Conference on Applied Machine Learning for Information Security (CAMLIS) 2024

October 24-25, 2024, Arlington, VA

Curl: Private LLMs through Wavelet-Encoded Look-Up Tables

Manuel B. Santos[1], **Dimitris Mouris**[1], Mehmet Ugurbil[1], Stanislaw Jarecki[1,2], José Reis[1], Shubho Sengupta[3] and Miguel de Vega[1]

{ manuel.santos, dimitris, memo, stanislaw.jarecki, jose.reis, miguel }@nillion.com

ssengupta@meta.com

https://ia.cr/2024/1127

https://github.com/jimouris/curl

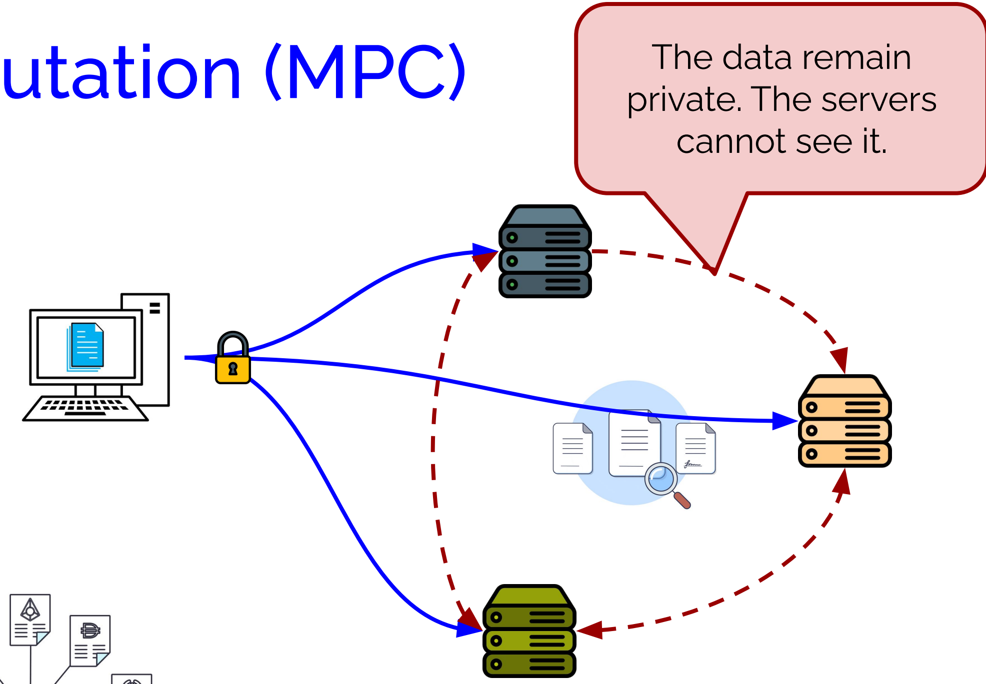
1 nillion

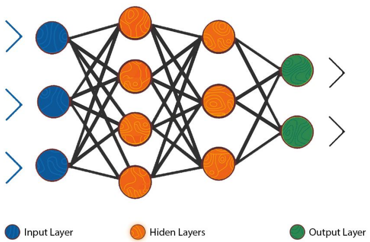
2 The University of California Irvine

3 Meta

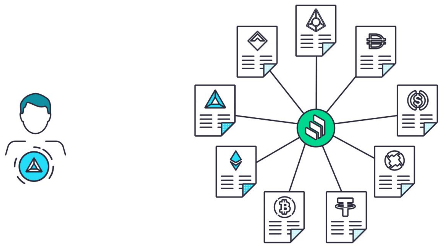
Secure MultiParty Computation (MPC)

MPC enables computing directly on private data!





