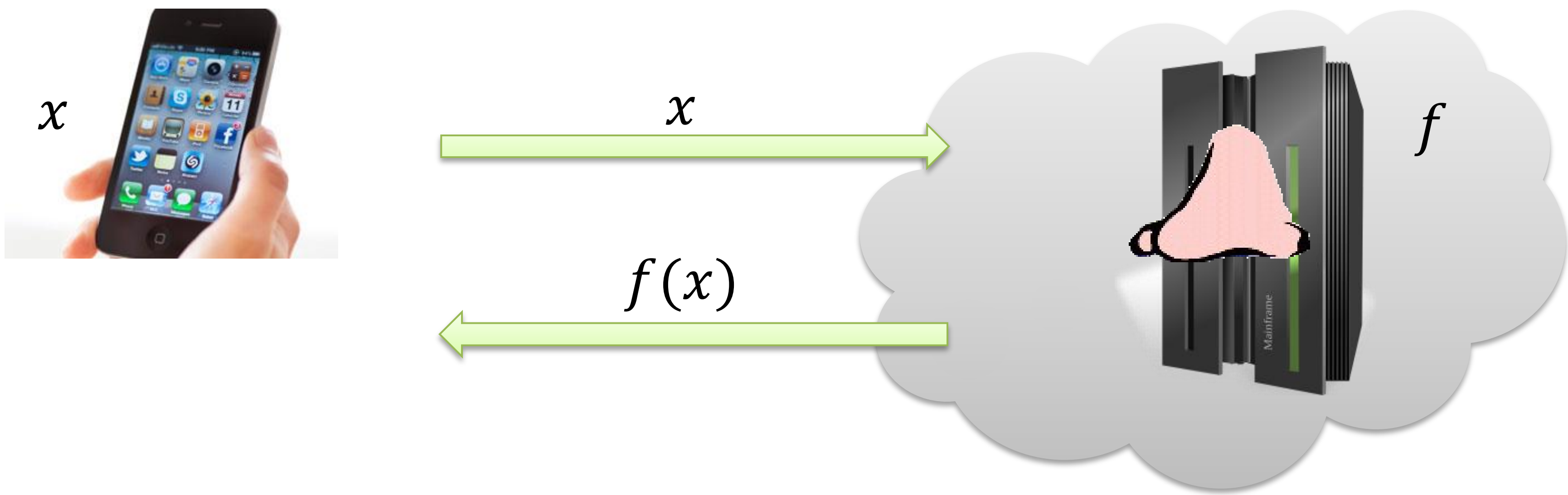
5 years of FHE

Zvika Brakerski
Weizmann Institute of Science

Aarhus MPC Workshop, May 2014

Outsourcing Computation

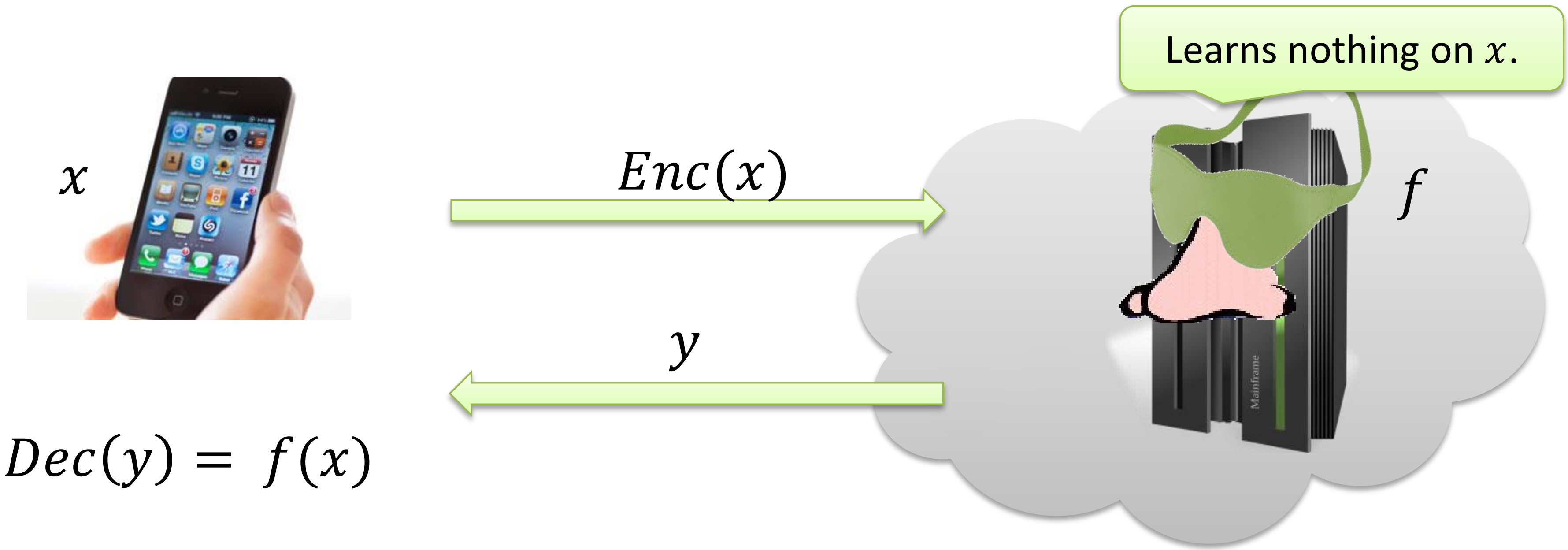


Email, web-search, navigation, social networking…

Search query, location, business information, medical information…

What if x is private?

Outsourcing Computation – Privately



WANTED

Homomorphic Evaluation function:

$$Eval: f, Enc(x) \rightarrow Enc(f(x))$$

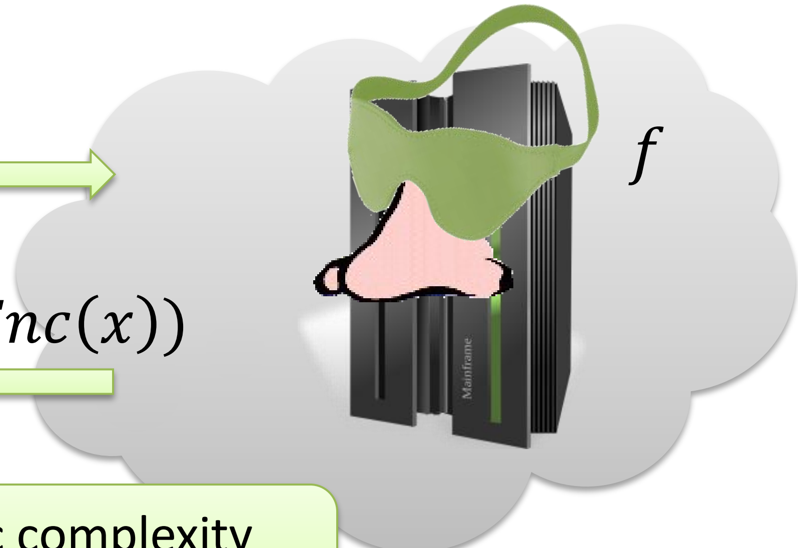
Fully Homomorphic Encryption (FHE)

sk, pk

x

$Enc_{pk}(x)$

evk



f

$y = Eval_{evk}(f, Enc(x))$

Correctness:

$$Dec_{sk}(y) = f(x)$$

Input privacy:

$$Enc(x) \cong Enc(0)$$

Compactness: Dec complexity independent of f.

Fully Homomorphic = Correctness for any efficient f

= Correctness for universal set

- NAND.
- $(+,\times)$ over $\mathbb{Z}_2$ (= binary XOR, AND)

Some Applications

In the cloud:

- Private outsourcing of computation.
- Near-optimal private outsourcing of storage (single-server PIR). [G09,BV11b]
- Verifiable outsourcing (delegation). [GGP11,CKV11,KKR13]
- Private machine learning in the cloud. [GLN12,HW13]

