Homomorphic Matrix Computation & Application to Neural Networks

Xiaoqian Jiang, Miran Kim (University of Texas, Health Science Center at Houston)

Kristin Lauter (Microsoft Research), **Yongsoo Song** (University of California, San Diego)

Background

Primitives for Secure Computation

- Differential Privacy
  - Limited Applications (e.g. count, average). Privacy budget.
- (Secure) Multi-Party Computation
  - Lower complexity, but higher communication costs.

  e.g. 40GB for GWAS analysis for 100K individuals [Nature Biotechnology'17]
  - Protocol has many rounds.
- (Fully) Homomorphic Encryption
  - Higher complexity, but less communication costs.
  - One round protocol.

HE vs. MPC

	Homomorphic Encryption	Multi-Party Computation
Re-usability	High (non-interactive) One-time encryption No further interaction from the data owners	Single-use encryption Not good for long-term storage Interaction between parties each time
Sources	Unlimited	Limited participants (due to complexity constraints)
Speed	Slow in computation (but can speed-up using SIMD)	Slow in communication (due to large circuit to be exchanged)

HE is ideal for long term storage and non-interactive computation

Summary of Progresses

- 2009-10: Plausibility
  - [GH'11] A single bit operation takes 30 minutes.
- 2011-12: Real Circuits
  - [GHS'12] A 30,000-gate in 36 hours
- 2013-16: Usability
  - HElib [HS'14]: IBM's open-source implementation of the BGV scheme

    The same 30,000-gate in 4-15 minutes
- 2017-Today: Practical uses for real-world applications
  - HE Standardization workshops
  - iDASH Privacy & Security competition (2013~)

Secure Health Data Analysis

- Predicting Heart Attack
  - ~0.2 seconds.
- Sequence matching
  - ~27 seconds, Edit distance of length 8.
  - ~180 seconds, Approximate edit distance of length 10K (iDASH'15)
- Searching of Biomarkers
  - ~0.2 seconds, 100K database (iDASH'16)
- Training Logistic Regression Model
  - ~7 minutes, 18 features * 1600 samples (iDASH'17)

Homomorphic Matrix Operation

- HElib (Crypto'14)
  - (Matrix) * (Vector)
- CryptoNets (ICML'16)
  - (Plain matrix) * (Element-wisely encrypted vector)
- GAZELLE (Usenix Security'18)
  - (Column-wisely encrypted matrix) * (Plain vector)
- Homomorphic Evaluation of (Deep) Neural Networks
  - [BMMP17] Evaluation of discretized DNN, [CWM+17] Classification on DNN.
  - Evaluation of Plain model on Encrypted data.

Homomorphic Matrix Operation

- HElib (Crypto'14)
  - (Matrix) * (Vector)

O(d) complexity for (matrix*vector).
→ $O(d^2)$ for (matrix*matrix): not optimal.

- CryptoNets (ICML'16)
  - (Plain matrix) * (Element-wisely encrypted vector)
- GAZELLE (Usenix Security'18)
  - (Column-wisely encrypted matrix) * (Plain vector)
- Homomorphic Evaluation of (Deep) Neural Networks
  - [BMMP17] Evaluation of discretized DNN, [CWM+17] Classification on DNN.
  - Evaluation of Plain model on Encrypted data.

Motivation

- Scenarios (Data/Model owner; Cloud server; Individuals)
  I. Data owner trains a model and makes it available on the cloud.
  II. Model provider encrypts a trained model & uploads it to the cloud to make predictions on encrypted inputs from individuals.
  III. Cloud trains a model on encrypted data and uses it to make predictions on new encrypted inputs.

