Computation Budgeting in Leveled Homomorphic Encryption Schemes

Kyle Yates

Data Security and Machine Learning Research Workshop

March 2, 2024

What is Homomorphic Encryption?

Homomorphic encryption allows for **addition** and **multiplication** to be performed on ciphertexts (ie, encrypted messages) without needing to decrypt or posess any knowledge of private information.

Furthermore, ciphertext computations result in the same output as the corresponding plaintext computations. That is, for messages m_0, m_1 and encryption key k,

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

Applications in Secure Cloud Computing

- Cloud computing has become increasingly popular
  - Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP)
- AWS - Encryption services provided in transit and at rest.
- Homomorphic encryption allows us to maintain encryption during computation.

Traditional cloud storage and computation

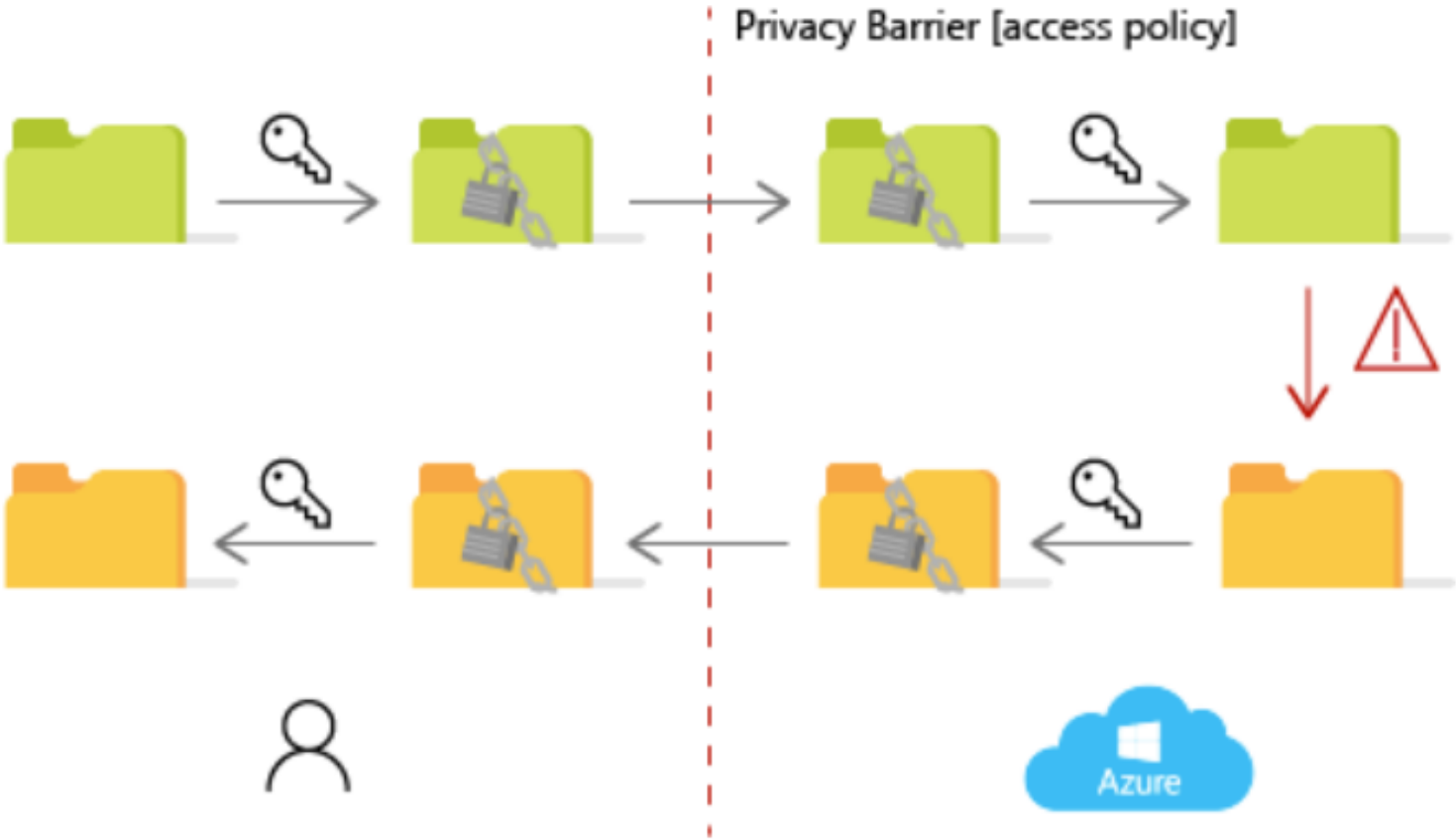


Figure from https://www.microsoft.com/en-us/research/project/microsoft-seal/

Microsoft SEAL cloud storage and computation

