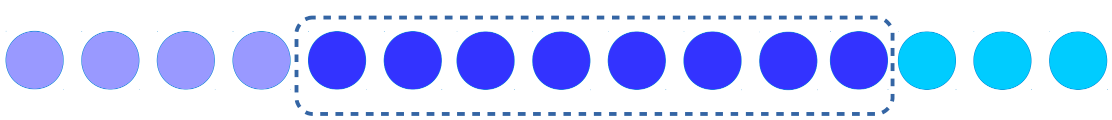
The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$



$\mathrm{X}_5$

# A Key Observation

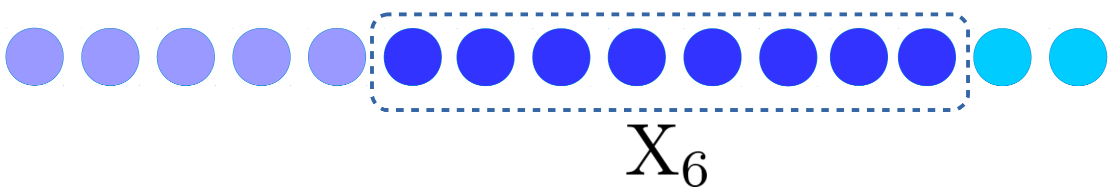
The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$



# A Key Observation

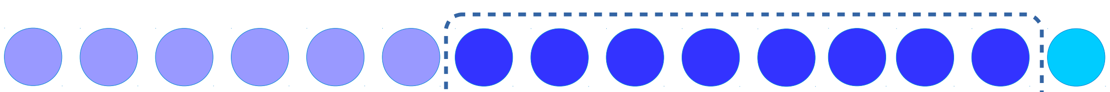
The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$



$\mathrm{X}_7$

# A Key Observation

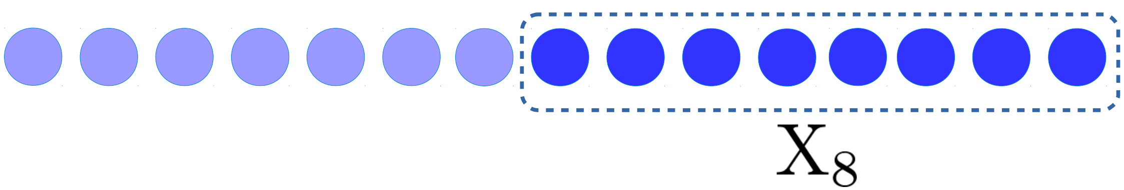
The recurrence relation

$\mathrm{X} = \{\mathrm{X}_1, X_2, \cdots, X_m\}$ where $X_i = \prod_{j=i}^{i+m} p_j$

- How many multiplications to evaluate X?

  ***It is O(m), not O(m²)***

- $X_{i+1} = X_i \times p_{i+1} \times p_i^{-1} \quad \forall i \in [1..m-1]$



# Fast Reconstruction for Q1

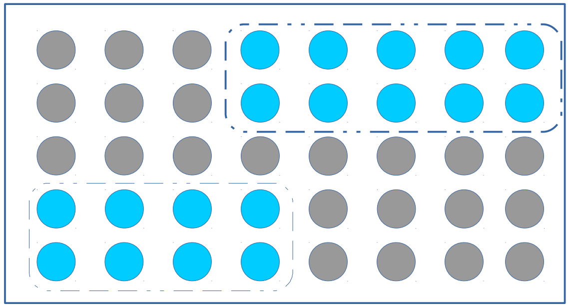
For Q1 with S = [1,m]:

- $\hat{g}_i = g_{n+1+i},\ R_i = \prod_{j \in S} \hat{g}_{i-j} \Longrightarrow \rho_i = \hat{g}_i^{-1} R_i$
- A special recurrence relation:

$$R_{i+1} = (\hat{g}_{i-m})^{-1} \cdot R_i \cdot \hat{g}_i$$

i.e. obtaining $R_{i+1}$ from $R_i$ with two multiplications.

=> In general, reconstructing samples in d-dimensional range query requires only O(d|S|) multiplications; i.e. linear time.