Privacy-preserving ML



Threshold Signatures

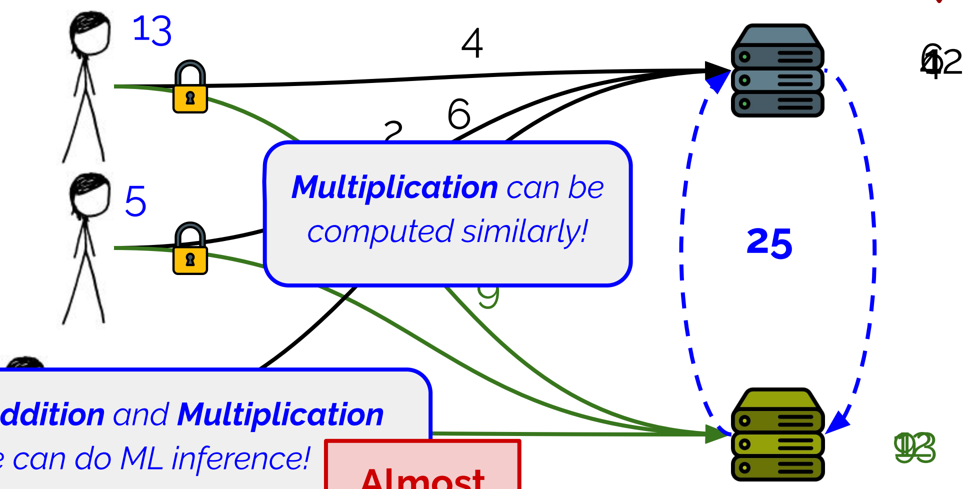
MPC for Machine Learning

Servers maintain a private sum

Three users want to compute the **sum** of their **private inputs**.

Each user **secret shares** their input into random looking numbers.

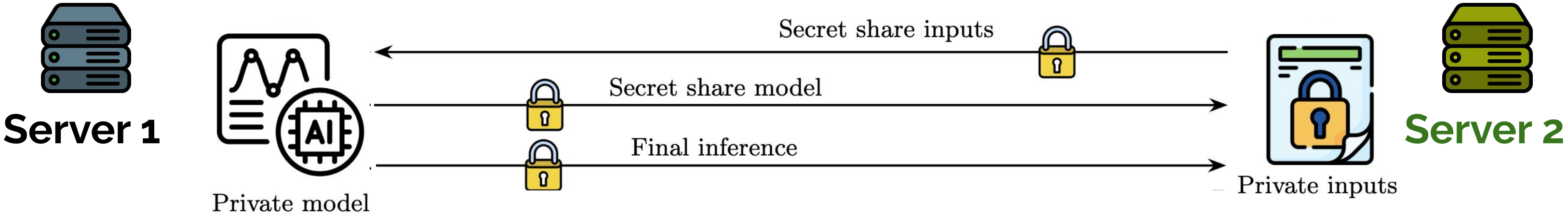
(E.g.,: 4 and 9 reveal nothing about 13)



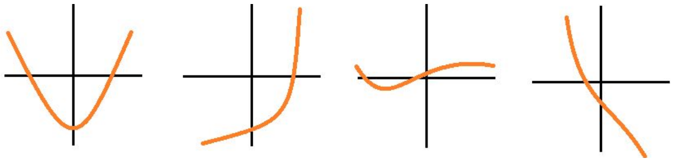
__Multiplication__ can be computed similarly!

Using __Addition__ and __Multiplication__ we can do ML inference!

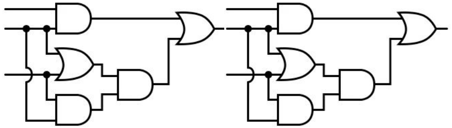
Almost



Non-Linear Functions in MPC



MPC protocols cannot evaluate **non-linearities** directly!



→**Boolean (aka garbled) circuits** can be used but are big and **expensive**.

→**Polynomial Approximations** can be used but are **slow** (**high communication**) and introduce big approximation **errors**.

