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

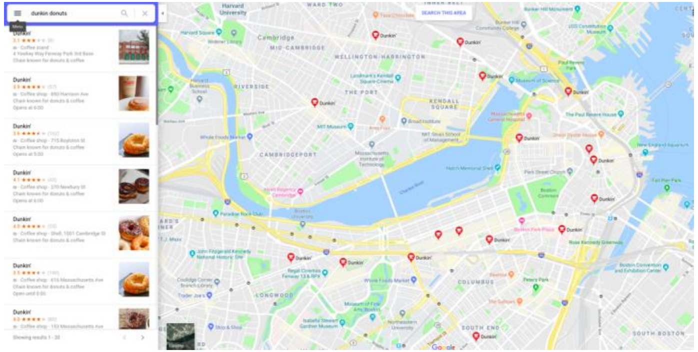
Many algorithms follow these lines:

Input: n items $(d_1,\ldots,d_n)$

✓ Find: items that match a filter

? Report: those items

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



Problem - Efficient w.r.t. communication

Input: n items $(d_1,\dots,d_n)$
Query: a filter $IsMatch(\cdot,q)$
Report: All indices i such that $x_i = 1$

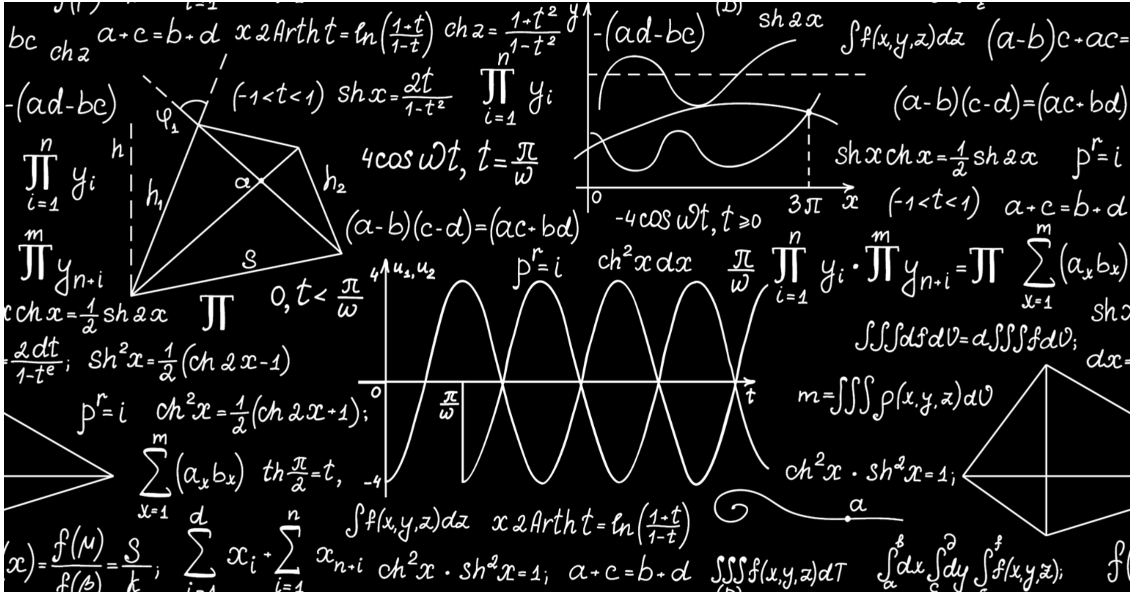
$IsMatch(d_i,q) = x_i \in \{0,1\}$

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.

Enc*(x, pk) =* $[x]$

Dec*([x], sk) = x*

Dec(***Add****([x], [y])) = x+y* $[x]+[y]$; $[x]+y$

Dec(***Mul****([x], [y])) = xy* $[x][y]$; $[x]y$ $= [x]+[x]+[x]+\ldots$

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

	Our Results	Direct Approach
Report all s matches	**Degree:** d **Comm:** $O(s^2 \log^2 n)$ **Client:** $(s \log n)^{O(1)}$	**Degree:** $O(d\ n)$ **Comm:** $O(s \log n)$ **Client:** $O(s \log n)$

d = degree(isMatch)

Example: Report all DD <1 mile away

Input: Dunkin store gps $(d_1, \ldots, d_n)$

Query: *[location]*

$x_i = isMatch(d_i, [location])$

$\quad dist(d_i, [location]) < 1mile$

Report *i* s.t. $x_i=1$

n = Gazillion

s < 10

A Dunkin service to find the nearest store

Without telling where you are.

Without downloading the entire database.

Direct Approach

Input:

binary $(x_1, \dots, x_n)$ with at most s 1's

Output:

Output[1] - index of 1st 1 in $(x_1, \dots, x_n)$

Output[2] - index of 2nd 1 in $(x_1, \dots, x_n)$

…

Output[s] - index of sth 1 in $(x_1, \dots, x_n)$

Direct Approach

(1,0,0,…, 1,0,0,1,0,0,1…)

$Output[t] = \Sigma^n_{j=1} \; j \cdot x_j \cdot isEqual(x_1+x_2+\ldots+x_{j-1}, t-1)$

isEqual(a,b) = returns 1 if $a=b$, 0 otherwise.