# Fast Reconstruction for Q1

For e.g., with S [1..5], system capacity n = 20:

$$\rho_1 = g_{17} \times \boxed{g_{18} \times g_{19} \times g_{20}}$$
$$\rho_2 = \boxed{g_{18} \times g_{19} \times g_{20}} \times g_{22}$$

# Fast Reconstruction for Q1

For e.g., with S [1..5], system capacity n = 20:

$$
\begin{aligned}
\rho_1 &= g_{17} \times g_{18} \times g_{19} \times g_{20} \\
\rho_2 &= g_{18} \times g_{19} \times g_{20} \times g_{22} \\
\rho_3 &= g_{19} \times g_{20} \times g_{22} \times g_{23}
\end{aligned}
$$

# Fast Reconstruction for Q1

For e.g., with S [1..5], system capacity n = 20:

$$\rho_1 = g_{17} \times g_{18} \times g_{19} \times g_{20}$$
$$\rho_2 = g_{18} \times g_{19} \times g_{20} \times g_{22}$$
$$\rho_3 = g_{19} \times g_{20} \times g_{22} \times g_{23}$$
$$\rho_4 = g_{20} \times g_{22} \times g_{23} \times g_{24}$$

# Fast Reconstruction for Q1

For e.g., with S [1..5], system capacity n = 20:

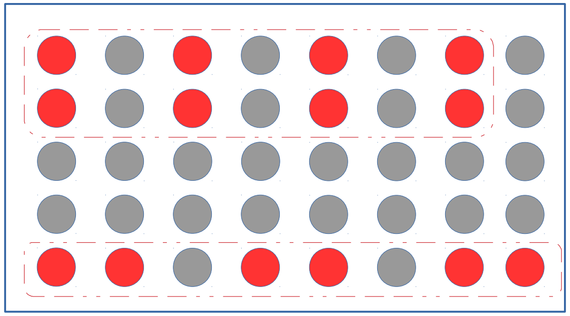
$$
\begin{aligned}
\rho_1 &= g_{17} \times g_{18} \times g_{19} \times g_{20} \\
\rho_2 &= g_{18} \times g_{19} \times g_{20} \times g_{22} \\
\rho_3 &= g_{19} \times g_{20} \times g_{22} \times g_{23} \\
\rho_4 &= g_{20} \times g_{22} \times g_{23} \times g_{24} \\
\rho_5 &= g_{22} \times g_{23} \times g_{24} \times g_{25}
\end{aligned}
$$

# Fast Reconstruction for Q2

Transform and Conquer strategy:

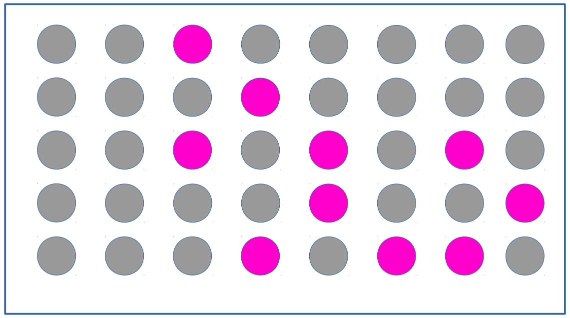
- Transform the coordinate system such that indices of the required samples correspond to integer coordinates.
- Apply the special recurrence relation as in Q1.

=> Also requires only O(d|S|) multiplications; i.e. linear time.



# Fast Reconstruction for Q3

- Samples' indices in Q3 may not have an special structure, to which the special recurrence could not apply.
- Problem transformation:
  - Let $P_i$ be a multi-set comprising of all $g_x$ required to compute $\rho_i$ , T the target collection comprising of all $P_i$.
  - A computation plan to evaluate all $\rho_i$ is equivalent to that of constructing T.



# Fast Reconstruction for Q3

Minimum Spanning Tree based Strategy:

- Define dist(i, j) = $|P_i \backslash P_j| + |P_j \backslash P_i|$
- A computation plan is determined by solving for the MST on a graph G = (V,E):
  - G is complete.
  - V comprises of |T|+1 vertices: Vertex $v_i$ represent a multiset $P_i$, and special vertex $\overline{v}$ represents empty multiset.
  - An edge $e_{ij}$ connecting $v_i$ and $v_j$ has weigh of dist(i,j). All edges orignating from $\overline{v}$ have weight of |T| - 2.

# Fast Reconstruction for Q3

For e.g. S = [2,4,5,7,9], n = 20:

$$\rho_2 = g_{14} \times g_{16} \times g_{18} \times g_{19}$$

$$\rho_4 = g_{16} \times g_{18} \times g_{20} \times g_{23}$$

$$\rho_5 = g_{17} \times g_{19} \times g_{22} \times g_{24}$$

$$\rho_7 = g_{19} \times g_{23} \times g_{24} \times g_{26}$$

$$\rho_9 = g_{23} \times g_{25} \times g_{26} \times g_{28}$$

# Fast Reconstruction for Q3

For e.g. S = [2,4,5,7,9], n = 20:

$$\rho_2 = g_{14} \times g_{16} \times g_{18} \times g_{19}$$

$$\left.\begin{aligned} \rho_4 &= g_{16} \times g_{18} \times g_{20} \times g_{23} \\ \rho_5 &= g_{17} \times g_{19} \times g_{22} \times g_{24} \end{aligned}\right\} \text{dist } (P_4,P_5) = 8$$

$$\left.\begin{aligned} \rho_7 &= g_{19} \times g_{23} \times g_{24} \times g_{26} \\ \rho_9 &= g_{23} \times g_{25} \times g_{26} \times g_{28} \end{aligned}\right\} \text{dist } (P_7,P_9) = 4$$