- Our Work: Homomorphic Operations between Encrypted Matrices

Main Idea

Functionality of HE Schemes

- Packing Method
  - Vector encryption & Parallel operations.
  - $Enc(x_1,\ldots, x_n) * Enc(y_1,\ldots, y_n) = Enc(x_1 * y_1,\ldots, x_n * y_n)$

Functionality of HE Schemes

- Packing Method
  - Vector encryption & Parallel operations.
  - $Enc(x_1,\ldots, x_n) * Enc(y_1,\ldots, y_n) = Enc(x_1 * y_1,\ldots, x_n * y_n)$
- Scalar Multiplication
  - $(a_1,..,a_n) * Enc(x_1,.., x_n) = Enc(a_1x_1,.., a_nx_n)$
- Rotation
  - $Enc(x_1,\ldots, x_n) \rightarrow Enc(x_2,\ldots, x_n, x_1)$

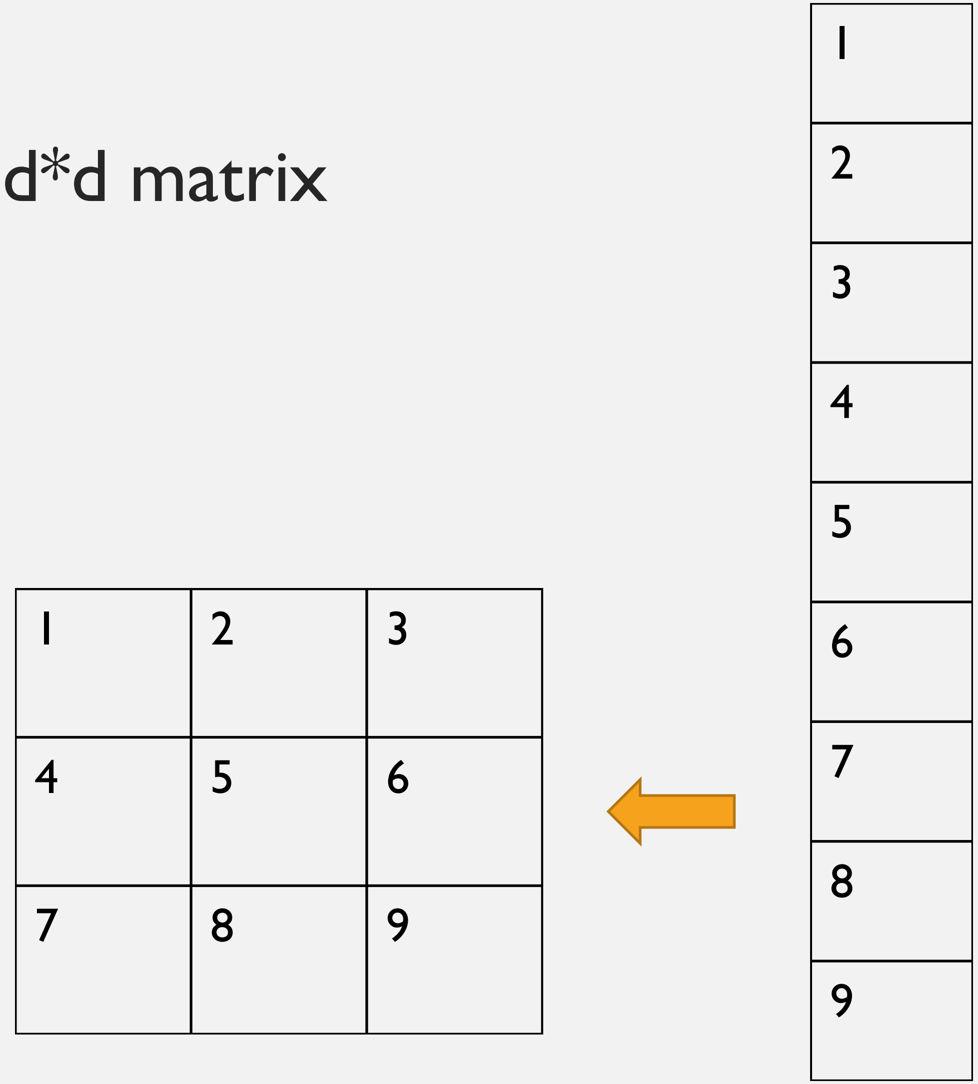
Functionality of HE Schemes

- Packing Method
  - Vector encryption & Parallel operations.
  - $Enc(x_1,\ldots, x_n) * Enc(y_1,\ldots, y_n) = Enc(x_1 * y_1,\ldots, x_n * y_n)$
- Scalar Multiplication
  - $(a_1,..,a_n) * Enc(x_1,.., x_n) = Enc(a_1x_1,.., a_nx_n)$
- Rotation
  - $Enc(x_1,\ldots, x_n) \rightarrow Enc(x_2,\ldots, x_n, x_1)$
- Composition of basic operations
  - Permutation, linear transformation (Expensive)

How to Represent Matrix Arithmetic Using HE-Friendly Operations?

Matrix Encoding

- Identify $n=d^2$ dimensional vector to $d*d$ matrix
  - Addition is easy.



Matrix Encoding

- Identify n=d^2 dimensional vector to d*d matrix
  - Addition is easy.
  - Row or Column shifting permutations are cheap.

    (Depth 1, Complexity O(1))

1	2	3
4	5	6
7	8	9

4	5	6
7	8	9
1	2	3

Matrix Encoding

- Identify $n=d^2$ dimensional vector to d*d matrix
  - Addition is easy.
  - Row or Column shifting permutations are cheap.

