On the Hardness of Scheme-Switching Between SIMD FHE Schemes

KARIM ELDEFRAWY, **NICHOLAS GENISE**, NATHAN MANOHAR

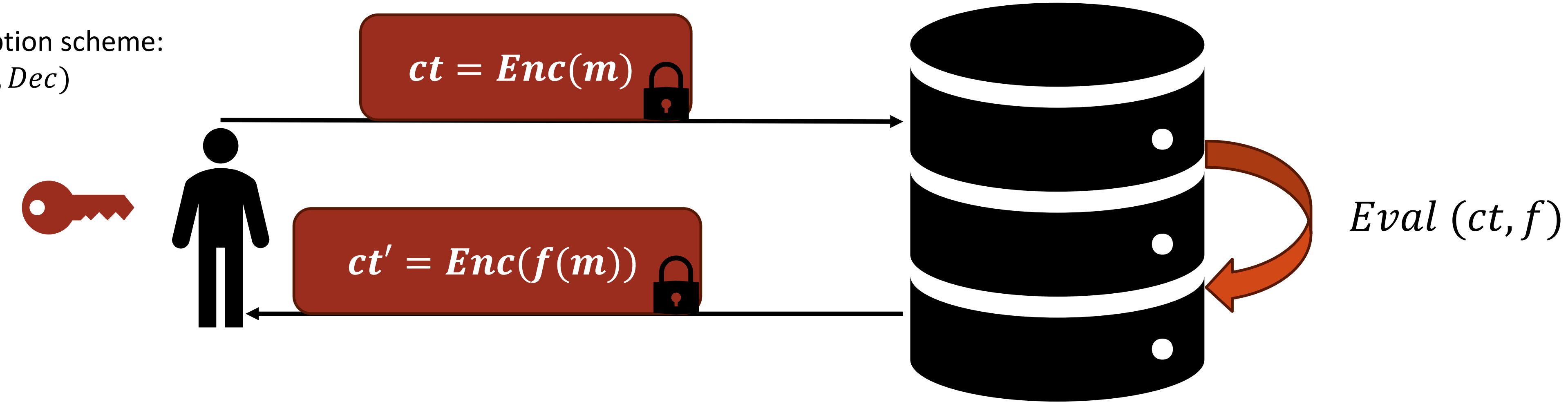
8/16/2023

Background

Homomorphic Encryption

Public-key encryption scheme:
$(KeyGen, Enc, Dec)$

Homomorphic encryption scheme:
$(KeyGen, Enc, \boldsymbol{Eval}, Dec)$



Here $f(\cdot)$ can be a medical diagnosis, classifier, or a DNN inference.

The scheme is <u>fully</u> homomorphic (FHE) if f can be any efficiently computable function and it is compact: decryption is the same throughout.

History of FHE

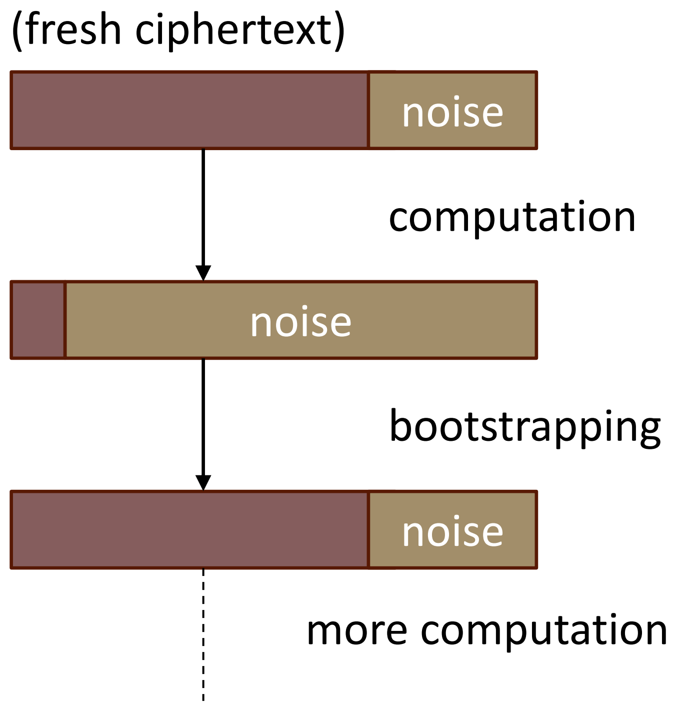
The idea of fully homomorphic encryption (FHE) was first thought of in 1978 by Rivest, Adleman, and Dertouzos.

In 2009, Craig Gentry, then a student at Stanford, described the first plausible construction using ideal lattices.