# Fast Reconstruction for Q3

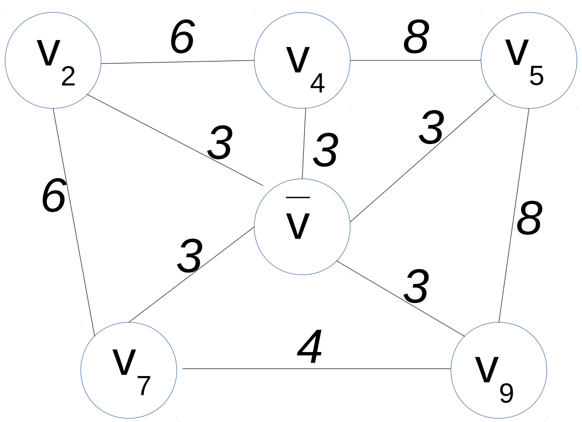
For e.g. S = [2,4,5,7,9], n = 20:

$$\rho_2 = g_{14} \times g_{16} \times g_{18} \times g_{19}$$

$$\left.\begin{aligned}\rho_4 &= g_{16} \times g_{18} \times g_{20} \times g_{23} \\ \rho_5 &= g_{17} \times g_{19} \times g_{22} \times g_{24}\end{aligned}\right\} \text{dist}\,(P_4,P_5) = 8$$

$$\left.\begin{aligned}\rho_7 &= g_{19} \times g_{23} \times g_{24} \times g_{26} \\ \rho_9 &= g_{23} \times g_{25} \times g_{26} \times g_{28}\end{aligned}\right\} \text{dist}\,(P_7,P_9) = 4$$

T = $\{P_2,P_4,P_5,P_7,P_9\}$




*Some edges in the above graph are ignored for visual clarity.

# Fast Reconstruction for Q3

Even better computation plan can be achieved by:

- Finding a *minimum-weight Steiner tree* on G
  - Introduce intermediate vertices; i.e. intermediate values.
- Trade-off between number of aggregated keys and reconstruction time:
  - Split S into several subqueries, issuing one key for each query.
  - The splitting is done using *single-linkage clustering* method.
  - The distance betwee two “clusters” $S_a$ and $S_b$ are total number of multiplications required to reconstruct samples in the union cluster $S_a \cup S_b$.