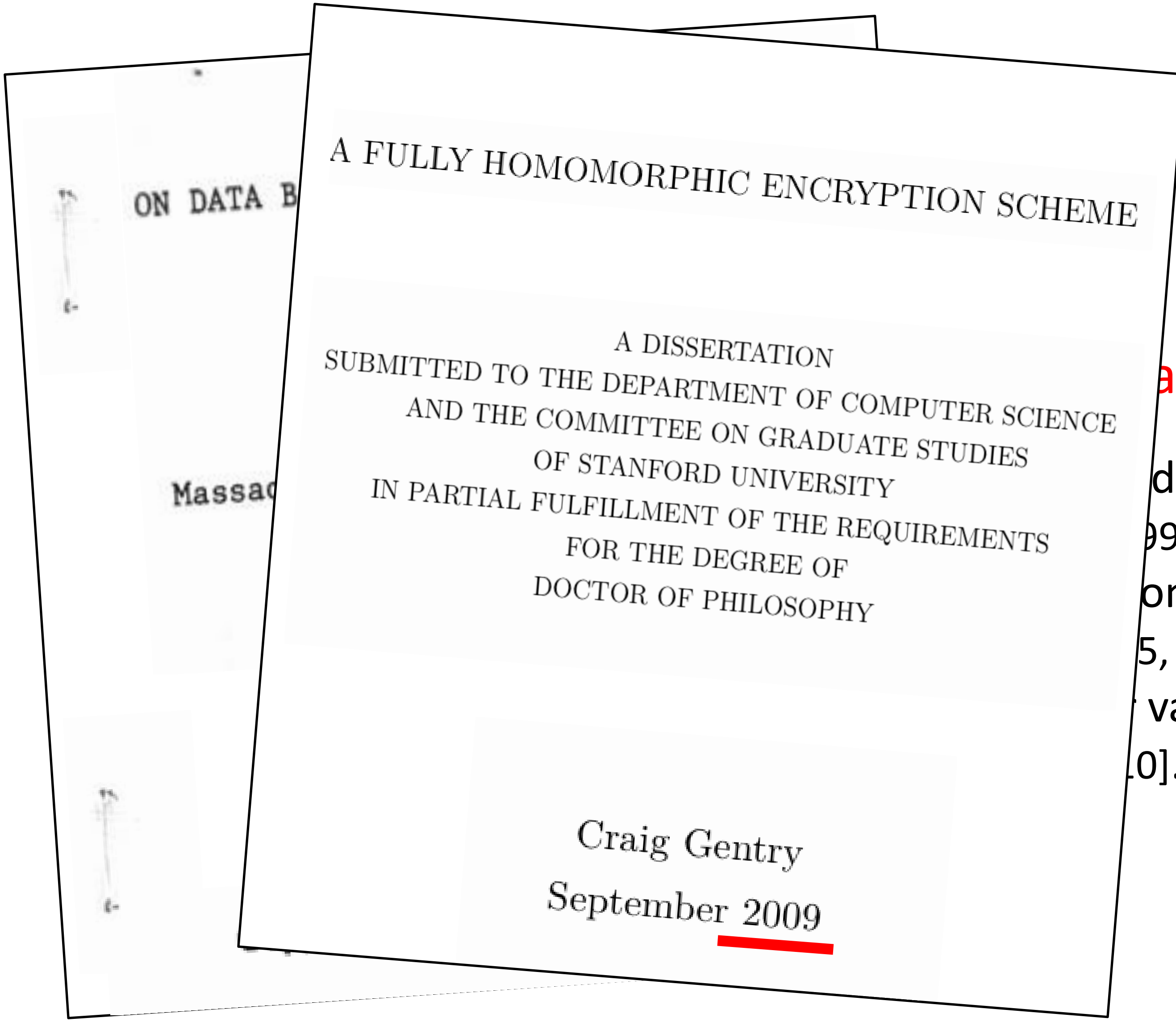
Secure multiparty computation:

- Low-communication multiparty computation. [AJLTVW12,LTV12]
- More efficient MPC. [BDOZ11,DPSZ12,DKLPSS12]

Primitives:

- Succinct argument systems. [GLR11,DFH11,BCCT11,BC12,BCCT12,BCGT13,…]
- General functional encryption. [GKPVZ12]
- Indistinguishability obfuscation for all circuits. [GGHRSW13]

Making Crypto History



of hardly scratching
ace:

ddition [RSA78, R79, GM82,
99, R05].
on + 1 multiplication
5, GHV10].
variants [SYY99, IP07,
0].

… is it even possible?

Constructing Homomorphic Encryption [G09]

For now: Any non-trivial homomorphism

secret algebraic equivalence
e.g. (mod p) for secret p

Basic Idea: Find scheme s.t. $c \approx m + 2e$

ciphertext message small (even) noise

Add/multiply ciphertexts $\Rightarrow$ Add/multiply messages

Noise grows with homomorphic evaluation –
must not grow "too much"!