    (Depth 1, Complexity O(1))

1	2	3
4	5	6
7	8	9

2	3	1
5	6	4
8	9	7

Matrix Multiplication

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, B = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$AB = A_0 \odot B_0 + A_1 \odot B_1 + A_2 \odot B_2$$

$\odot$: Element-wise Multiplication

$$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 4 \\ 9 & 7 & 8 \end{bmatrix} \odot \begin{bmatrix} a & e & i \\ d & h & c \\ g & b & f \end{bmatrix} + \begin{bmatrix} 2 & 3 & 1 \\ 6 & 4 & 5 \\ 7 & 8 & 9 \end{bmatrix} \odot \begin{bmatrix} d & h & c \\ g & b & f \\ a & e & i \end{bmatrix} + \begin{bmatrix} 3 & 1 & 2 \\ 4 & 5 & 6 \\ 8 & 9 & 7 \end{bmatrix} \odot \begin{bmatrix} g & b & f \\ a & e & i \\ d & h & c \end{bmatrix}$$

Matrix Mult = Generation of Ai, Bi & d-homomorphic add/mult.

Matrix Multiplication

A =

1	2	3
4	5	6
7	8	9

, B =

a	b	c
d	e	f
g	h	i

$$AB = A_0 \odot B_0 + A_1 \odot B_1 + A_2 \odot B_2$$

$\odot$: Element-wise Multiplication

1	2	3
5	6	4
9	7	8

$\odot$

a	e	i
d	h	c
g	b	f

\+

2	3	1
6	4	5
7	8	9

$\odot$

d	h	c
g	b	f
a	e	i

\+

3	1	2
4	5	6
8	9	7

$\odot$

g	b	f
a	e	i
d	h	c

Matrix Mult = Generation of Ai, Bi & d-homomorphic add/mult.

Matrix Multiplication

$A =$

1	2	3
4	5	6
7	8	9

$, B =$

a	b	c
d	e	f
g	h	i

$$AB = A_0 \odot B_0 + A_1 \odot B_1 + A_2 \odot B_2$$

$\odot$: Element-wise Multiplication

1	2	3
5	6	4
9	7	8

$\odot$

a	e	i
d	h	c
g	b	f

$+$

2	3	1
6	4	5
7	8	9

$\odot$

d	h	c
g	b	f
a	e	i

$+$

3	1	2
4	5	6
8	9	7

$\odot$

g	b	f
a	e	i
d	h	c

Matrix Mult = Generation of A_i, B_i & d-homomorphic add/mult.