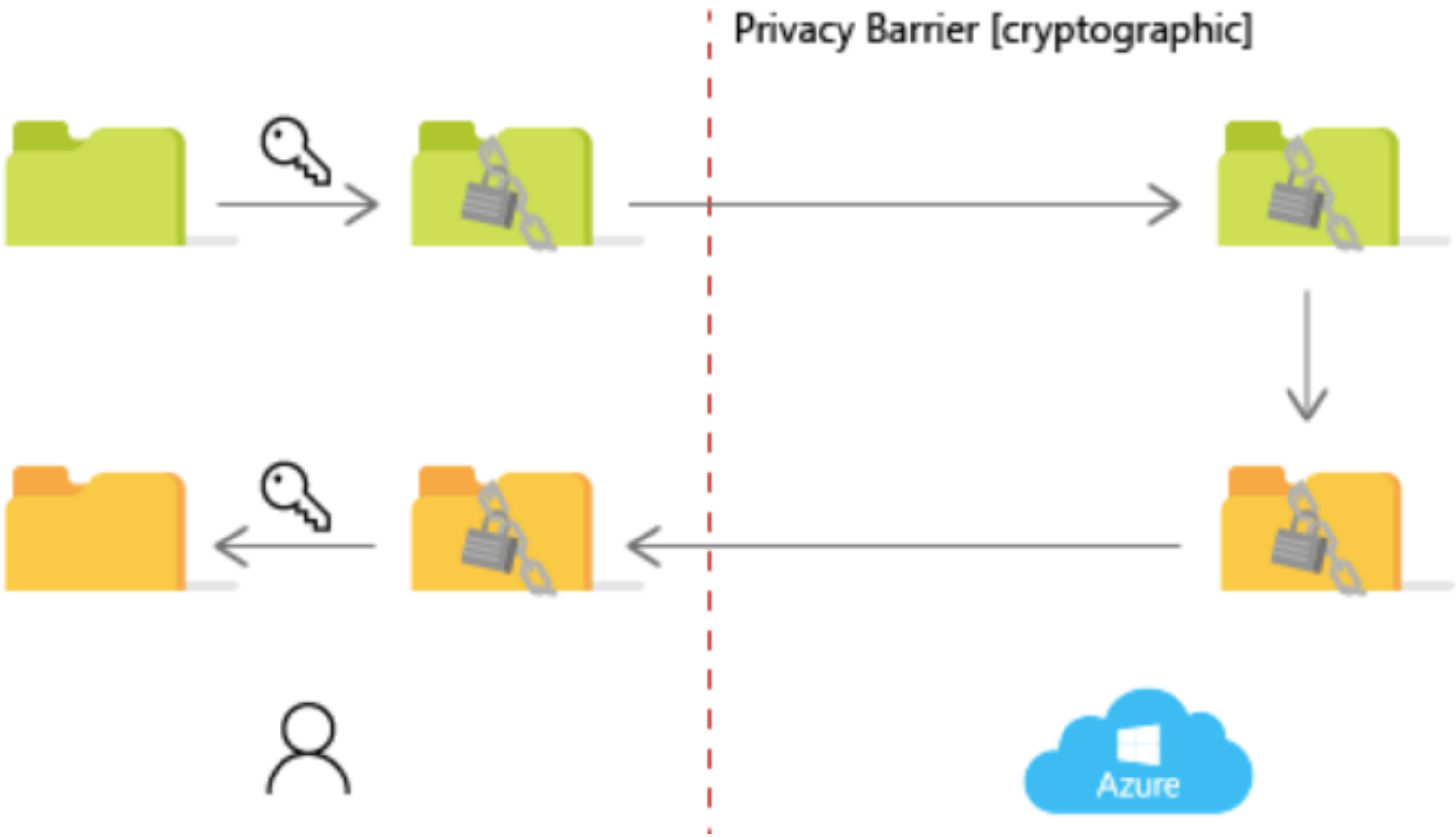


Figure from https://www.microsoft.com/en-us/research/project/microsoft-seal/

Today's talk will cover

1. Ciphertext structures
2. Ciphertext noise growth
3. Noise management techniques & computation budgeting

We will **not** discuss how the actual homomorphic operations work, but the results today are important to budgeting homomorphic computation.

Three main schemes in homomorphic encryption: BFV and BGV (exact), and CKKS (approximate). We will focus on the **BFV** scheme.

Notation

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

When given an integer x, we denote $[x]_q$ as the reduction of x into the interval $\mathbb{Z}_q$ such that q divides $[x]_q - x$. When x is a polynomial or vector, $[x]_q$ means applying $[\cdot]_q$ to each coefficient.

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, $\Phi(x) = x^n + 1$.

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

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

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

therefore using centered representatives, for any $f(x) \in R_{n,q}$ we have $\|f(x)\|_\infty \leq q/2$.

