Secure Data Retrieval on the Cloud: Homomorphic Encryption meets Coresets

Adi Akavia (University of Haifa), Dan Feldman (University of Haifa), Hayim Shaul (University of Haifa)

CHES `19
Motivation

- Useful building block - many applications
- Shows link between secure computation and coresets
Motivation

Many algorithms follow these lines:

Input: \( n \) items \((d_1, ..., d_n)\)
Find: items that match a filter
Report: those items

\[ \text{IsMatch}(d_i, q) = x_i \in \{0, 1\} \]
Problem - Efficient w.r.t. communication

Input: \( n \) items \( (d_1,...,d_n) \)
Query: a filter \( \text{IsMatch}(.,q) \)
Report: All indices \( i \) such that \( x_i = 1 \)

\( \text{IsMatch}(d_i,q) = x_i \in \{0,1\} \)

Easy to extend: report \( d_i \) s.t. \( x_i=1 \)

Many indices - report all. We therefore assume at most \( s << n \) matches

We want: comm. complexity = function of \( s \)
Additive/Fully Homomorphic Encryption
Fully Homomorphic Encryption (FHE)

Public key encryption scheme.

\[ \text{Enc}(x, pk) = [x] \]
\[ \text{Dec}([x], sk) = x \]

\[ \text{Dec( Add}([x], [y]) \text{) } = x+y \]
\[ \text{Dec( Mul}([x], [y]) \text{) } = xy \]

\[ [x] + [y] ; [x] + y \]
\[ [x][y] ; [x]y = [x] + [x] + [x] + ... \]
Any algorithm can be implemented

Any polynomial can be evaluated with FHE

Any algorithm can be expressed as a polynomial of the input

Objective: keep the degree small
### Our Results

<table>
<thead>
<tr>
<th></th>
<th>Our Results</th>
<th>Direct Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report all $s$ matches</td>
<td><strong>Degree</strong>: $d$  &lt;br&gt; <strong>Comm</strong>: $O(s^2 \log^2 n)$  &lt;br&gt; <strong>Client</strong>: $(s \log n)^{O(1)}$</td>
<td><strong>Degree</strong>: $O(d \cdot n)$  &lt;br&gt; <strong>Comm</strong>: $O(s \log n)$  &lt;br&gt; <strong>Client</strong>: $O(s \log n)$</td>
</tr>
</tbody>
</table>

$d = \text{degree(isMatch)}$
Example: Report all DD < 1 mile away

Input: Dunkin store gps \((d_1, \ldots, d_n)\)
Query: \([location]\)

\[\begin{align*}
x_i &= \text{isMatch}(d_i, [location]) \\
\text{dist}(d_i, [location]) &< 1\text{mile}
\end{align*}\]

Report \(i\) s.t. \(x_i=1\)

A Dunkin service to find the nearest store
Without telling where you are.
Without downloading the entire database.

\(n = \text{Gazillion}\)
\(s < 10\)
Direct Approach

**Input:**

binary \((x_1, \ldots, x_n)\) with at most \(s\) 1’s

**Output:**

Output[1] - index of 1\(^{st}\) 1 in \((x_1, \ldots, x_n)\)
Output[2] - index of 2\(^{nd}\) 1 in \((x_1, \ldots, x_n)\)
...
Output[s] - index of s\(^{th}\) 1 in \((x_1, \ldots, x_n)\)
Direct Approach

\[ \text{Output}[t] = \sum_{j=1}^{n} j \cdot x_j \cdot \text{isEqual}(x_1+x_2+...+x_{j-1}, t-1) \]

\text{isEqual}(a,b) = \text{returns 1 if } a=b, 0 \text{ otherwise.}

Tests if there are \((t-1)\) matches in \(x_1, \ldots, x_{j-1}\)

Using Fermat’s Little Theorem:

\[ \text{isEqual}(a,b) = 1 - (a-b)^{p-1} \mod p \]

Since \(p > n\) the degree is \(\Theta(n)\)
Coresets for FHE

“Borrowed” from computational geometry: $C$ is a coreset of $P$ if:
1. $C$ is short
2. $P := \text{Decode}(C)$ is efficient

We will transform $(x_1, \ldots, x_n)$ to a different representation to improve performance.
Indyk-Ngo-Rudra (2010) Sketch

A \((s,n)\) sketch matrix

\[ S \in \{0,1\}^{k \times n} \]

transforms a long vector \(x \in \{0,1\}^n\) with at most \(s\) 1’s into a short vector \(y = S \cdot x \in \{0,...,s\}^k\) s.t.

there exists Decode alg., where \(x = \text{Decode}(y)\).
Example (1,7) Sketch Matrix

\[
S = \begin{pmatrix}
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
0 \\
0 \\
1 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

Because multiplying by a 1-sparse vector \( x \in \{0,1\}^7 \) with 1 at the \( i \)-th place gives the \( i \)-th column of \( S \) which is the binary rep. of \( i \).

Decode: parse binary value.
Indyk-Ngo-Rudra (2010):

For every $s,n$ exists a $(s,n)$-sketch matrix $S \in \{0,1\}^{k \times n}$

With

\[ k = O(s^2 \log n) \]

and decode time

\[ \text{Poly}(k) \]
Coresets for Report

\[ [x] = ( [0], [1], ..., [0] ) \]

\[ S[x] \]  \( \rightarrow \)  \( (0, 1, ..., 0) \)  

Decrypt

Decode
Polynomial Degree Analysis

Since $S \in \{0,1\}^{k \times n}$ is clear text, multiplying $S[x]$ can be done by adding elements of $x$.

The Degree is therefore 1. - Additive HE is enough.
Experimental Results

- HElib
- 64 cores
Conclusion

- Using coresets we can improve performance
- Report a $s$ sparse vector of size $n$ requires only additive HE

Open Problems

- More coreset applications
- Improve constants
Thank You