# Experiments

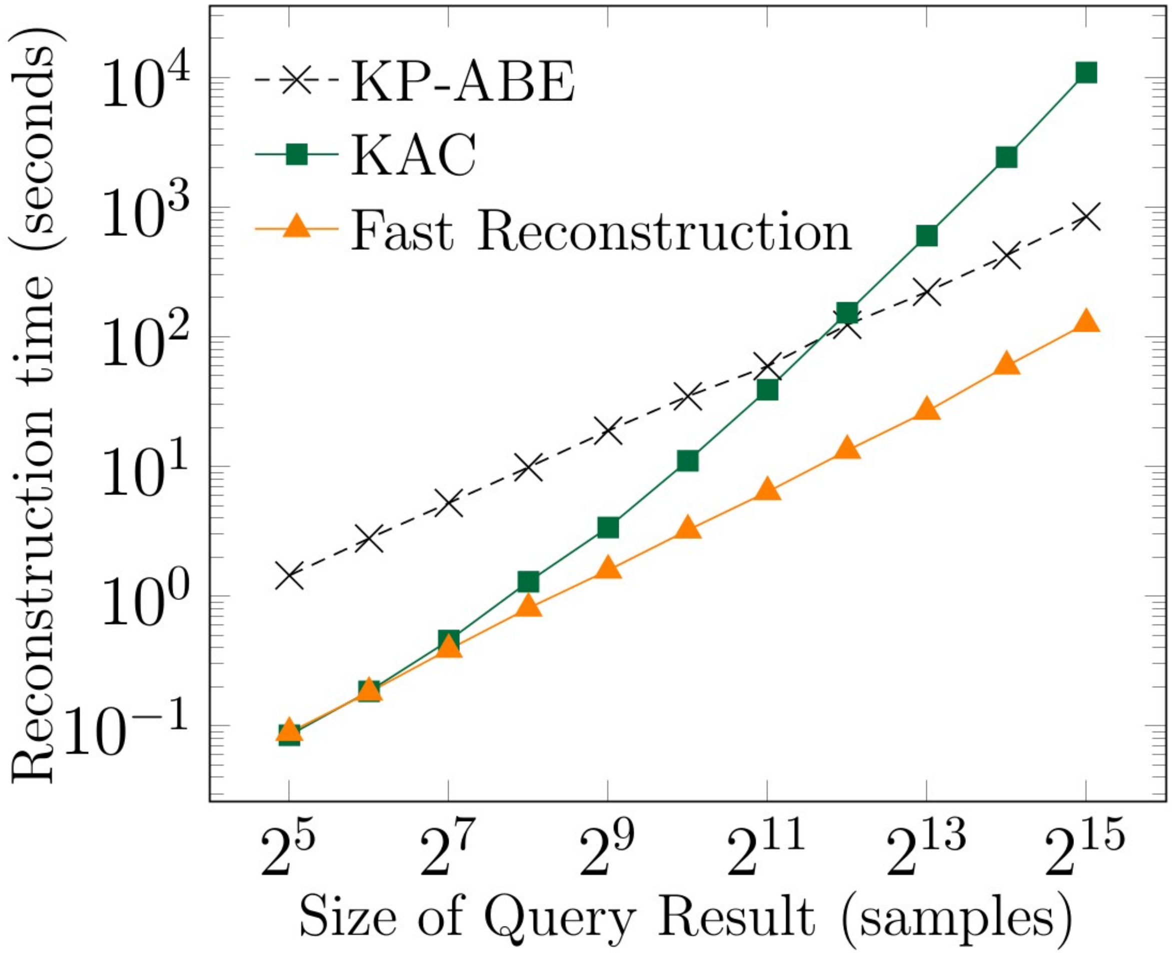
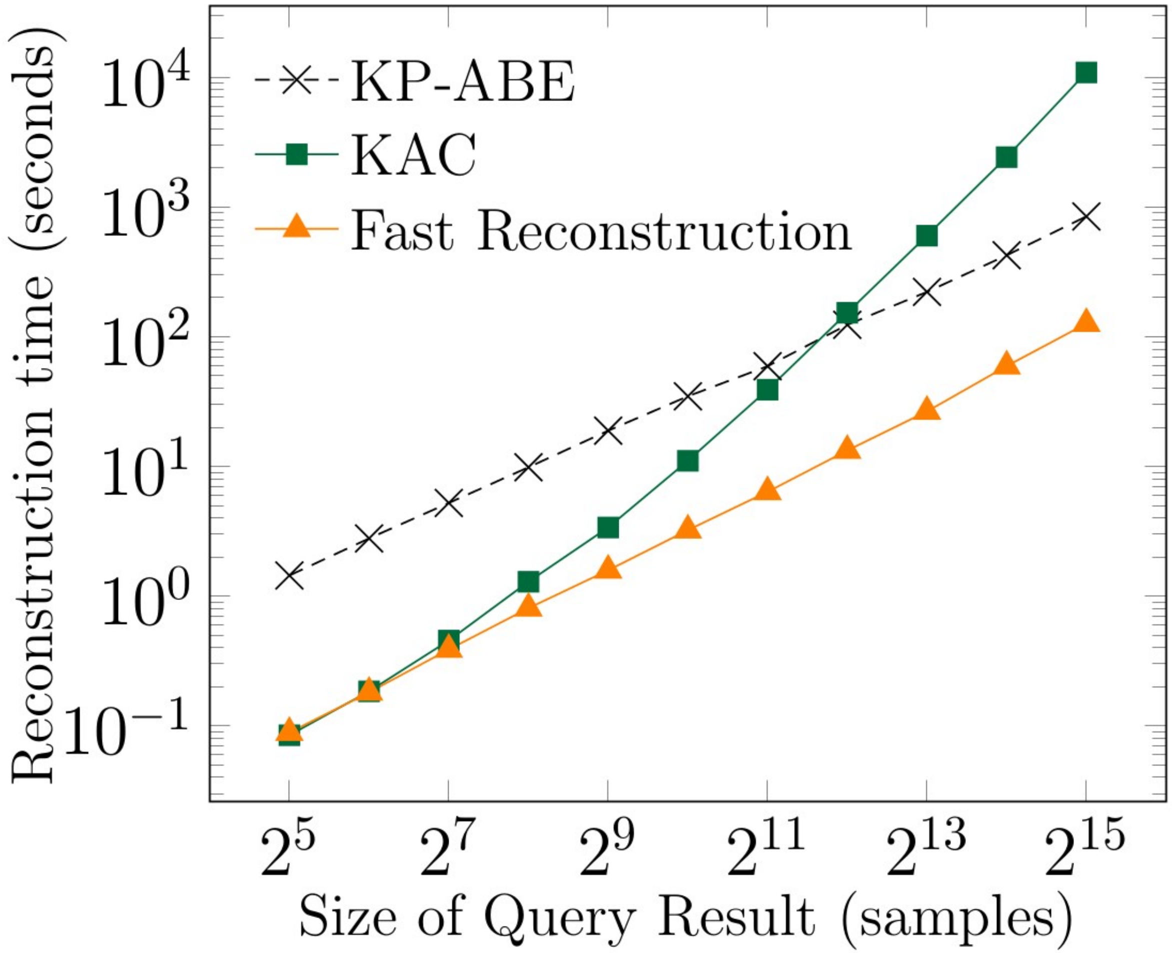



