FHE Compiler Using Buildit

Secure Arithmetic Scheduling using BGV and Noise Reduction Techniques

DEVA SUVEDH
CS22BTECH11016

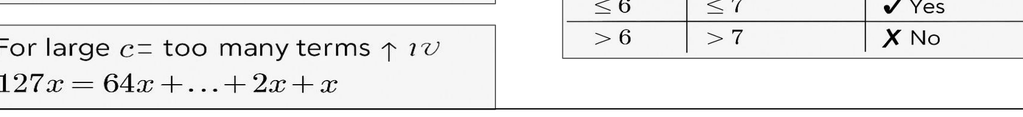
MEDIKONDA SREEKAR
CS22BTECH11037

MENTOR
RAJIV SHAILESH CHITALE

GUIDE
RAMAKRISHNA UPADRASTA

Introduction

- Homomorphic Encryption - computation on encrypted data.
- BGV scheme supports exact integer and logical operations.
- BuildIt separates symbolic scheduling from encrypted execution for optimization.



Problem Statement

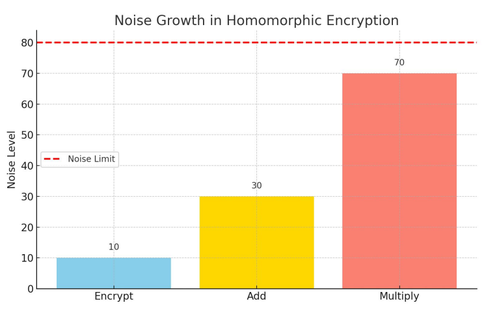
- Exploring compilation+optimization techniques for FHE.
- Study the application of BUILDIT compilers to obtain performance improvements.

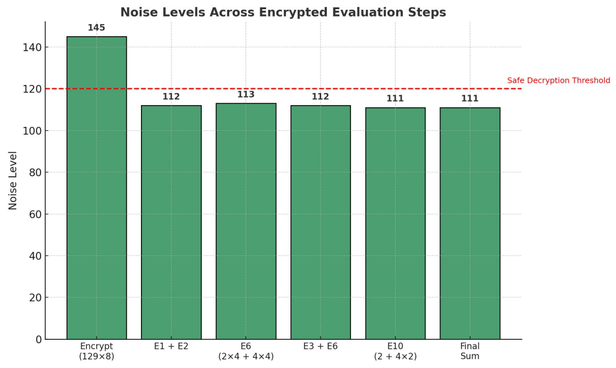
Motivation

- Protect sensitive data during outsourced computation with FHE.
- Handle noise growth challenges to enable encrypted computations.
- Simplify and automate encrypted computation scheduling with BuildIt.

Noise Growth & Its Effects

- Every operation increases ciphertext noise
- Excessive noise leads to decryption failure





Mathematical Insight:

- Any c decomposed as: $c \cdot x = \binom{\lfloor \log_2 c \rfloor}{i=0} \cdot b_i \, 2 \cdot \frac{c}{i} \cdot x = \sum_{b_i \in 0} b_i \, (2 \cdot x)$

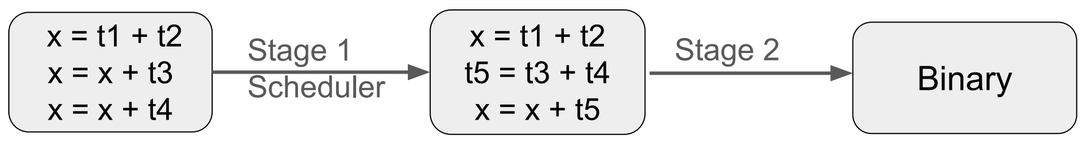
- For small c/x : keep circuit shallow
  $13x = (8+4+1)x = 8x + 4x + x$

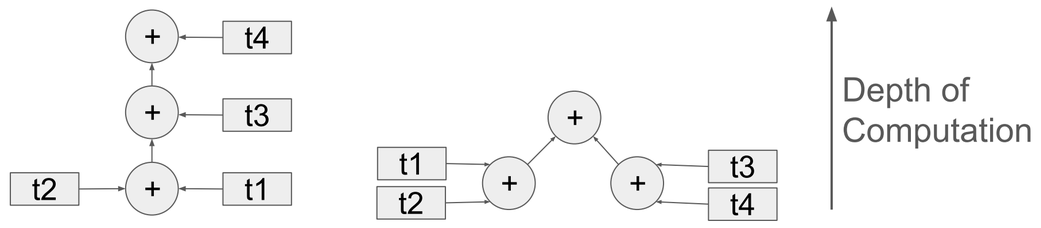
- For large c= too many terms ↑ $\imath v$
  $127x = 64x + \ldots + 2x + x$

$log_2 c$	Terms in Sum	Efficient?
≤ 6	≤ 7	✓ Yes
> 6	> 7	✗ No

Scheduler with BuildIt

- Stage 1 - symbolic analysis and optimization
  - Strength Reduction of multiplications
  - Instruction rescheduling builds balanced, low-depth trees.
- Stage 2 - compiles optimized schedule of encrypted operations