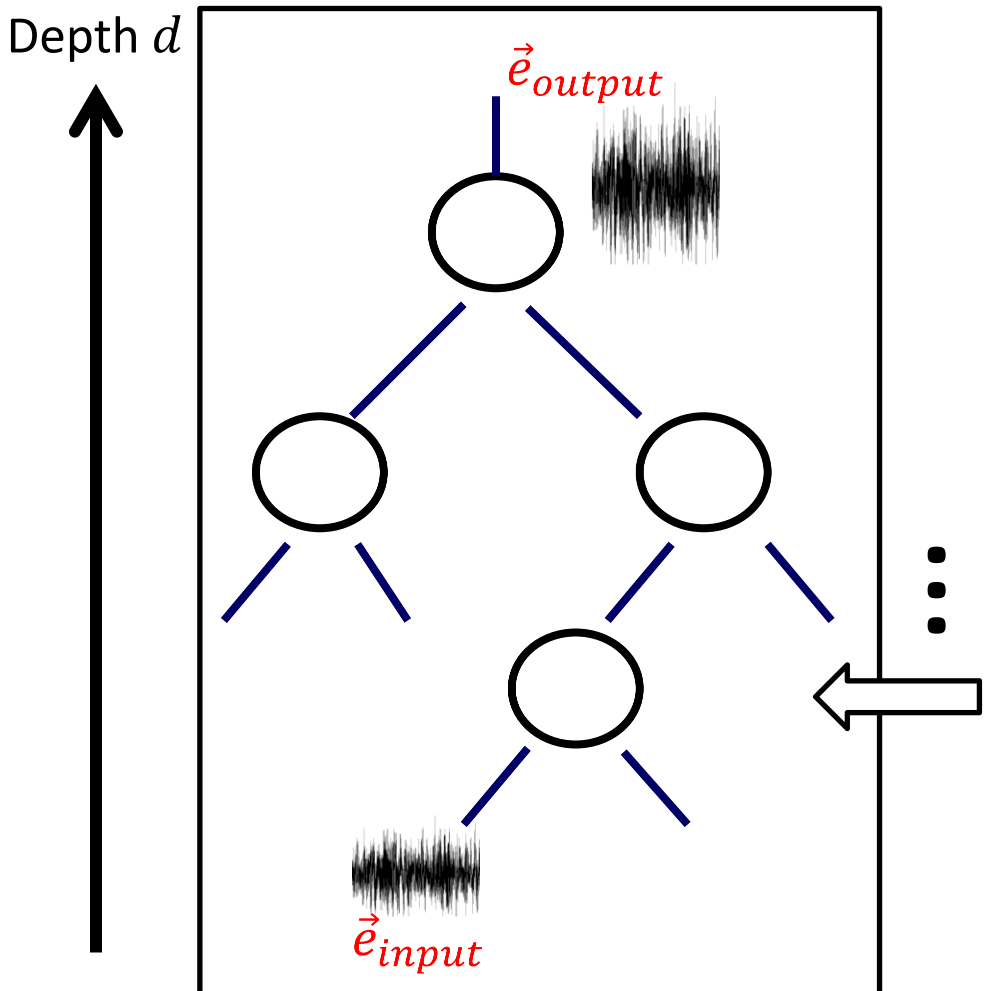
In this example [DGHV10]: $\|e_{mult}\| \approx \|e_{input}\|^2$

Noise in Homomo

Noise grows during hom

Benchmark: $E = N, d = \log N$

$$\frac{\left\|\vec{e}_{output}\right\|}{\left\|\vec{e}_{input}\right\|} = N^N$$



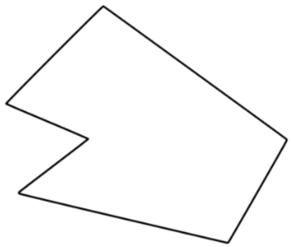
$$\left\|\vec{e}_{output}\right\| \le E^{2^d}$$

$$\left\|\vec{e}_{i+1}\right\| \le \left\|\vec{e}_i\right\|^2$$

$$\left\|\vec{e}_{input}\right\| \le E$$

FHE Challenges

Simplicity.



Security.

- Assumptions.
- Security notions.

Efficiency.

- Size of keys/ciphertexts.
- Time overhead for Eval.
- Computational model.