Figure 1: Reconstruction time for Q1 & Q2.

# Experiments




**90x speedups**

Figure 1: Reconstruction time for Q1 & Q2.

# Experiments

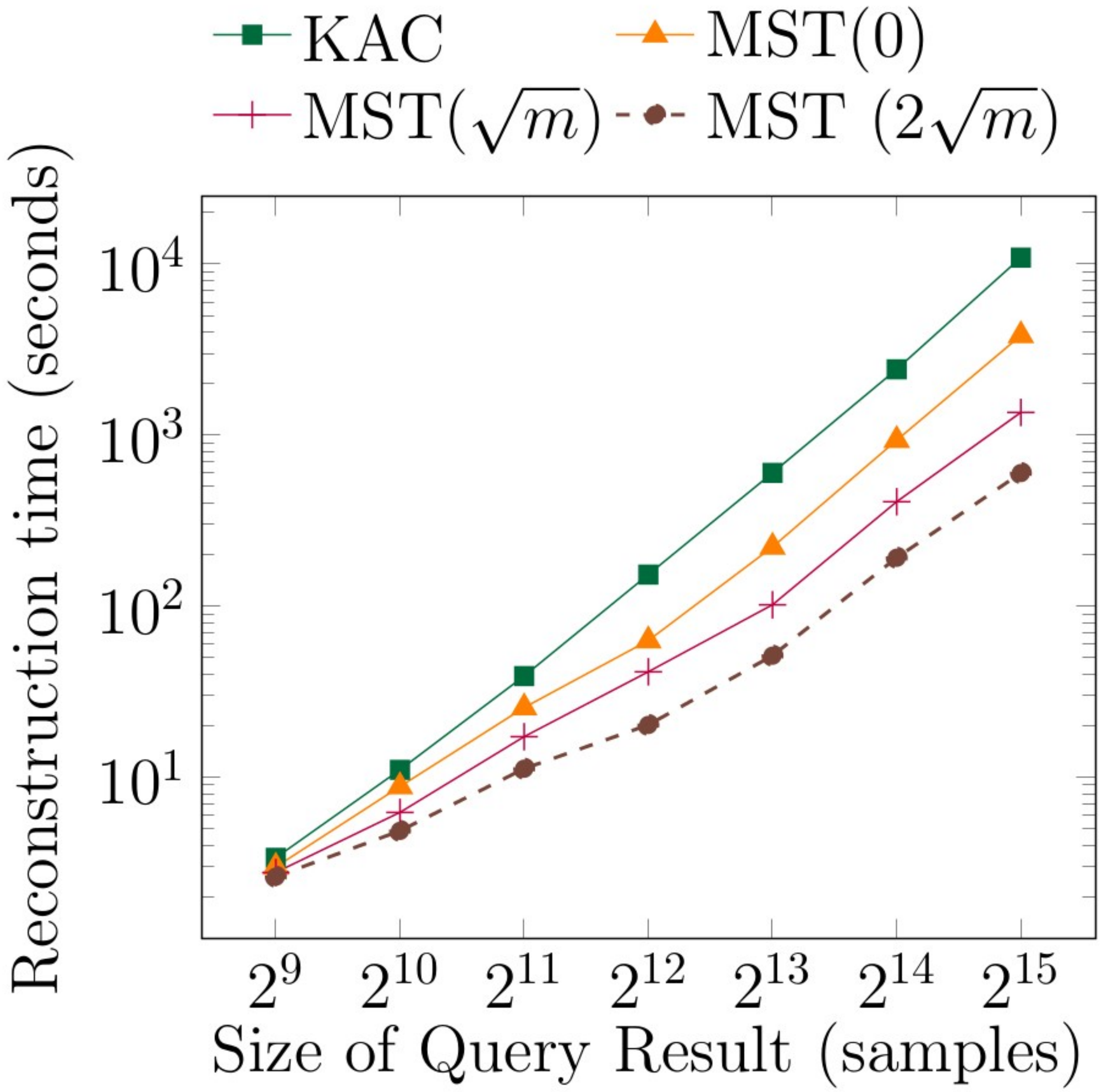