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 and a given probability distribution χ, we let $\chi(S)$ denote that distribution on S. We will denote $U(S)$ as a uniform distribution on S.

Keys and Messages

We'll only look at the private key version of BFV, but a public key version is achievable with some simple modifications.

The secret key is chosen as some $\mathtt{sk} = s \in R_{n,3}$, which is a polynomial with coefficients in $\{-1, 0, 1\}$.

The messages are of the form $m_0 \in R_{n,t}$, where t is a positive integer $\ll q$.

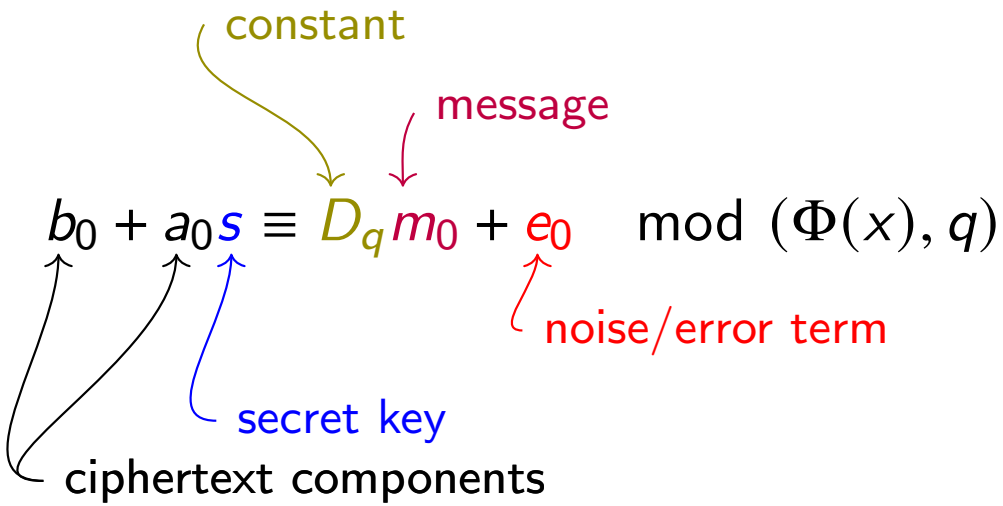
BFV Private Key Encryption

Consider a message $m_0 \in R_{n,t}$. We then encrypt the message using the secret key $\mathtt{sk} = s \in R_{n,3}$, a constant $D_q = \lfloor q/t \rfloor \in \mathbb{Z}^+$, and a chosen parameter $\rho \in \mathbb{Z}^+$ as follows:

1. Sample $e_0 \leftarrow \chi(R_n)$ such that $\|e_0\|_\infty \leq \rho$.
2. Sample $a_0 \leftarrow U(R_{n,q})$.
3. Compute $b_0 \in R_{n,q}$ via $b_0 = -a_0 s + D_q m_0 + e_0 \mod (\Phi(x), q)$.
4. Return $\mathtt{ct}_0 = (a_0, b_0)$.

