Leveled Homomorphic Encryption Schemes for Homomorphic Encryption Standard

Shuhong Gao and **Kyle Yates***

School of Mathematical and Statistical Sciences, Clemson University
2025 Joint Mathematics Meetings

January 11, 2025

*Presenting Author

Overview

This presentation is based on a recent paper. A preprint of our manuscript is available from `https://eprint.iacr.org/2024/991`.

Our goals in this paper:

1. Provide the theoretical mathematical foundations necessary for leveled versions of the schemes outlined in the Homomorphic Encryption Standard (Albrecht et al., 2019 [1]).
2. Propose various theoretical improvements in leveled schemes
    1. Worst-case noise bounds and noise management
    2. Parameter conditions for computation budgeting based on noise expansion

What is Homomorphic Encryption?

Homomorphic encryption allows for **addition** and **multiplication** to be performed on ciphertexts without needing to decrypt or possess any knowledge of private information.

Furthermore, ciphertext computations result in the same output as the corresponding message computations. That is, for messages m_0, m_1 and encryption key $\mathtt{k}$,

$$\mathtt{Enc}(m_0, \mathtt{k}) + \mathtt{Enc}(m_1, \mathtt{k}) = \mathtt{Enc}(m_0 + m_1, \mathtt{k})$$
$$\mathtt{Enc}(m_0, \mathtt{k}) \times \mathtt{Enc}(m_1, \mathtt{k}) = \mathtt{Enc}(m_0 \cdot m_1, \mathtt{k})$$

Types of Schemes

We discuss leveled homomorphic encryption schemes which are based on the **Ring Learning With Errors (RLWE) problems**.

Leveled homomorphic encryption schemes are schemes which allow for circuits evaluations of a predetermined (multiplicative) depth.

This differs from fully homomorphic encryption schemes, which allow for evaluation of circuit of arbitrary depth. This can be expensive in practice (especially bootstrapping).

Types of Schemes

Three popular styles of leveled schemes:

1. Brakerski-Fan-Vercauteren (BFV)
2. Brakerski-Gentry-Vaikuntanathan (BGV)
3. Cheon-Kim-Kim-Song (CKKS)

Our paper discusses all three of these schemes. For sake of time constraints, we'll only focus on BFV [2] in this talk.

Our theoretical improvements are focused on noise management and computation, rather than implementations and security. However, we do outline the basics of these in the paper.

Notation

Define $\mathbb{Z}_q$ as the ring of centered representatives $\mathbb{Z}_q := \mathbb{Z} \cap (-q/2, q/2]$.

We define R_n as the ring

$$R_n := \mathbb{Z}[x]/(\Phi(x))$$

where $\Phi(x)$ is an mth cyclotomic polynomial of degree n, a power of two. Namely, $\Phi(x) = x^n + 1$.

$$R_{n,q} := \mathbb{Z}_q[x]/(\Phi(x))$$

Define the expansion factor δ_R of R_n as $\delta_R = \max\left\{ \frac{\|uv\|_\infty}{\|u\|_\infty \|v\|_\infty} : u, v \in R_n \right\}$.

Notation

For any polynomial $f(x) = \sum_{i=0}^{k} a_i x^i$ with $a_i \in \mathbb{R}$, the *infinity norm of* $f(x)$ is defined as

$$\|f(x)\|_\infty = \max\{|a_0|, \ldots, |a_k|\}.$$

The symbols $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ will denote floor and ceiling respectively, whereas $\lfloor \cdot \rceil$ will denote rounding to the nearest integer.

For a set S, we will denote $U(S)$ as a uniform distribution on S. We denote χ_ρ as any probability distribution on R_n, where each coefficient is random in $[-\rho, \rho]$ and independent.

A Standard BFV Public Key Encryption Algorithm

Key Generation:

	$\texttt{Keygen}(q)$
Input:	$q \in \mathbb{N}$.
Output:	$\mathtt{sk} = s \in R_{n,3}$ secret key,
	$\mathtt{pk} = (\mathtt{k}_0, \mathtt{k}_1) \in R_{n,q}^2$ public key.
Step 1.	Choose randomly $s \in R_{n,3}$.
Step 2.	Sample $\mathtt{k}_0 \leftarrow U(R_{n,q})$ and $e \leftarrow \chi_\rho$.
	Compute $\mathtt{k}_1 := [-(\mathtt{k}_0 s + e)]_{\Phi(x),q}$.
Step 3.	Return $\mathtt{sk} = s$ and $\mathtt{pk} = (\mathtt{k}_0, \mathtt{k}_1)$.