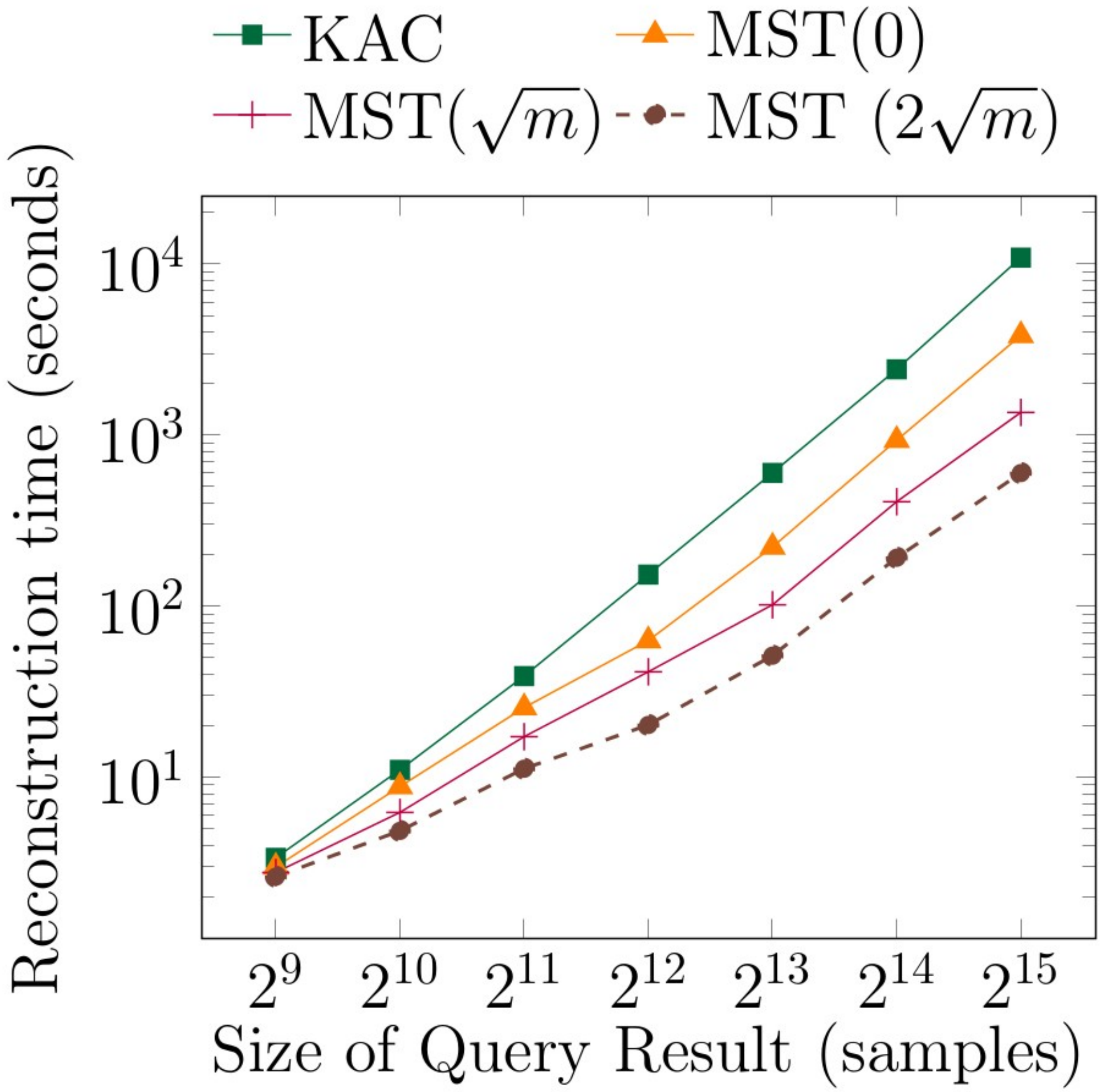



Figure 2: Reconstruction time for Q3. MST(o) indicates the computation plan constructed with o intermediate values. m is the size of query result.