The ordered pair $\mathtt{ct}_0 = (a_0, b_0) \in R_{n,q}^2$ is our BFV ciphertext.

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

$m_0 \in R_{n,t}$

$e_0 \in R_n$

This format is special because we can define addition and multiplication on ciphertexts. That is, given $\texttt{ct}_0$ and $\texttt{ct}_1$, we can compute either of

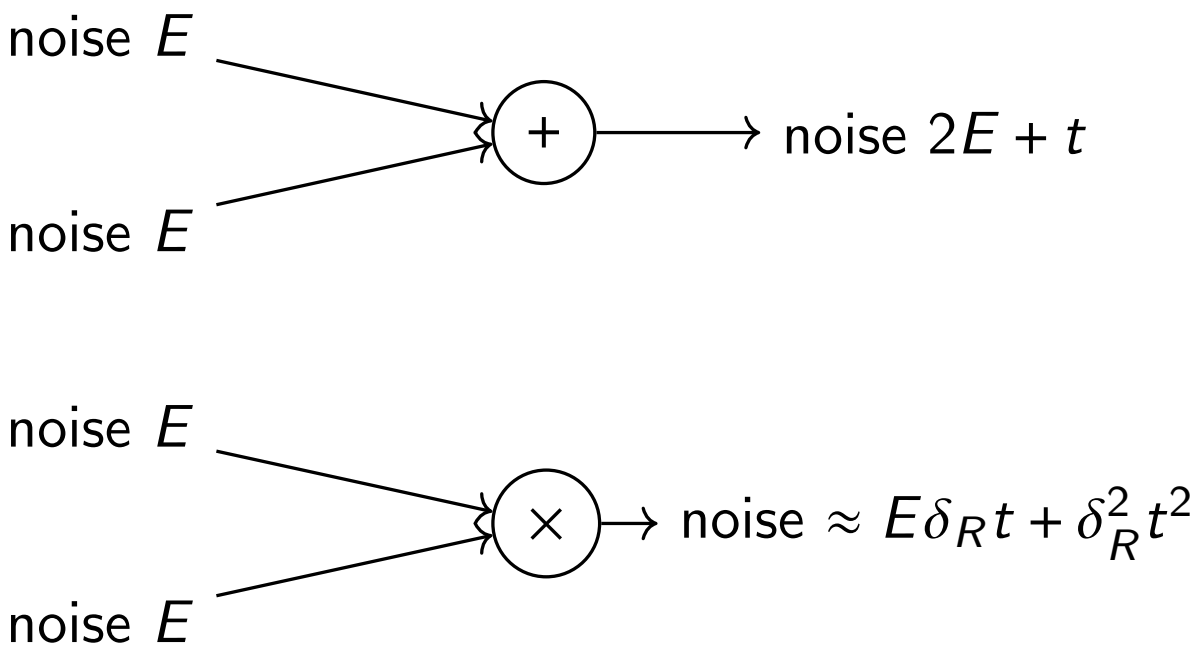
$$\texttt{ct}_0 + \texttt{ct}_1$$

$$\texttt{ct}_0 \times \texttt{ct}_1$$

The main issue with ciphertext operations: as we perform ciphertext operations, the noise term grows. If noise gets too big, we can no longer accurately decrypt our message.

Suppose $\mathtt{ct}_0, \mathtt{ct}_1$ are BFV ciphertexts with noise terms e_0, e_1 respectively such that $\|e_0\|_\infty, \|e_1\|_\infty \leq E$.

Ciphertext Noise Growth in Operations



Here, $\delta_R = \max\{\frac{\|a \cdot b\|_\infty}{\|a\|_\infty \cdot \|b\|_\infty} : a, b \in R_n\}$ is the expansion factor of R_n.