SOTA MPC protocols evaluate non-linearities as lookup tables (LUTs), but

LUTs **scale poorly** for high precision → **very high communication**

The **Curl** Framework

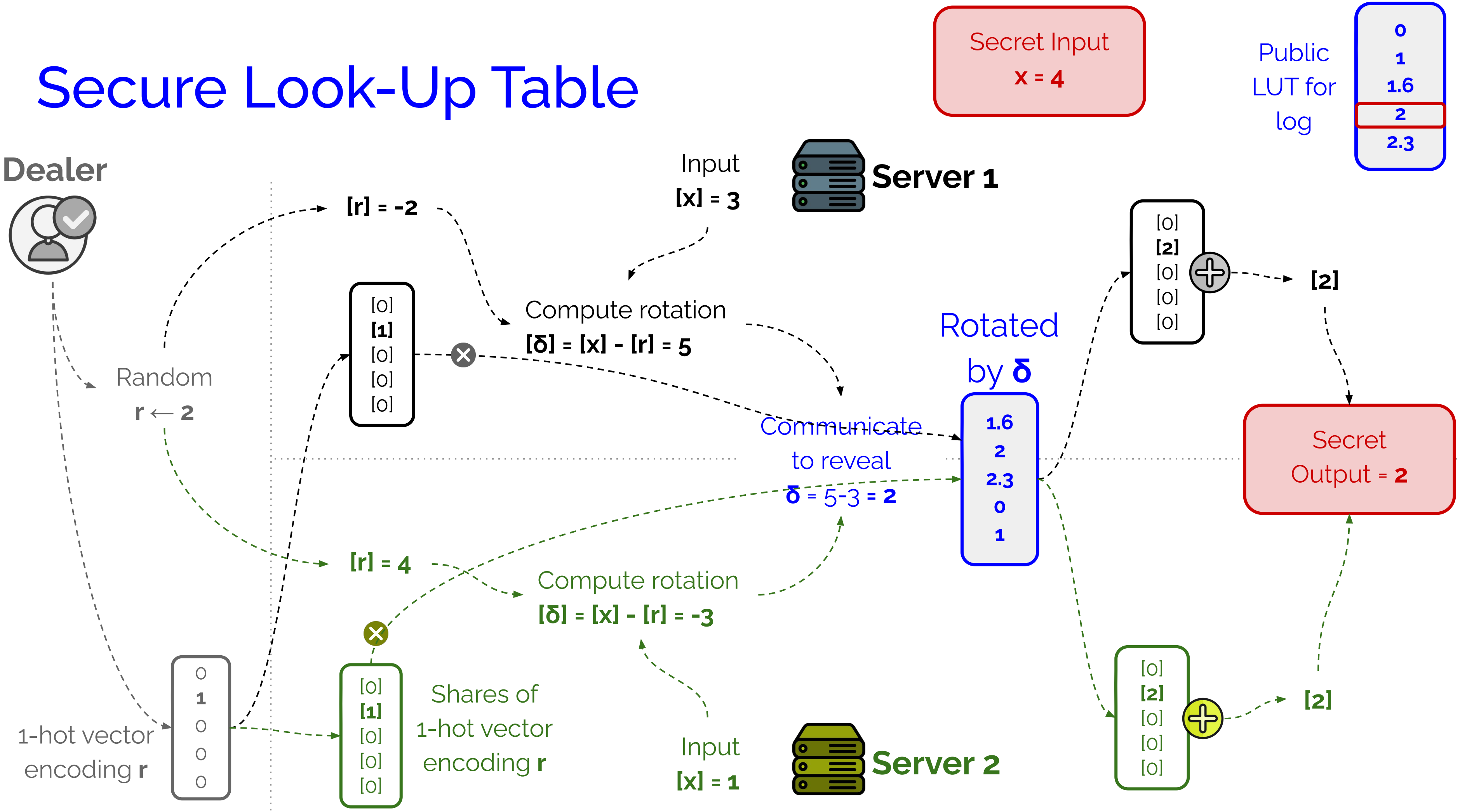
Curl

- Construct smaller LUTs **without sacrificing accuracy**
  - Using Discrete Wavelet Transforms (DWT)

- MPC-tailored protocols for evaluating DWT LUTs:
  - **Haar DWT:** faster, higher errors
  - **Biorthogonal DWT:** slower, lower errors

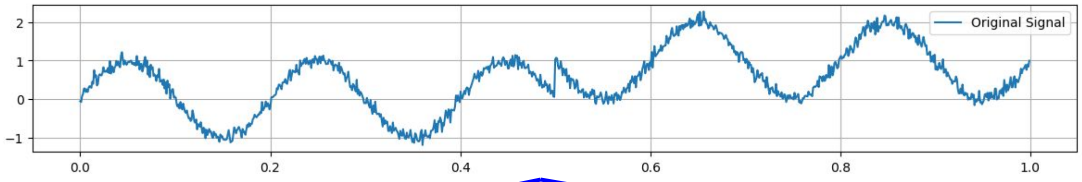
- Experiments over a suite of commonly used non-linear functions + LLMs.