Second Generation FHE [BV11b]

Any non-trivial homomorphism

secret key

publicly computable

Basic Idea: Find scheme s.t. $c \cdot s \approx m + 2e$

ciphertext

message

small (even) noise

Different $(\cdot) \Leftrightarrow$ different assumption:

- Vector IP $\Rightarrow$ Learning with Errors (LWE) / Lattice
- Vector of polynomials $\Rightarrow$ Ring LWE / Ideal lattice [BV11a]
- Polynomials over a ring $\Rightarrow$ NTRU / Ideal lattice [LTV12]

Second Generation FHE [BV11b]

Any non-trivial homomorphism

For $m = s^2$:

$$c_{s^2 \to s} \cdot s \approx s^2 + 2e$$

Basic Idea: Find scheme s.t. $c \cdot s \approx m + 2e$

Multiplication (very high level): $c_{mult} = (c_1 \cdot c_2) \cdot c_{s^2 \to s}$

$$(c_1 \cdot c_2) \cdot (s^2) \approx m_1 m_2 + 2e$$

$$(c_1 \cdot c_2) \cdot c_{s^2 \to s} \cdot s \approx m_1 m_2 + 2e$$

key-switching ciphertext

In evk (public parameter)

Key switching ⇒ proxy re-encryption

Second Generation FHE [BV11b]

Follow-ups [BGV12,GHS12a,B12,GHS12c,BGH13]:

- Improved noise behavior: $E \to (N+1) \cdot E$ (instead of E^2)
  $\Rightarrow$ I/O noise ratio ($E = N, d = \log N$) drops to $N^{\log N}$.
- Improved security reductions.
- Significant efficiency improvements using "batching".

Conclusion:

- Simplified constructions.
- Improved hardness assumptions (quasi-poly apx. to worst case lattice problems).
- More efficient by orders of magnitude.

The "Approximate Eigenvector" Method [GSW13]

Basic Idea: Find scheme s.t. $c \cdot s \approx m\, s + 2e$

Multiplication (very high level): $c_{mult} = c_1 \cdot c_2$

$$(c_1 \cdot c_2) \cdot s \approx c_1 m_2 s + 2e \approx m_1 m_2 s + 2e$$

No need for key-switching ciphertext in evk!

$\Rightarrow$ IB-FHE, AB-FHE via [GPV08,CPHK10,ABB10,GVW13]

The "Approximate Eigenvector" Method [GSW13]

Basic Idea: Find scheme s.t. $c \cdot s \approx m\, s + 2e$

Actually implied by previous method:

Starting with $m = 0$: $c \cdot s \approx 0 + 2e$

$\Rightarrow \; c' = c + 1 \cdot m$: $c' \cdot s \approx m\, s + 2e$

Howev

Ciphertext size = $N^2 \Rightarrow$ Large!

C = matrix, $\vec{s}$ = "eigenvector".

Approximate Eigenvector Method [GSW13]

$C_1 \cdot \vec{s} = m_1 \vec{s} + \vec{e}_1$

$C_2 \cdot \vec{s} = m_2 \vec{s} + \vec{e}_2$

$C_{mult} = C_1 \cdot C_2$:

Can also use $C_2 \cdot C_1$

$$\begin{aligned}
(C_1 \cdot C_2) \cdot \vec{s} &= C_1 (m_2 \vec{s} + \vec{e}_2) \\
&= m_2 C_1 \vec{s} + C_1 \vec{e}_2 \\
&= m_2 (m_1 \vec{s} + \vec{e}_1) + C_1 \vec{e}_2 \\
&= m_2 m_1 \vec{s} + \underbrace{m_2 \vec{e}_1 + C_1 \vec{e}_2}_{\vec{e}_{mult}}
\end{aligned}$$

$||C_1||$ can be reduced to $\approx N$.