If ciphertext noise grows too much, we can no longer decrypt. The question becomes: **How can we manage this noise?**

We use a technique called **modulus reduction** to manage the growth of ciphertext noise.

Modulus Reductions (Noise Control)

The idea behind modulus reductions:

- Take a ciphertext $\mathtt{ct}_0 \in R^2_{n,Q}$ for some integer ciphertext modulus Q.
- Scale down $\mathtt{ct}_0$ into $R^2_{n,q}$ for some $q < Q$ while preserving the message and structure of the format.

Modulus Reductions (Noise Control)

For the following, let $D_Q = \lfloor Q/t \rfloor$ and $D_q = \lfloor q/t \rfloor$ with $Q > q$.

Input: $Q \in \mathbb{Z}^+$ an integer, $q \in \mathbb{Z}^+$ an integer, and $\mathrm{ct}_0 = (a_0, b_0) \in R_{n,Q}^2$ a BFV ciphertext satisfying $b_0 + a_0 s \equiv D_Q m_0 + e_0 \mod (\Phi(x), Q)$.

1. Compute $a_0' = \lfloor \frac{q a_0}{Q} \rceil$ and $b_0' = \lfloor \frac{q b_0}{Q} \rceil$.
2. Return $\mathrm{ct}_0' = (a_0', b_0') \in R_{n,q}^2$, a BFV ciphertext satisfying $b_0' + a_0' s \equiv D_q m_0 + e_{\mathrm{MR}} \mod (\Phi(x), q)$ for some e_{MR}.

Questions we have: *Does this actually work? What happens to the noise?*

Let $\epsilon_Q = Q/t - D_Q$, $\epsilon_q = q/t - D_q$, $\epsilon_{a_0} = qa_0/Q - a_0'$, and $\epsilon_{b_0} = qb_0/Q - b_0'$. By assumption, we first note that $(a_0, b_0) \in R_{n,Q}^2$ is a BFV ciphertext. That is,

$$b_0 \equiv -a_0 s + D_Q m_0 + e_0 \mod (\Phi(x), Q)$$

Therefore, there is some $r_Q \in R_n$ such that $b_0 + a_0 s \equiv D_Q m_0 + e_0 + Q r_Q \mod \Phi(x)$. Then,

$$\begin{aligned} b_0' &= \frac{qb_0}{Q} - \epsilon_{b_0} \\ &\equiv -\frac{qa_0 s}{Q} + \frac{qD_Q}{Q} m_0 + \frac{qe_0}{Q} - \epsilon_{b_0} + qr_Q \mod \Phi(x) \end{aligned}$$

Note that as $D_Q = Q/t - \epsilon_Q$, we have that $qD_Q/Q = q/t - q\epsilon_Q/Q$. Since $q/t = D_q + \epsilon_q$, we have $qD_Q/Q = D_q + \epsilon_q - q\epsilon_Q/Q$.

Therefore,

$$\begin{aligned} b_0' &\equiv -\frac{qa_0 s}{Q} + \frac{qD_Q}{Q} m_0 + \frac{qe_0}{Q} - \epsilon_{b_0} + qr_Q \mod \Phi(x) \\ &\equiv -a_0' s + \epsilon_{a_0} s + D_q m_0 + (\epsilon_q - \frac{q\epsilon_Q}{Q}) m_0 + \frac{qe_0}{Q} - \epsilon_{b_0} + qr_Q \mod \Phi(x) \\ &\equiv -a_0' s + D_q m_0 + e_{\mathrm{MR}} \mod (\Phi(x), q) \end{aligned}$$

where

$$e_{\mathrm{MR}} = \frac{qe_0}{Q} + (\epsilon_q - q\epsilon_Q / Q) m_0 - \epsilon_{b_0} + \epsilon_{a_0} s$$