Secure Look-Up Table



Discrete Wavelet Transform (DWT)

Initial signal **s**



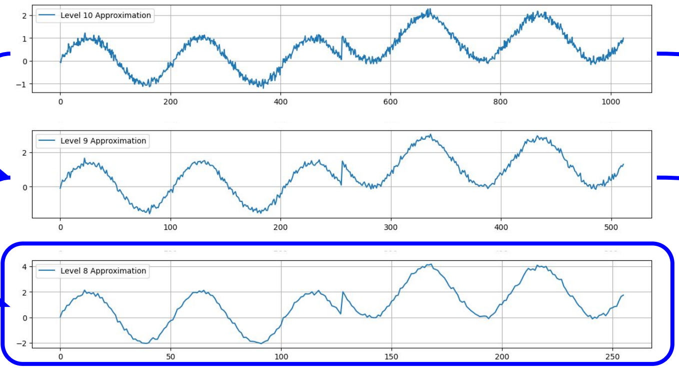
½

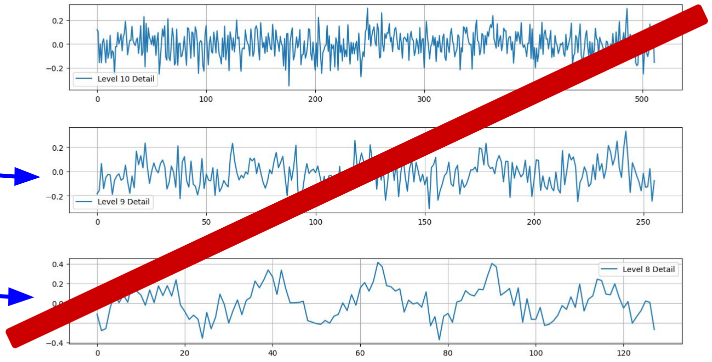
Approximations $\boldsymbol{a}$

Details $\boldsymbol{d}$

½

½





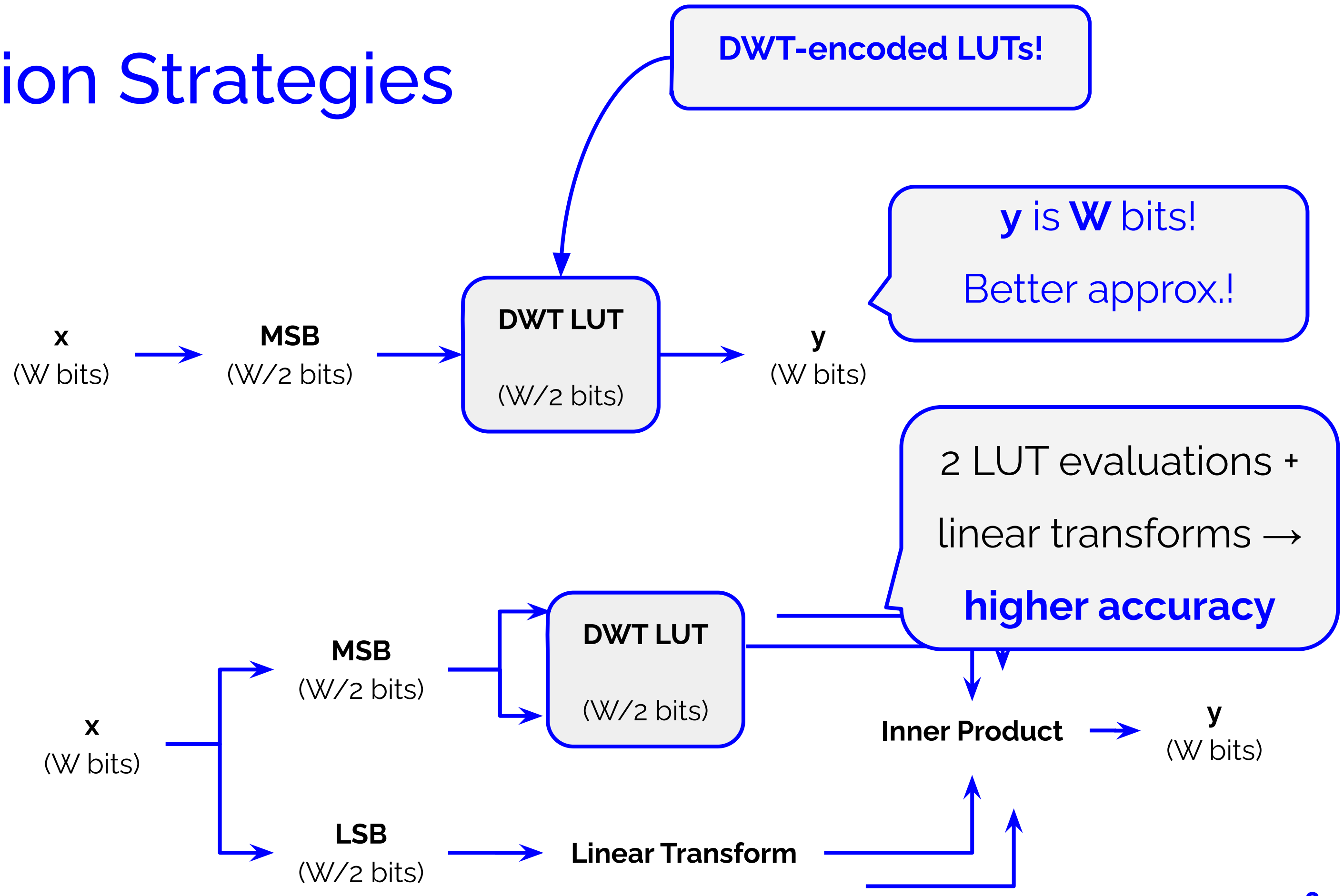
Smooth part of **s** remains unchanged!