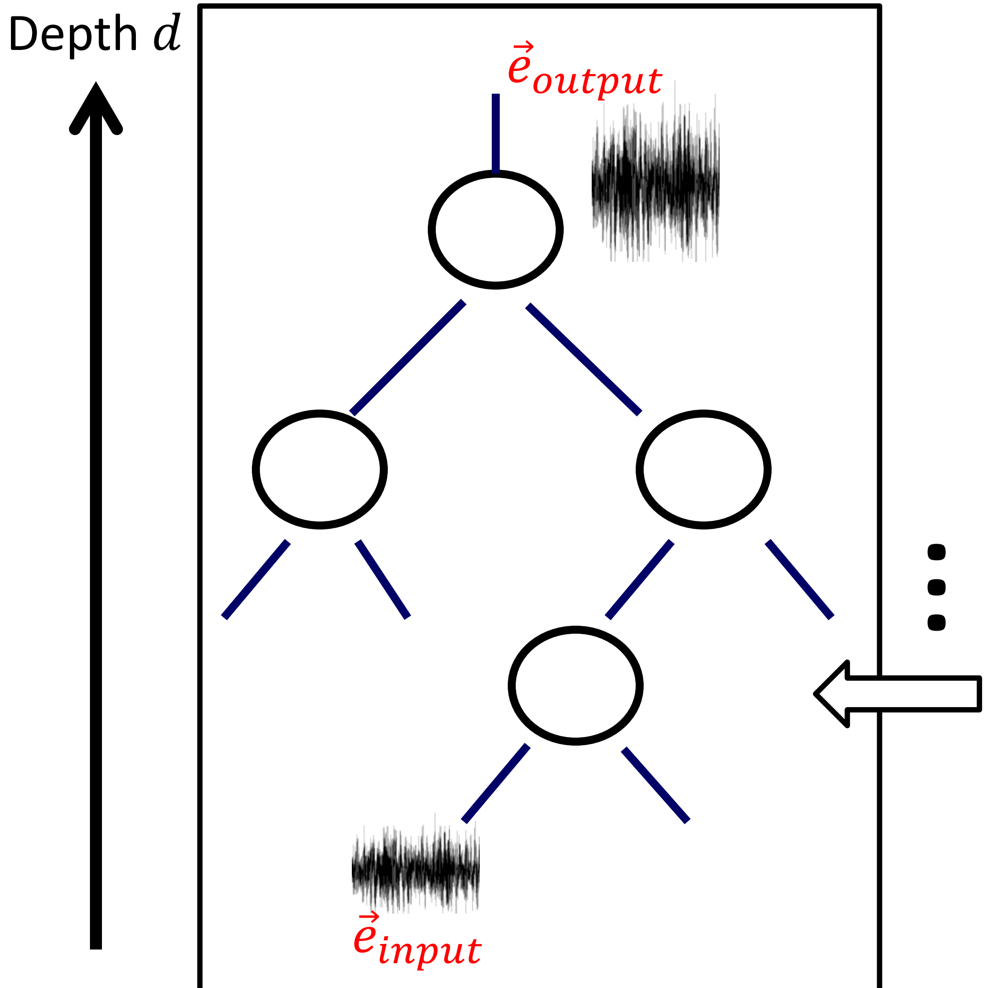
$$||\vec{e}_{mult}|| \leq N \cdot ||\vec{e}_2|| + m_2 \cdot ||\vec{e}_1|| \leq (N+1) \cdot \max\{||\vec{e}_1||, ||\vec{e}_2||\}$$