Generation of A_i

$A =$

1	2	3
4	5	6
7	8	9

$A_0 =$

1	2	3
5	6	4
9	7	8

$A_1 =$

2	3	1
6	4	5
7	8	9

$A_2 =$

3	1	2
4	5	6
8	9	7

A_0 Generation : $O(d)$ homomorphic operations.

A_i = ColumnShifting(A_0, i) : $O(1)$ for each.

Generation of B_i

$B =$

a	b	c
d	e	f
g	h	i

$B_0 =$

a	e	i
d	h	c
g	b	f

$B_1 =$

d	h	c
g	b	f
a	e	i

$B_2 =$

g	b	f
a	e	i
d	h	c

B_0 Generation : O(d) homomorphic operations.

B_i = RowShifting(B_0, i) : O(1) for each.

Summary

A, B : d * d matrices

AB = $A_0 \circledcirc B_0 + A_1 \circledcirc B_1 + \ldots + A_{d-1} \circledcirc B_{d-1}$

Generation of A_0, B_0 : General permutation - O(d).

Generation of A_i, B_i 's: Column/Row shifting from A_0, B_0 - O(d).

Element-wise product and summation: O(d).

Total complexity: O(d) homomorphic operations (optimal?).

Depth: 2 (scalar mult) + 1 (homo mult).

Other Operations

- Matrix Transposition
  - Complexity $O(d^{0.5})$ + Depth 1.
- Parallelization
  - When the number of plaintext slots > d^2.
  - Encrypt several matrices in a single ciphertext.
- Multiplication between Non-square Matrices

Implementation

Experimental Results

Dim	Throughput	Message size	Expansion rate	Encoding+ Encryption	Decoding+ Decryption	Relative time per matrix		
						HE-MatAdd	HE-MatMult	HE-MatTrans
4	1	0.47 KB	3670	34 ms	9 ms	0.62 ms	779 ms	363 ms
	16	0.75 KB	229	41 ms	12 ms	0.05 ms	47 ms	18 ms
	256	12.0 KB	14.3	95 ms	81 ms	0.03 ms	3 ms	1 ms
16	1	0.75 KB	229	33 ms	13 ms	0.62 ms	2501 ms	847 ms
	4	3.0 KB	57.3	48 ms	27 ms	0.19 ms	649 ms	211 ms
	16	12.0 KB	14.3	97 ms	78 ms	0.04 ms	162 ms	49 ms
64	1	12.0 KB	14.3	108 ms	76 ms	0.62 ms	9208 ms	2557 ms

Based on the HEAAN library for fixed-point operation ($n = 2^{13}$).
All numbers have 24-bit precision.

Evaluation of Neural Networks

	Stage	Latency (s)	Relative time per image (ms)
Data owner	Encoding + Encryption	1.56	24.42
Model provider	Encoding + Encryption	12.33	-
Cloud	Convolution	5.68	88.75
	1^{st} square	0.10	1.51
	FC-1	20.79	324.85
	2^{nd} square	0.06	0.96
	FC-2	1.97	30.70
	Total	**28.59**	**446.77**
Authority	Decoding + Decryption	0.07	1.14

1 Convolution layer + 2 Fully connected layers.

Parallel evaluation on 64 images.

Comparison?

Framework	Method	Runtime (s)				Communication (MB)			
		Offline	Online	Total	Amortized	Offline	Online	Total	Per instance
CryptoNets	HE	-	-	570	0.07	-	-	595.5	0.07
MiniONN	HE, MPC	0.88	0.40	1.28	1.28	3.6	44	47.6	47.6
GAZELLE	HE, MPC	0	0.03	0.03	0.03	0	0.5	0.5	0.5
E2DM	HE	-		28.59	0.45	-	-	17.48	0.27

Plain model (previous work) vs. Encrypted model (ours)

Questions?

Thanks for listening