Details can be set to zero!

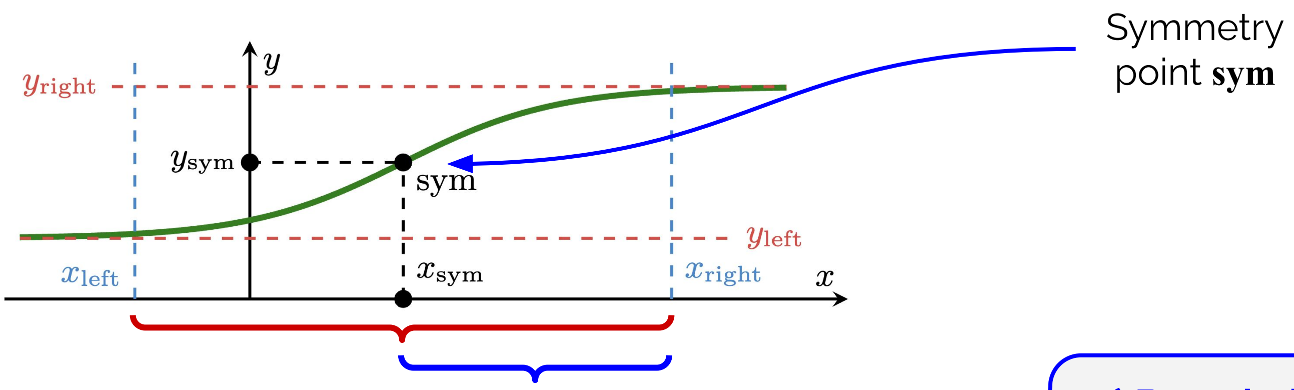
Approximation Strategies

1) Haar DWT

2) Biorthogonal DWT



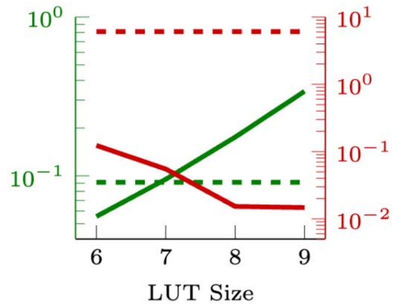
Bounded & S-Shaped Functions



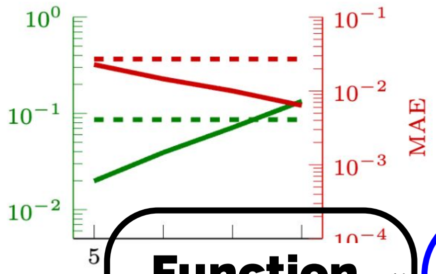
1) Bounded: Almost constant outside of $\mathbf{x}_{\text{right}}$ and $\mathbf{x}_{\text{left}}$