Noise in Homomo

Benchmark: $E = N, d = \log N$

$$\frac{\|\vec{e}_{output}\|}{\|\vec{e}_{input}\|} = N^{\log N}$$

Noise grows during homo



$$\|\vec{e}_{output}\| \leq (N+1)^d \cdot E$$

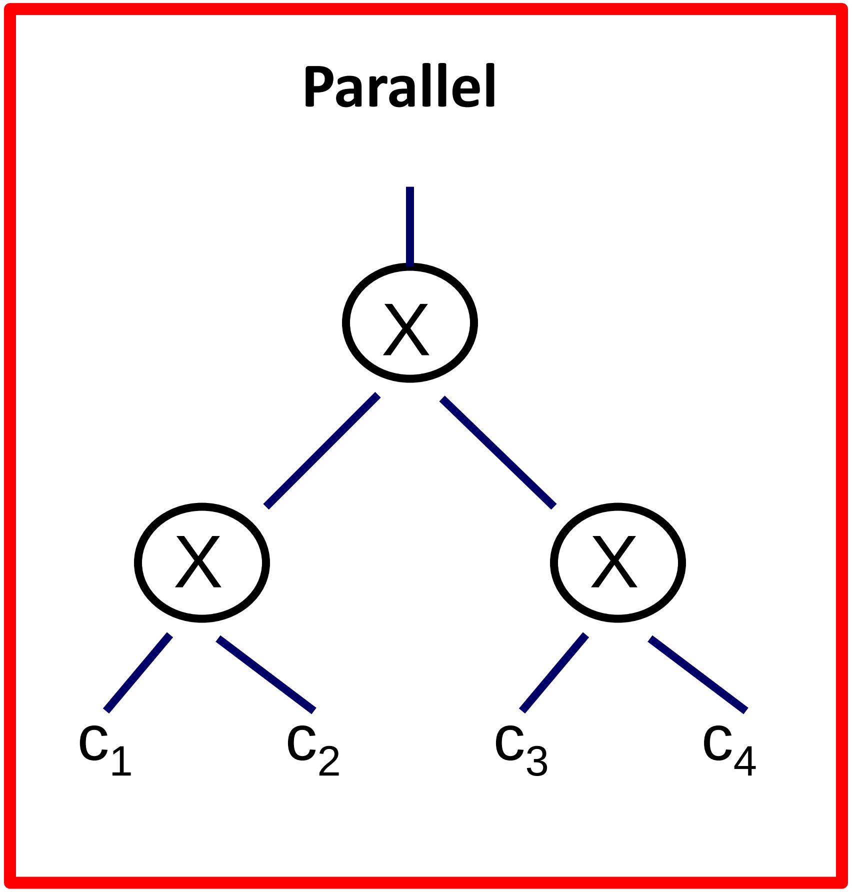
$$\vdots$$

$$\|\vec{e}_{i+1}\| \leq (N+1)\|\vec{e}_i\|$$

$$\|\vec{e}_{input}\| \leq E$$

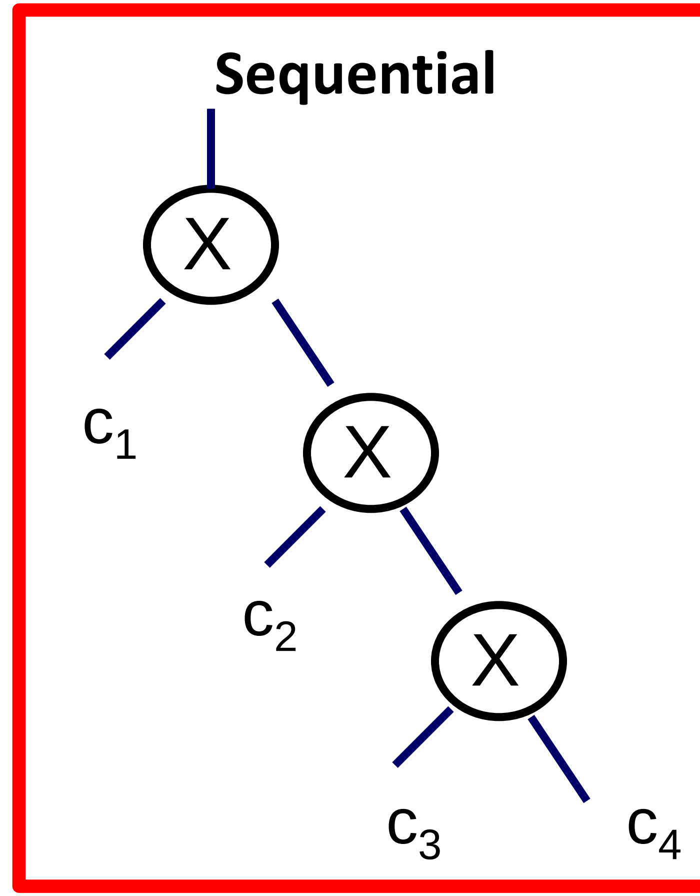
Sequentialization [BV14]

What is the best way to evaluate a product of k numbers?



Conventional wisdom

vs.