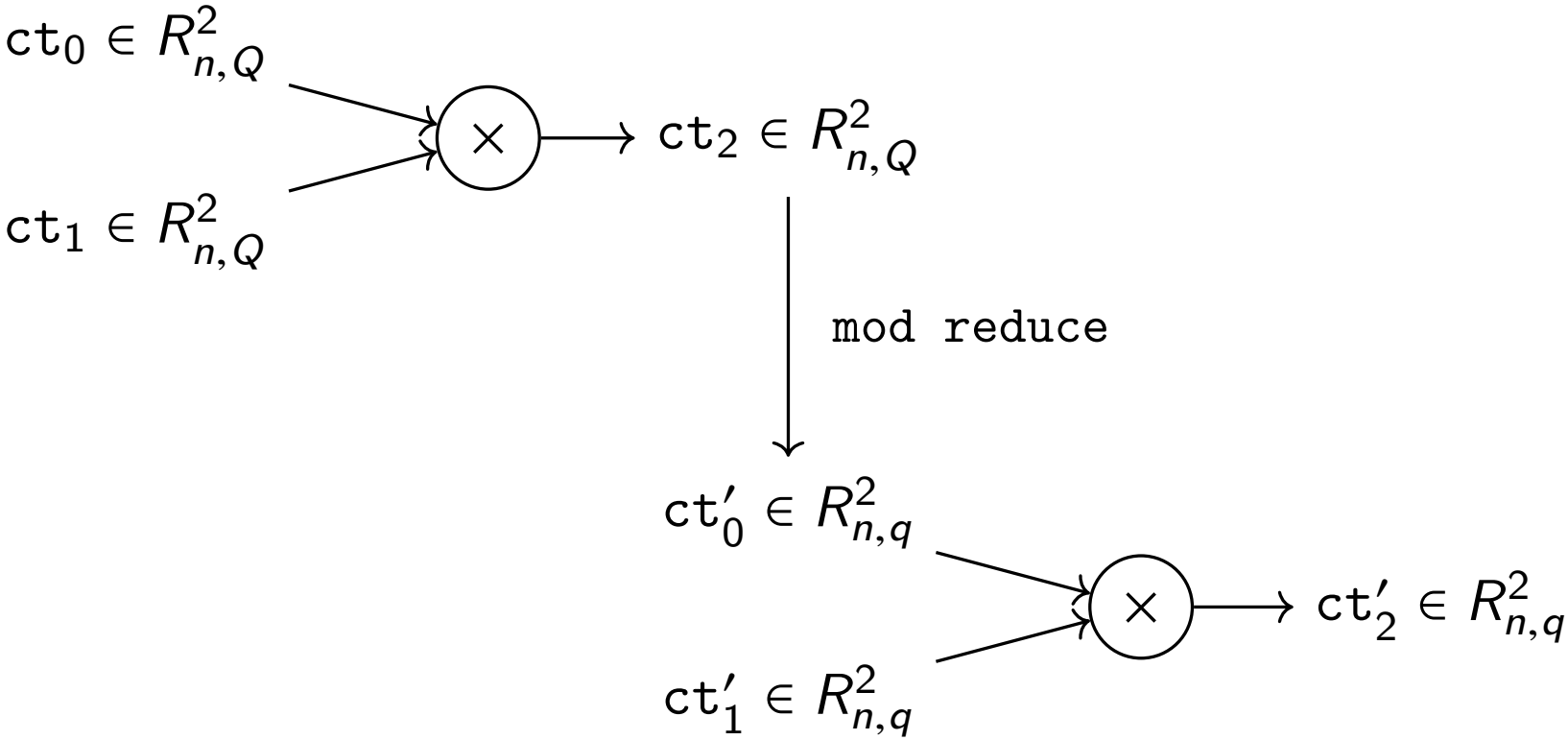
Therefore, $b_0' + a_0' s \equiv D_q m_0 + e_{\mathrm{MR}} \mod (\Phi(x), q)$.

Reducing the modulus reduces the noise!

At the end of the day, we get an noise bound of the following for our new ciphertext after modulus recduction.

$$\|e_{\mathsf{MR}}\|_\infty \leq \frac{q}{Q}E + \frac{t+1}{2} + \frac{\delta_R \|s\|_\infty}{2}$$

Here, E is our initial input noise bound and $Q > q$. We also have $Q, q, t, \delta_R, \|s\|_\infty$ as either constants or predetermined parameters.