Algorithm: BFV Key Generation

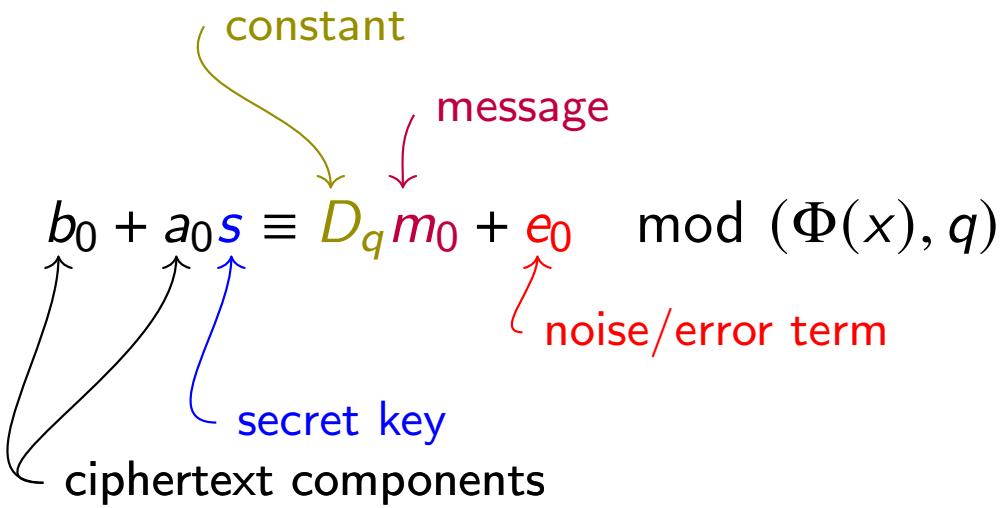
A Standard BFV Public Key Encryption Algorithm

Encryption:

	$\texttt{Encrypt}(m_0, D_q, \texttt{pk})$
Input:	$m_0 \in R_{n,t}$ message,
	$D_q = \lfloor q/t \rfloor \in \mathbb{N}$ constant,
	$\texttt{pk} = (\texttt{k}_0, \texttt{k}_1) \in R^2_{n,q}$ public key.
Output:	$\texttt{ct}_0 = (a_0, b_0) \in R^2_{n,q}$ BFV ciphertext.
Step 1.	Sample $u \leftarrow U(R_{n,3})$ and sample $e_1, e_2 \leftarrow \chi_\rho$.
Step 2.	Compute $(a_0, b_0) \in R^2_{n,q}$ where
	$a_0 := [\texttt{k}_0 u + e_1]_{\Phi(x),q}$,
	$b_0 := [\texttt{k}_1 u + D_q m_0 + e_2]_{\Phi(x),q}$.
Step 3.	Return $\texttt{ct}_0 = (a_0, b_0) \in R^2_{n,q}$.

Algorithm: BFV Encryption

Our BFV ciphertext $\mathtt{ct}_0 = (a_0, b_0) \in R_{n,q}^2$ satisfies the following relationship:



Public: $q, t \in \mathbb{Z}$ with $q \gg t$

$D_q = \lfloor q/t \rfloor \in \mathbb{Z}$

$\Phi(x) \in \mathbb{Z}[x]$ of degree n

$\mathtt{ct}_0 = (a_0, b_0) \in R_{n,q}^2$

Private: $s \in R_{n,3}$

$m_0 \in R_{n,t}$

$e_0 \in R_n$

As homomorphic computation is performed, the **noise term** expands (especially with multiplication procedures).

We introduce a standard technique of modulus reduction to reduce ciphertext noise. This procedure reduces the integer modulus of the ciphertext space, while simultaneously reducing ciphertext noise.

Let $D_Q = \lfloor Q/t \rfloor$ and $D_q = \lfloor q/t \rfloor$, with $Q > q$.