2) Symmetry: Function is the same (flipped) around **sym**

Evaluations: Approximations



Square Root



Inv

→ **Lower errors** than CrypTen

→ **Faster** for LUTs < 2^7

Function s				**Curl**					**CrypTen**				
Op.	Prot ol	Domain	LU	Latency (sec.)†	Com.‡ Rounds	Com.‡ MB	Error§ MAE	Error§ MRE	Latency (sec.)	Com. Rounds	Com. MB	Error MAE	Error MRE
log	Fig. 7	$(0, 64)$	2	0.17	4	2.6	2.09e-2	5.48e-2	0.17	40	39.8	2.14e-2	6.36e-3
reciprocal	Fig. 7	$(1, 64)$	2	0.09	4	2.6	7.18e-4	1.43e-3	0.11	59	38.3	1.7e-4	7.05e-3
sqrt	Fig. 7	$(0, 256)$	2	0.06	4	2.6	1.23-1	1.11e-2	0.09	26	17.3	6.09e+0	4.04e-1
invsqrt	Fig. 6	$(0, 256)$	2	0.04	2	1.0	1.45e-2	1.14e-1	0.09	24	15.7	2.69e-2	0.405e-1
sin	App. B.3	$(-64, 64)$	2	0.08	16	20.4	4.55e-3	1.14e-2	0.11	37	24.1	8.52e-1	1.58e+0
cos	App. B.3	$(-64, 64)$	2	0.08	16	20.4	4.77e-3	9.85e-2	0.10	37	24.1	8.86e-1	1.45e+0
sigmoid	Fig. 11	$(-64, 64)$	2	0.10	22	33.6	4.70e-5	7.83e-2	0.11	26	26.2	7.00e-5	3.49e+0
	Fig. 7	$(-64, 64)$	2	0.10	4	2.6	1.11e-2	6.59e-2	0.11	26	26.2	7.00e-5	3.49e+0
tanh	Fig. 11	$(-64, 64)$	2	0.09	22	33.6	2.31e-4	3.96e-4	0.13	26	26.2	8.60e-5	1.19e-4
erf	Fig. 11	$(-64, 64)$	2	0.09	22	33.6	8.98e-4	1.83e-3	0.21	56	36.2	3.39e+7	3.40e+7
GeLU	App. B.2	$(-64, 64)$	2	0.10	30	47.7	5.95e-3	2.79e+0	N/A	N/A	N/A	N/A	N/A
	Fig. 7	$(-4, 4)$	2	0.11	4	2.6	2.60e-3	5.02e-2	N/A	N/A	N/A	N/A	N/A
SiLU	App. B.2	$(-64, 64)$	2	0.14	30	47.7	2.61e-3	5.48e-3	N/A	N/A	N/A	N/A	N/A
	Fig. 7	$(-64, 64)$	2	0.09	4	2.6	1.54e-1	1.18e-1	N/A	N/A	N/A	N/A	N/A

Evaluations: Running LLMs in seconds

Sequence length = 64

Model	Latency (s)	Rounds	Com. (GB)
BERT Tiny	3.55	409	1.34
BERT Base	13.63	1,629	2.8
BERT Large	33.93	3,093	5.66
GPT-2	16.61	1,630	3.77
GPT-Neo	103.4	3,118	14.9

BERT Base
(seq. len = 128)

Framework	Latency (s)	Rounds	Com. (GB)
Iron [35]	475	13,663	281
MPCFormer [47]	55.3	–	12.1
Puma [21]	33.9	–	10.8
Bolt [57]	185	10,509	59.6
Bolt (WE) [57]†	91	10,901	25.7
Curl	22.5	1,629	5.7

† In Bolt, WE stands for word elimination.

Fastest runtime

Fewer Rounds

Lowest Communication

Conclusions

- Lookup Tables (LUTs) can be used to evaluate non-linear functions in MPC
  - LUTs **scale poorly** for high precision → enormous communication.
  - **Polynomial approximations** and **quantization** yield **low accuracy**!

- **Curl:** smaller LUTs without sacrificing accuracy
  - Using Discrete Wavelet Transforms (DWT) → **low communication**
  - Reduced LUT sizes → **high accuracy**
  - Run LLMs (BERT Tiny/Base/Large, GPT-2, GPT Neo) → **in seconds!**

- Curl's technique can enhance related works and even **FHE** (e.g., Ripple [1])

[1] C. Gouert, M. Ugurbil, D. Mouris, M. de Vega, and N. G. Tsoutsos. **Ripple: Accelerating Programmable Bootstraps for FHE with Wavelet Approximations.** In International Conference on Information Security (ISC), 2024.

Curl: Private LLMs through Wavelet-Encoded Look-Up Tables

Manuel B. Santos[1], **Dimitris Mouris**[1], Mehmet Ugurbil[1], Stanislaw Jarecki[1,2], José Reis[1], Shubho Sengupta[3] and Miguel de Vega[1]

{ manuel.santos, dimitris, memo, stanislaw.jarecki, jose.reis, miguel }@nillion.com

ssengupta@meta.com

https://ia.cr/2024/1127

https://github.com/jimouris/curl

1 nillion

2 The University of California, Irvine

3 Meta