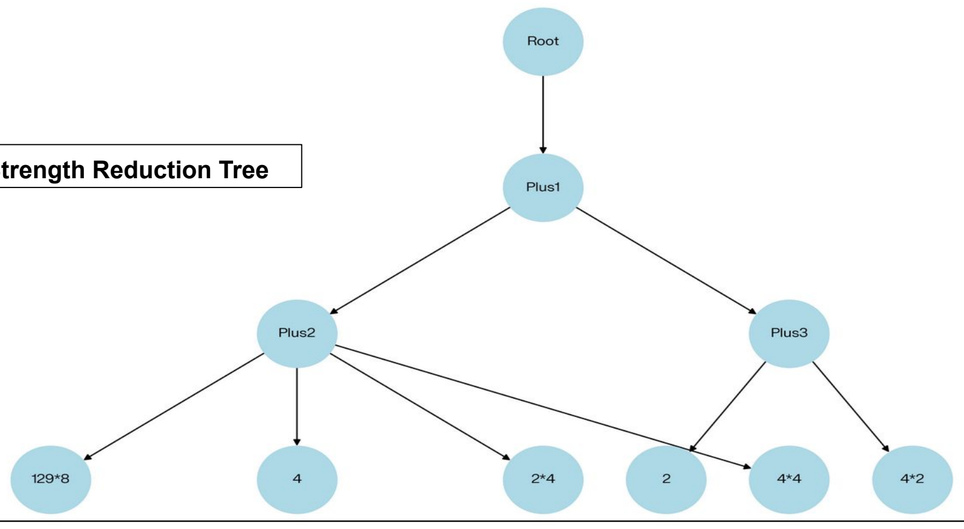
Optimizations

- Relinearization and modulus switching
- Rotation for vectorized operations
- Minimize multiplications to reduce noise growth
- BatchEncoder enables parallel encrypted ops via value packing.

Balancing Expression Evaluation

- Arithmetic expressions are converted to postfix for processing.
- Structured into balanced trees to minimize depth and noise.
- Terms stored as (coefficient, variable) for stack-based evaluation.

Strength Reduction Tree



Homomorphic Evaluation (BGV Scheme)

- Operands are encrypted using Microsoft SEAL's BGV scheme
- Arithmetic is performed directly on encrypted integers
- Supports both addition and multiplication

Work Done

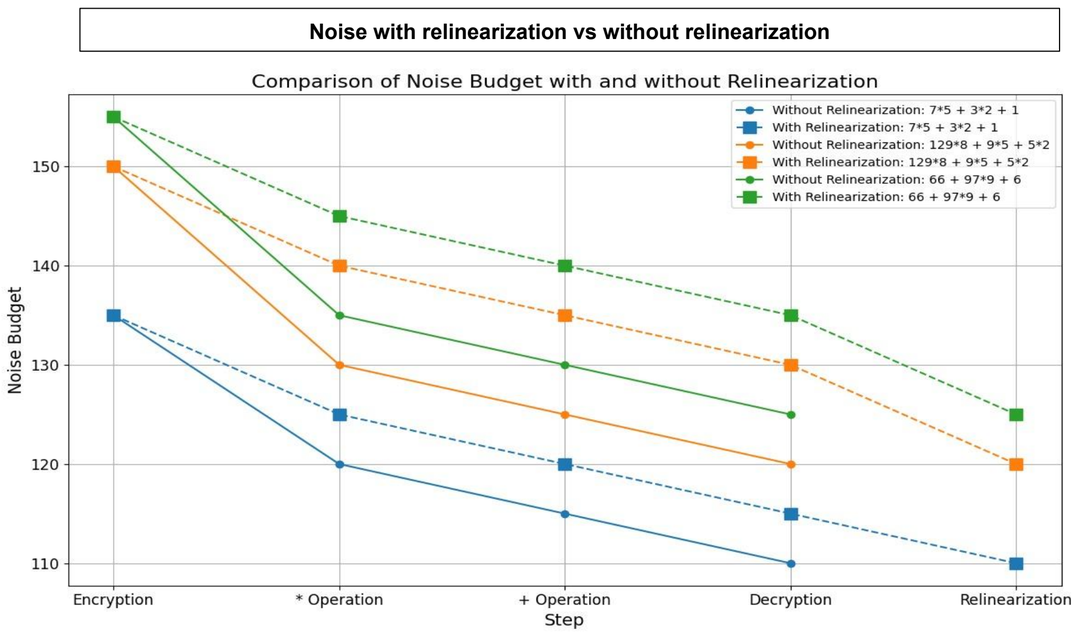
- Implemented expression parsing and postfix conversion for structured evaluation.
- Designed scheduler for noise reduction based on BuildIt framework
- Tested strength reduction with SEAL BGV.
- Applied batching, relinearization, and NTT optimizations.
- Monitored noise budget to ensure successful decryption of result.

References

- Microsoft SEAL (Simple Encrypted Arithmetic Library)
- Gentry, C. "Fully Homomorphic Encryption Using Ideal Lattices"
- Brakerski, Z., Gentry, C., & Vaikuntanathan, V.“(Leveled) Fully Homomorphic Encryption without Bootstrapping.”
- Buildit Framework for DSLs

Results

Noise with relinearization vs without relinearization



Encrypted Arithmetic Computation Pipeline with strength reduction