	$\texttt{BFV.Modreduce}(Q, q, \texttt{ct}_0)$
Input:	$Q \in \mathbb{N}$ an integer,
	$q \in \mathbb{N}$ an integer,
	$\texttt{ct}_0 = (a_0, b_0) \in R^2_{n,Q}$.
Output:	$\texttt{ct}'_0 = (a'_0, b'_0) \in R^2_{n,q}$.
Step 1.	Compute $a'_0 := \lfloor \frac{qa_0}{Q} \rceil$ and $b'_0 := \lfloor \frac{qb_0}{Q} \rceil$.
Step 2.	Return $\texttt{ct}'_0 = (a'_0, b'_0) \in R^2_{n,q}$.

Algorithm: BFV Modulus Reduction

Lemma

Suppose the input of `BFV.Modreduce` *is a BFV ciphertext such that* $\|e_0\|_\infty \leq E$. *Let* $\mathtt{ct}'_0$ *be the output of* `BFV.Modreduce`. *If* $t|(Q-1)$ *and* $t|(q-1)$, *then*

$$b'_0 + a'_0 s \equiv D_q m_0 + e'_0 \mod (\Phi(x), q)$$

and $\left\|e'_0\right\|_\infty \leq \frac{q}{Q}E + 1 + \frac{\delta_R \|s\|_\infty}{2}$. *Furthermore, if* $Q/q > \frac{2E}{\delta_R \|s\|_\infty - 2}$, *then* $\left\|e'_0\right\|_\infty < \delta_R \|s\|_\infty$.

Modulus reduction is commonly used to periodically reduce ciphertext noise. Our lemma ensures noise is reduced within a constant bound given conditions on Q/q.

We propose a modified encryption style with a built-in modulus reduction (before message bits are added) to control fresh ciphertext noise.

Key Generation:

	$\texttt{BFV.Keygen}(q, p_0)$
Input:	$q \in \mathbb{N}$,
	$p_0 \in \mathbb{N}$ with $p_0 \geq 5\delta_R + 3$.
Output:	$\mathtt{sk} = s \in R_{n,3}$ secret key,
	$\mathtt{pk} = (\mathtt{k}_0, \mathtt{k}_1) \in R^2_{n,p_0 q}$ public key.
Step 1.	Choose randomly $s \in R_{n,3}$.
Step 2.	Sample $\mathtt{k}_0 \leftarrow U(R_{n,p_0 q})$ and $e \leftarrow \chi_\rho$.
	Compute $\mathtt{k}_1 := [-(\mathtt{k}_0 s + e)]_{\Phi(x), p_0 q}$.
Step 3.	Return $\mathtt{sk} = s$ and $\mathtt{pk} = (\mathtt{k}_0, \mathtt{k}_1)$.

Algorithm: BFV Key Generation

Modified Encryption:

	$\texttt{BFV.Encrypt}(m_0, D_q, \texttt{pk})$
Input:	$m_0 \in R_{n,t}$ message,
	$D_q \in \mathbb{N}$ constant,
	$\texttt{pk} = (\texttt{k}_0, \texttt{k}_1) \in R^2_{n,p_0q}$ public key.
Output:	$\texttt{ct}'_0 = (a'_0, b'_0) \in R^2_{n,q}$ BFV ciphertext.
Step 1.	Sample $u \leftarrow U(R_{n,3})$ and sample $e_1, e_2 \leftarrow \chi_\rho$.
Step 2.	Compute $(a_0, b_0) \in R^2_{n,p_0q}$ where
	$a_0 := [\texttt{k}_0 u + e_1]_{\Phi(x),p_0q}$,
	$b_0 := [\texttt{k}_1 u + e_2]_{\Phi(x),p_0q}$.
Step 3.	Compute
	$(a'_0, b^*_0) := \texttt{BFV.Modreduce}(p_0q, q, (a_0, b_0))$,
	$b'_0 := [b^*_0 + D_q m_0]_q$.
Step 4.	Return $\texttt{ct}'_0 = (a'_0, b'_0) \in R^2_{n,q}$.

Algorithm: Modified BFV Encryption

Bounds on Noise

We derive several new bounds on noise expansion for homomorphic operations. We emphasize that these are worst-case bounds, and are cleaner (but comparable) to existing bounds.