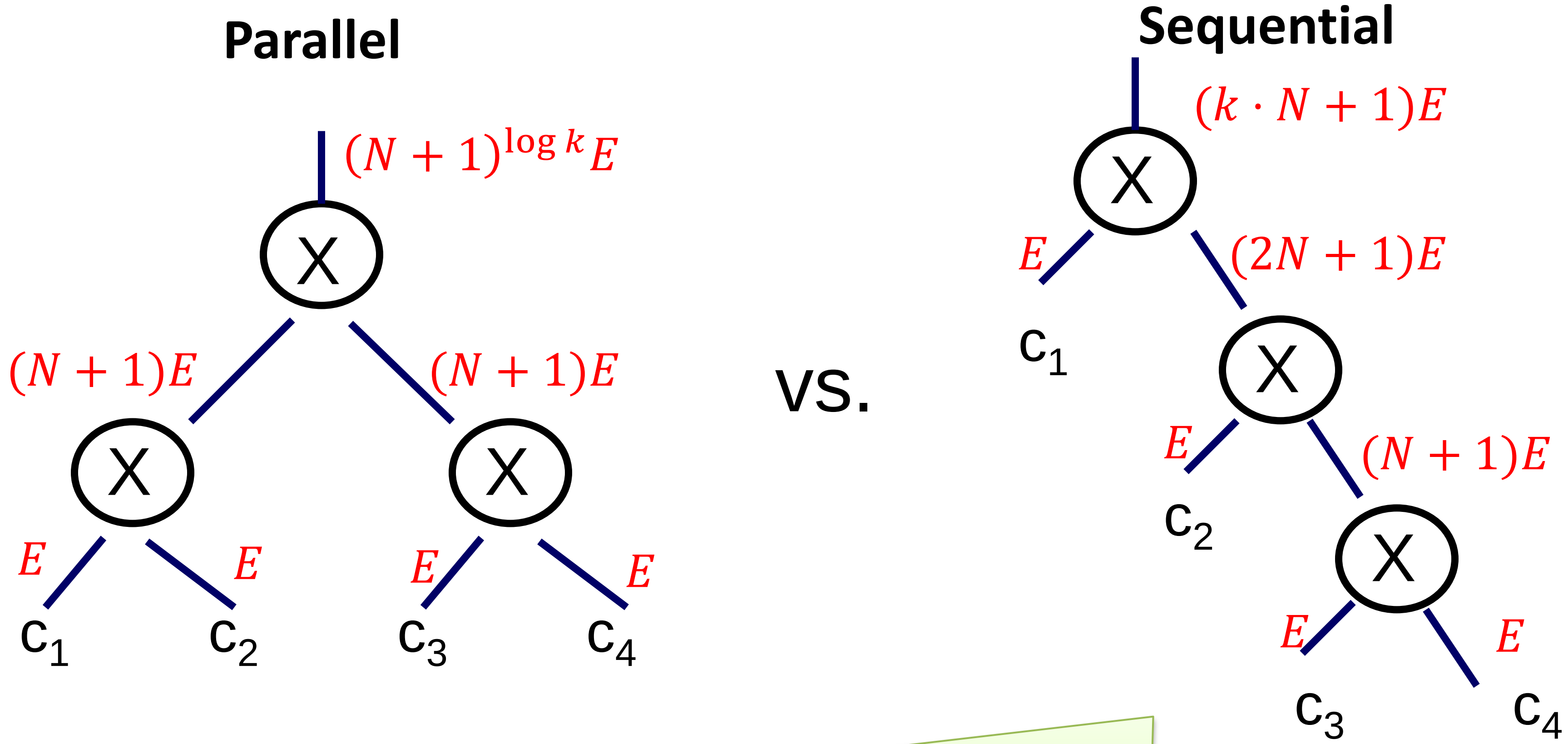


Actually better

(if done right)

Sequentialization [BV14]

$$\boxed{N \cdot \|\vec{e}_2\| + m_2 \cdot \|\vec{e}_1\|}$$



Actually even better! Only the 1-suffix matters.

⇒ MUXing only costs *additive* noise.

Sequentialization [BV14]

$$MUX(x, y_1, y_2) = x \cdot y_1 + (1-x) \cdot y_2 \quad \Rightarrow \quad \|e_{MUX}\| \leq \max\{\|e_y\|\} + N\|e_x\|$$

Barrington's Theorem [B86]: Every depth d computation can be transformed into a width-5 depth 4^d **branching program**.

A sequence of MUXes.

- Noise growth improved to $poly(N)$.

  $\Rightarrow$ Better security – breaks barrier of [BGV12, B12,GSW13].

- Using dimension-modulus reduction (from [BV11b]) $\Rightarrow$ same hardness assumption as non homomorphic encryption.

- Short ciphertexts (with bootstrapping – coming up).

Sequentialization [BV14]

Barrington's Theorem [B86]: Every depth d computation can be transformed into a width-5 depth 4^d **branching program**.

A sequence of MUXes.

- Noise growth improved to $poly(N)$.

  $\Rightarrow$ Better security – breaks barrier of [BGV12, B12,GSW13].
- Using dimension-modulus reduction (from [BV11b]) $\Rightarrow$ same hardness assumption as non homomorphic encryption.
- Short ciphertexts (with bootstrapping – coming up).

Implementations of FHE

- HElib (IBM/NYU)
  - Ring-LWE (ideal-lattice) scheme of [BGV12], optimizations of [GHS12a]
  - https://github.com/shaih/HElib

- "Stanford FHE"
  - LWE scheme of [B12] with optimizations
  - http://cs.stanford.edu/~dwu4/fhe.html

- Unpublished code
  - Ring-LWE implementation of [GHS12b].
  - Over the integers implementation of [CCKLLTY13].

Approximate eigenvalue method not implemented yet.