$\mathtt{ct}_0 \in R^2_{n,Q}$
$\mathtt{ct}_1 \in R^2_{n,Q}$
$\times$
$\mathtt{ct}_2 \in R^2_{n,Q}$
mod reduce
$\mathtt{ct}'_0 \in R^2_{n,q}$
$\mathtt{ct}'_1 \in R^2_{n,q}$
$\times$
$\mathtt{ct}'_2 \in R^2_{n,q}$

Modulus Leveling

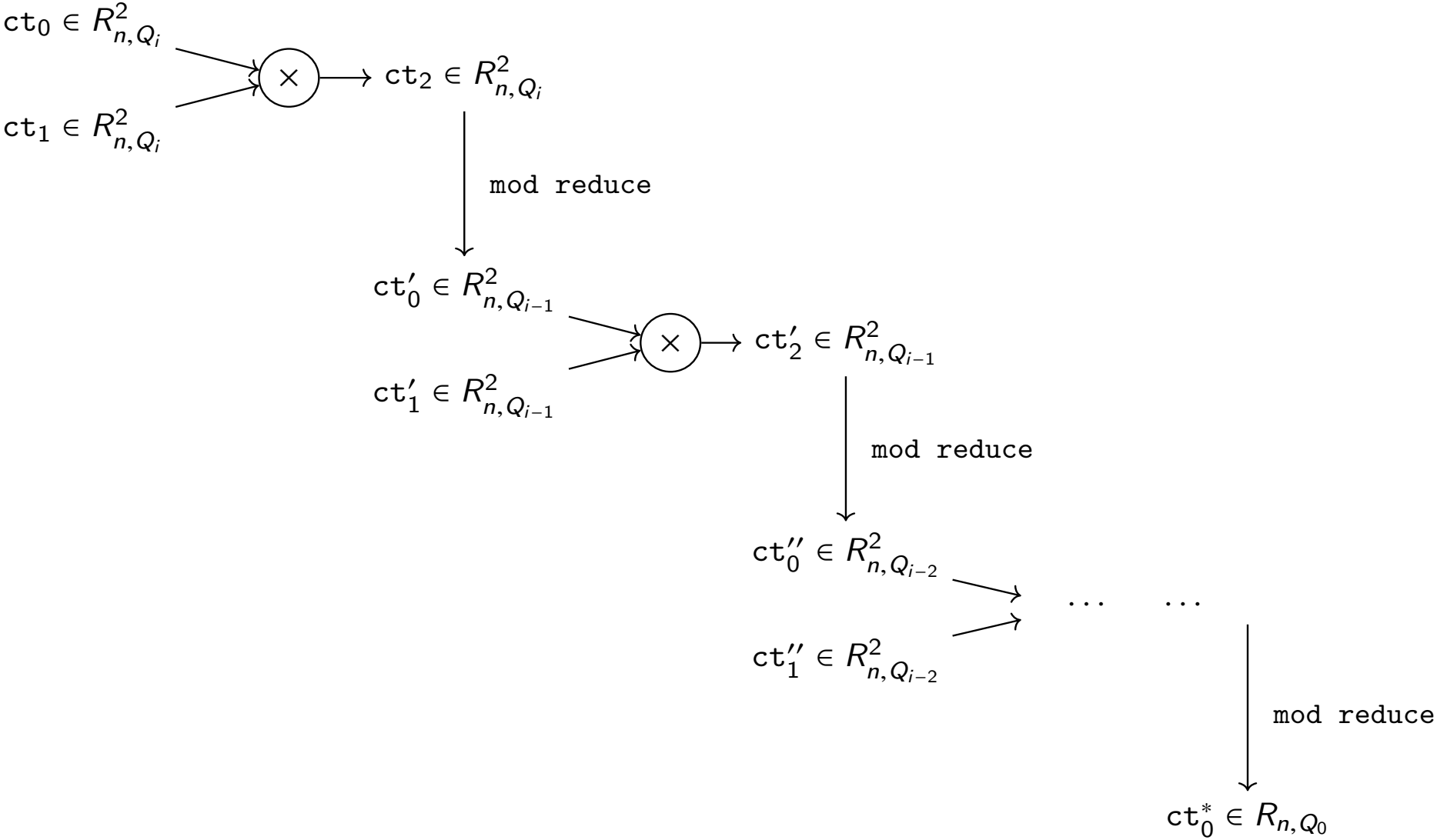
Let $q_{\ell+1} > q_\ell > \cdots > q_0$ be distinct primes, and define $Q_0, \ldots, Q_{\ell+1}$ as