Note we suppose $\rho = \delta_R \|s\|_\infty = n$, $\delta_R \geq 16$, $p_0 \geq 5\delta_R + 3$, $t|(q-1)$ and $t|(Q-1)$. Let E be the worst-case noise bound for each inputted ciphertext in the respective operation.

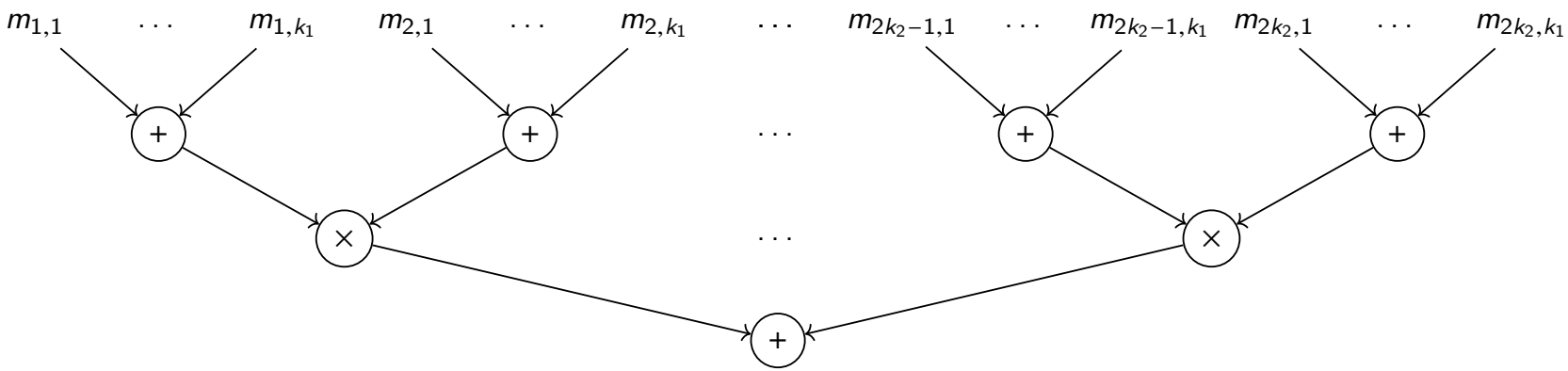
Operation	Noise Bound
Encryption	ρ
LinearCombo	$M(E+1)$
Multiply (initial)	$3.5Et\rho^2$
Multiply (with relinearization)	$3.6Et\rho^2$
ModReduce (if $Q/q > \frac{2E}{\delta_R\|s\|_\infty - 2}$)	ρ

Figure: Noise Bounds. Here, M is the sum of coefficients in absolute value of the linear combination.

Computation Budgeting Results

Definition (Depth-1 Multiplication)

Suppose we have a collection of messages. For fixed k_1 and k_2, we say that we can perform a depth-1 multiplication if we can perform $2k_2$ groups of $k_1 - 1$ additions, followed by one round of k_2 multiplications, followed by $k_2 - 1$ additions.



Algorithm: Plaintext Depth-1 Multiplication

We prove parameter conditions which guarantee a depth-1 homomorphic multiplication for a leveled integer modulus at level i:

$$Q_i = q_0 q_1 \cdots q_i.$$

Furthermore, a following modulus reduction reduces ciphertext noise within a fixed bound.

Lemma

Suppose $q_i > 9k_1k_2tn^2$ and $\delta_R \geq 16$. Then, for a collection of BFV ciphertexts at level i all with noise bounded by ρ, we can homomorphically compute a depth-1 multiplication, followed by a modulus reduction of $Q_i/Q_{i-1} = q_i$. The result is a BFV ciphertext at level $i-1$ with noise bounded by ρ.

Other Results in Our Paper

We outline various other topics and results in our paper:

- Modified encryptions and noise improvements for BGV and CKKS
- Depth-1 parameter conditions and lemmas for BGV and CKKS
- Theoretical security analysis

Slide References:


Martin Albrecht, Melissa Chase, Hao Chen and Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai and Vinod Vaikuntanathan. *Homomorphic encryption standard.* https://ia.cr/2012/144


Junfeng Fan and Frederik Vercauteren. *Somewhat practical fully homomorphic encryption.* https://ia.cr/2012/144

A full list of references is available in our preprint at `https://eprint.iacr.org/2024/991`.