# Experiments




**8x speedups**

Figure 2: Reconstruction time for Q3. MST(o) indicates the computation plan constructed with o intermediate values. m is the size of query result.

# Experiments

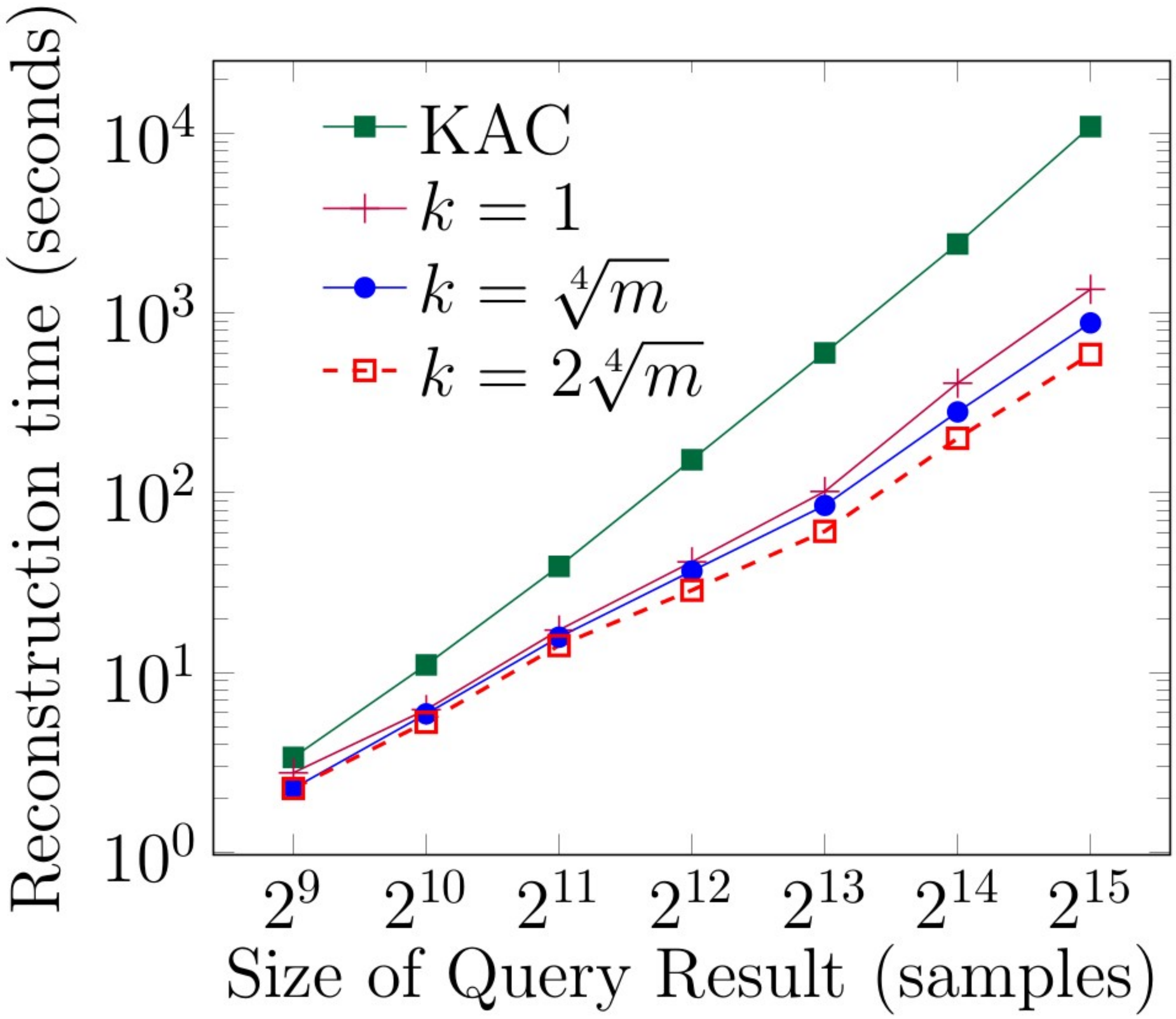
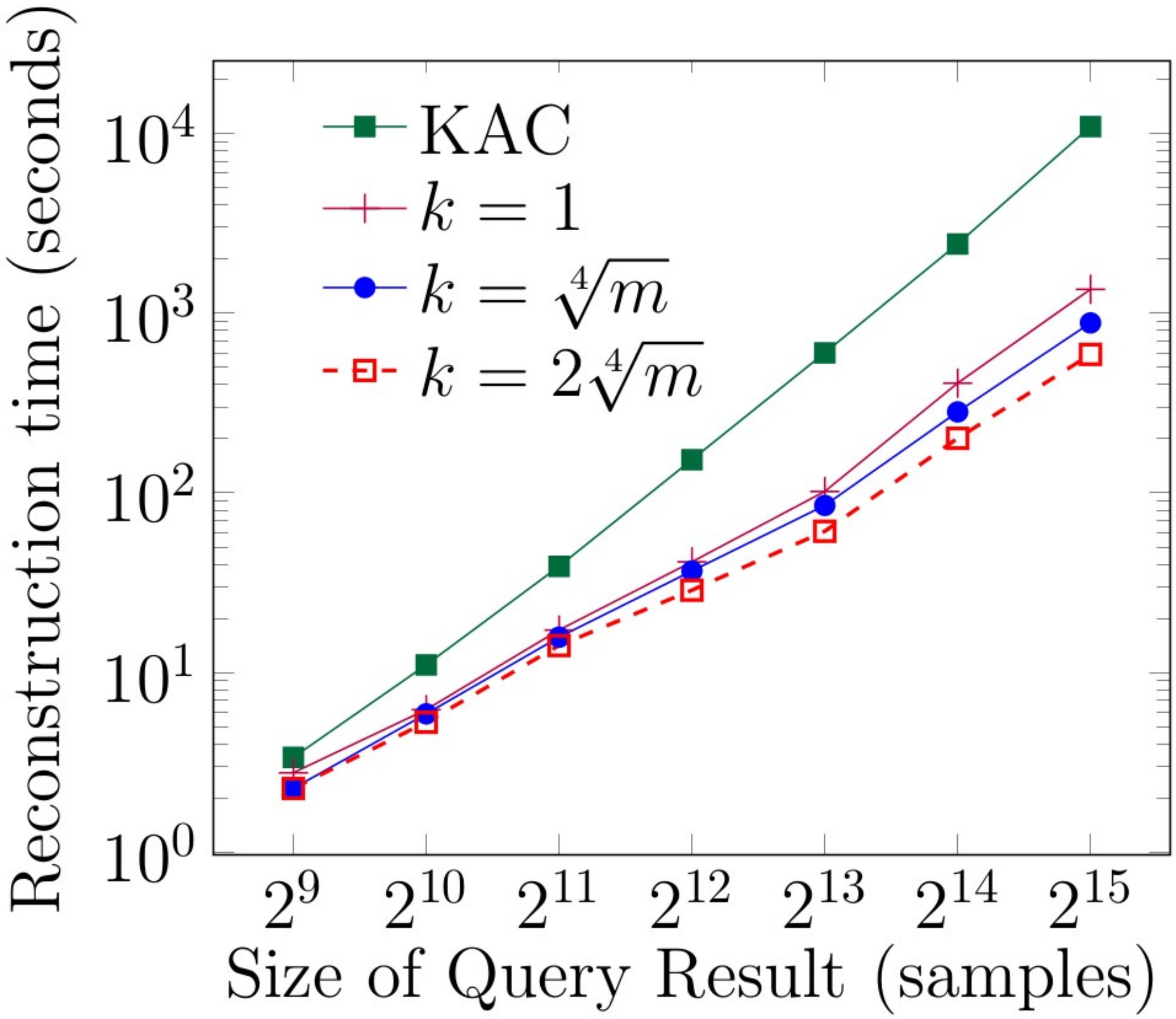



Figure 3: Trade-off between number of aggregated keys and reconstruction time for Q3.
k is number of sub-queries, m is the size of query result.

# Experiments




**19x speedups by splitting into 16 sub-queries.**

Figure 3: Trade-off between number of aggregated keys and reconstruction time for Q3.
k is number of sub-queries, m is the size of query result.

# Related Works

- Key sharing with hierarchical structures (e.g. trees) (Tzeng '02, Benaloh '09, Atallah '09)
  - Not applicable for multi-dimensional data not following hierarchical structure.
- Key Policy – Attribute based Encryption (Chase '06, Hohenberger '08, Lewko '09)
  - Prohibitive performance overhead.
- Complex queries over encrypted data (Boneh '07, Shi '07)
  - Irrelevant security requirement (e.g. secrecy of all attributes).
- KAC follow-ups (Tong '13, Deng '14)
  - Did not address the fast reconstruction techniques.

# Conclusions

- Fast reconstruction techniques for KAC enables scalable sharings of sensitive data.
- Our observation is also applicable to other cryptographic primitives involving group multiplications such as broadcast encryption and redactable signatures.

# Q & A

Hung Dang
hungdang@comp.nus.edu.sg