Intuition:

Lattice-based schemes are noisy with simple decryption functions: linear function, then rounding away the noise.

Bootstrapping homomorphically decrypts, lowering the noise.



Overview of FHE Families

Family	Operations	Payload/ciphertext
BGV/BFV	Arithmetic mod p	4k-64k SIMD mod p numbers
CKKS	Approximate arithmetic on fixed-point numbers	4k-64k SIMD fixed points numbers
FHEW/TFHE	Boolean arithmetic	A single 1-to-16-bit number

Applications

BGV/BFV:
- Private information retrieval (PIR)
- Private set intersection (PSI)
- Integer computations

CKKS:
- Neural network inference
- Logistic regression training
- Statistical analysis

FHEW/TFHE:
- Boolean circuits
- Lookup tables

Problems with Sticking to One Scheme Motivating Scheme-Switching

1. Hardware acceleration for each scheme differs beyond the “math layer” (NTT, mod. $+/\times$)
2. Some computations are much more efficient in certain schemes
3. Many real-world computations contain components that are more efficient in different schemes

Solutions

- Use a single scheme for every part of the computation (inefficient)
- Have client decrypt and re-encrypt under different scheme (requires interaction)
- Scheme-switch using bootstrapping
- Homomorphically scheme-switch between different FHE schemes w/out bootstrapping (focus of this work)

Structure of a BGV Ciphertext

A BGV ciphertext is a pair of polynomials such that:

$$ct = (c_0, c_1) \text{ with } c_0 + c_1 s = m + pe = m(X) + pe(X) = m \, mod \, p$$

p is a scalar and $e \sim \chi$ is noise.

The ciphertext modulus is Q and the polynomials are modulo $X^N + 1$, N a power of two,

ciphertext polynomials are $R_Q \coloneqq \mathbb{Z}_Q[X]/(X^N + 1)$.

$Q = \ q_1 q_2 \cdots q_D$ is a product of NTT-friendly machine-sized primes.

D is the **depth** and we reduce the modulus after each multiplication for noise-maintenance. This is called “modulus-switching” (“rescaling in CKKS”):

$$ct \leftarrow \lceil ct/q_D \rfloor_p \in R_{Q'}^2, \text{ for } Q' \coloneqq Q/q_D$$

SIMD packing: poly. interpolation, $m(X) = \ {DFT_p}^{-1}(\vec{m})$

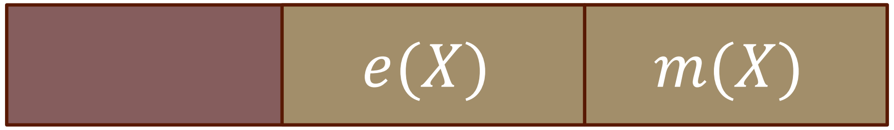
BFV/BGV/CKKS Ciphertexts

BFV has the plaintext message in the MSBs of the ciphertext (c_0, c_1):

$$c_0 + c_1 s = \lfloor\frac{Q}{p}\rceil m(X) + e(X)$$

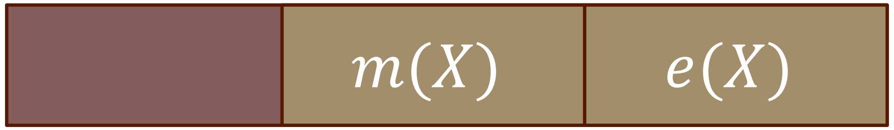
BGV has the plaintext message in the LSBs of (c_0, c_1):

$$c_0 + c_1 s = m(X) + pe(X)$$



CKKS has the plaintext message and the error as one:

$$c_0 + c_1 s = \Delta m(X) + e(X)$$

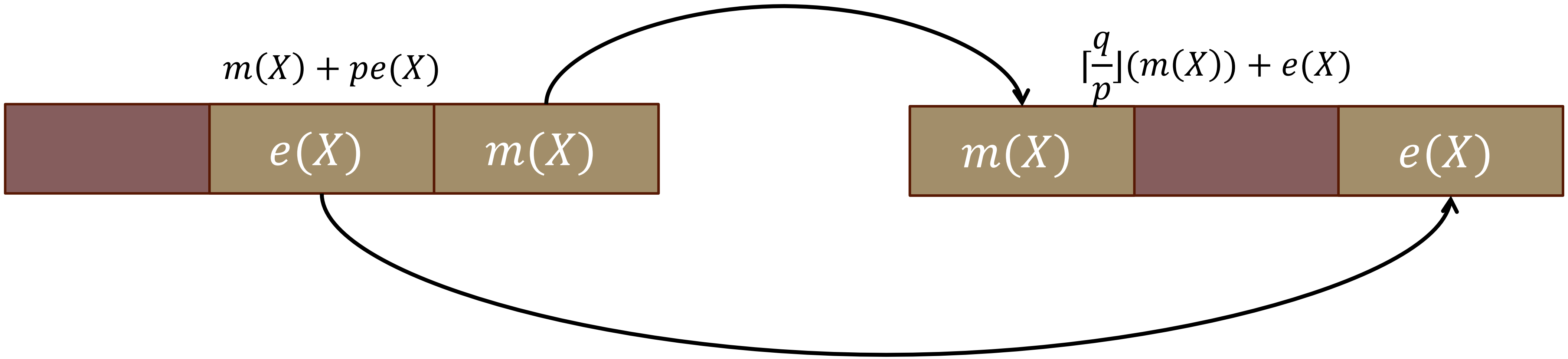


Switching Between BGV and BFV

Switching between BFV and BGV is done via scalar multiplications ([AP13]):

Let p, q be a coprime ciphertext modulus pair, $c_p p + c_q q = 1$ over the integers.

Using this, do a scalar multiplication to switch between BGV to BFV:



This Work:

How hard is it to scheme-switch between BGV/BFV and CKKS?

Can this be done without bootstrapping?

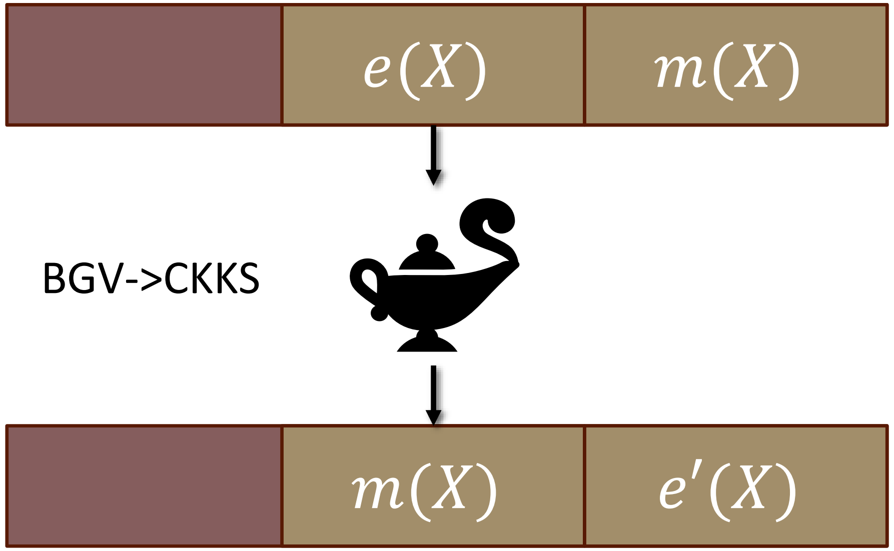
Main Result:

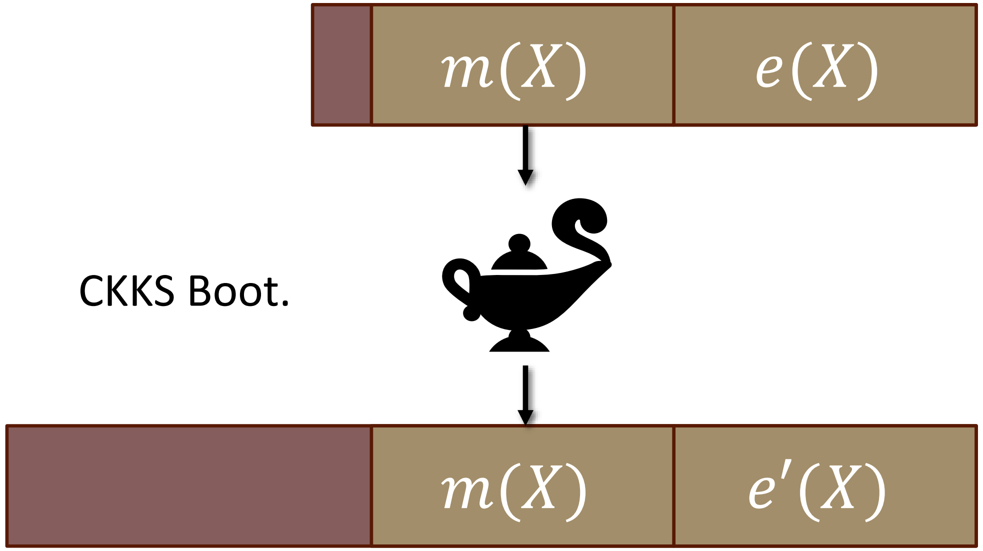
Switching between CKKS and BGV/BFV is as hard as bootstrapping!