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

Using Fermat's Little Theorem:

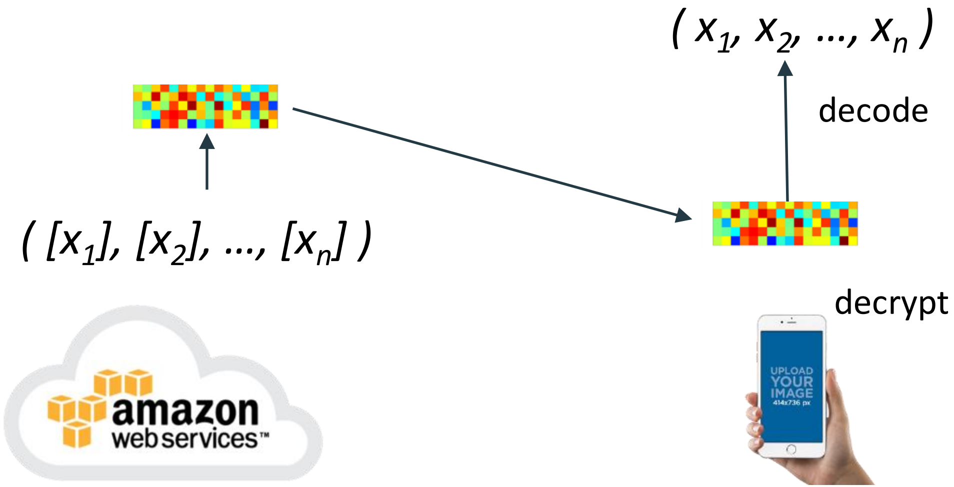
$isEqual(a,b) = 1 - (a-b)^{p-1} \bmod p$

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) $P := Decode(C)$ 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**

$$S \in \{0,1\}^{k x n}$$

transforms a long vector

$$x \in \{0,1\}^n$$ with at most s 1's

into a short vector

$$y = S \cdot x \in \{0,\ldots,s\}^k$$ s.t.

there exists Decode alg., where $x = Decode(y)$.

Example (1,7) Sketch Matrix

$$S = \begin{pmatrix} 0 & 0 & \boxed{0} & 1 & 1 & 1 & 1 \\ 0 & 1 & \boxed{1} & 0 & 0 & 1 & 1 \\ 1 & 0 & \boxed{1} & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Because multiplying by a 1-sparse vector $x\epsilon\{0,1\}^7$ 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 $S \epsilon \{0,1\}^{k x n}$

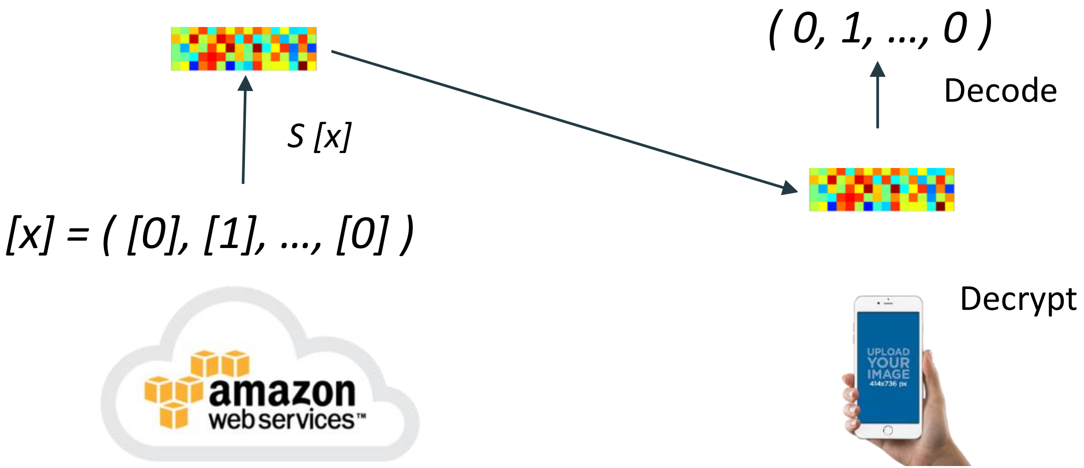
With

$k=O(s^2 \log n)$

and decode time

$Poly(k)$

Coresets for Report



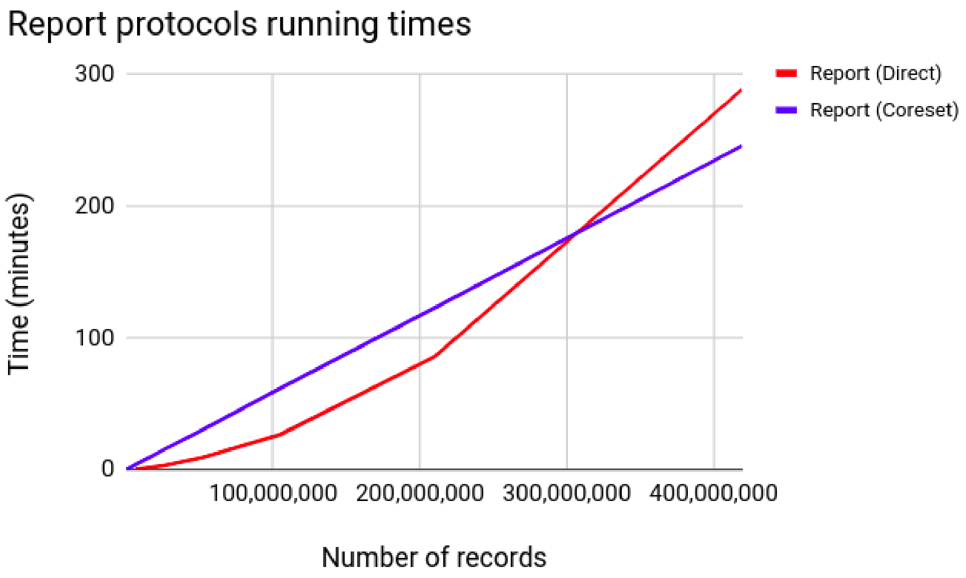
Polynomial Degree Analysis

Since $S \epsilon \{0,1\}^{kxn}$ 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