Theorem (Informal)

1) If we can scheme-switch from BGV/BFV to CKKS, then we can bootstrap a CKKS ciphertext by running the scheme-switching algorithm and performing one rescaling operation.

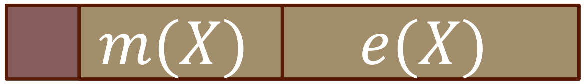
2) Analogously, we can bootstrap BGV/BFV with a CKKS to BGV/BFV oracle call (plus some lightweight ops).





CKKS Bootstrapping

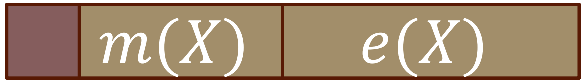
Input: $\mathrm{ct} = (c_0, c_1) \in R_q^2$ with $c_0 + c_1 s = \Delta m(X) + e(X)$ and not much of a gap between $\Delta m(X) + e(X)$ and q.



Output: $\mathrm{ct}' = (c'_0, c'_1) \in R_Q^2$ with $c'_0 + c'_1 s = \Delta m(X) + e'(X)$ with $Q \gg q$.

CKKS Bootstrapping

Input: An exhausted $\mathrm{ct} = (c_0, c_1) \in R_q^2$



1. Raise the ciphertext modulus to Q. This now decrypts to the following with $I(X)$ having small entries:
$$c_0 + c_1 s = \Delta m(X) + e(X) + I(X)q$$

2. Approximate the $f(y) = y \; mod \; q$ function homomorphically (involves hom. un/packing).

What about CKKS and BGV?

Can we switch without bootstrapping? What would it mean if we could?

Say we can and model this as an oracle:

$$\mathcal{O}_{B \hookrightarrow C}(\cdot; p, \Delta, Q)$$

This would take as input a BGV ciphertext $(c_0, c_1) \in R_Q^2$,

$$c_0 + c_1 s = m(X) + pe(X).$$

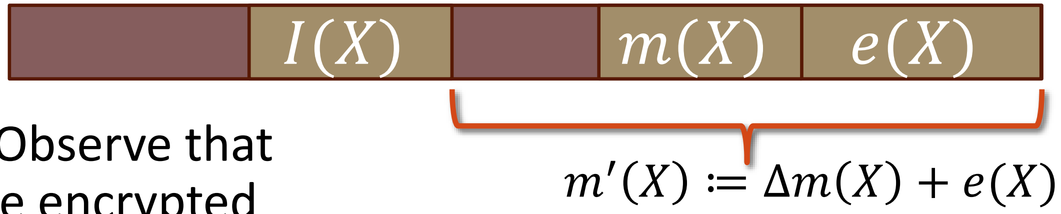
It would return a CKKS ciphertext under the same key: $(c'_0, c'_1) \in R_Q^2$,

$$c'_0 + c'_1 s = \Delta m(X) + e'(X).$$

CKKS Bootstrapping Via Scheme-Switching

Raise the ciphertext modulus to Q. This now decrypts to the following with $I(X)$ having small entries:

$$c_0 + c_1 s = \Delta m(X) + e(X) + I(X)q$$



View this as a BGV ciphertext with plaintext modulus q. Observe that $I(X)$ is the BGV error and $m'(X) \coloneqq \Delta m(X) + e(X)$ is the encrypted message.

Apply $\mathcal{O}_{B \hookrightarrow C}$ to get CKKS ciphertext encrypting

$$\Delta(\Delta m(X) + e(X)) + e'(X)$$

Rescale by Δ to get a CKKS encryption of $\Delta m(X) + e''(X)$

Summary

Additional contributions:

- We define weak scheme-switching and strong scheme-switching (input-output are packed ciphertexts)
- We relate weak and strong scheme-switching.
- We related bootstrapping and homomorphic comparisons (ReLU, max/min, etc.).

Conclusion: switching between BGV/BFV and CKKS is more powerful than bootstrapping since weak-switching is already enough to bootstrap.

Thank You!

https://eprint.iacr.org/2023/988

BGV/BFV and CKKS, the SIMD Schemes

- BGV/BFV and CKKS computations are measured by their multiplicative depth.
- CKKS messages measured by bits of precision.
- Bootstrapping in BGV/BFV and CKKS
  - is slower (minutes) but has high amortized efficiency
  - requires multiplicative depth