$$Q_i = \prod_{j=0}^{i} q_j$$

We call Q_i the modulus at level i.

$$
\begin{aligned}
Q_{\ell+1} &= q_0 q_1 q_2 \dots q_\ell q_{\ell+1} \\
Q_\ell &= q_0 q_1 q_2 \dots q_\ell \\
&\vdots \\
Q_3 &= q_0 q_1 q_2 q_3 \\
Q_2 &= q_0 q_1 q_2 \\
Q_1 &= q_0 q_1 \\
Q_0 &= q_0
\end{aligned}
$$

Note $Q_i / Q_{i-1} = q_i$ for any $i \in \{1, \dots, \ell + 1\}$. The idea is that we will periodically reduce the modulus from Q_i to Q_{i-1} to reduce the noise.


$\mathtt{ct}_0 \in R^2_{n,Q_i}$
$\mathtt{ct}_1 \in R^2_{n,Q_i}$
$\times$
$\mathtt{ct}_2 \in R^2_{n,Q_i}$
mod reduce
$\mathtt{ct}'_0 \in R^2_{n,Q_{i-1}}$
$\mathtt{ct}'_1 \in R^2_{n,Q_{i-1}}$
$\times$
$\mathtt{ct}'_2 \in R^2_{n,Q_{i-1}}$
mod reduce
$\mathtt{ct}''_0 \in R^2_{n,Q_{i-2}}$
$\mathtt{ct}''_1 \in R^2_{n,Q_{i-2}}$
$\cdots$ $\cdots$
mod reduce
$\mathtt{ct}^*_0 \in R_{n,Q_0}$

Modulus leveling allows for us to construct a **leveled homomorphic scheme**.

This means we have a scheme that allows for some *predetermined number* of addition and multiplication operations.

Improvements and Results

Lemma

Suppose that $t|q-1$ and $\delta_R \geq 16$. For two BFV ciphertexts with integer modulus q noise bound E, their product has noise e_{mult} satisfying

$$\|e_{mult}\|_\infty \leq 4Et\delta_R^2 \|s\|_\infty^2$$

Classic BFV:

$$\|e_{\text{mult}}\|_\infty \leq 2\delta_R tE(1+\delta_R \|s\|_\infty) + 2\delta_R^2 t^2(\|s\|_\infty + 1)^2 \approx E\delta_R t + \delta_R^2 t^2$$

Lemma

Suppose we have a BFV ciphertext with noise bounded by E. Let e_{MR} be the noise term resulting from performing a modulus reduction from Q to q. If $t|Q-1$, $t|q-1$, and $Q/q > \frac{2E}{\delta_R\|s\|_\infty - 2}$, then $\|e_{\text{MR}}\|_\infty < \delta_R \|s\|_\infty$.

Lemma

Suppose $q_i > 10kt\delta_R^2 \|s\|_\infty^2$. *Then, for two vectors of ciphertexts* $\mathbf{u} = (u_1, \ldots, u_k) \in (R^2_{n,Q_i})^k$ *and* $\mathbf{v} = (v_1, \ldots, v_k) \in (R^2_{n,Q_i})^k$, *each with noise bounded by* $\delta_R \|s\|_\infty$, *we can choose to homomorphically compute up to one of the following:*

1. $\sum u_j v_j$ *or*
2. $(\sum u_j)(\sum v_j)$

Furthermore, reducing the modulus by q_i *immediately after this computation always results in a ciphertext with noise bounded by* $\delta_R \|s\|_\infty$.

For the optimized case, this is just $q_i > 10ktn^2$.

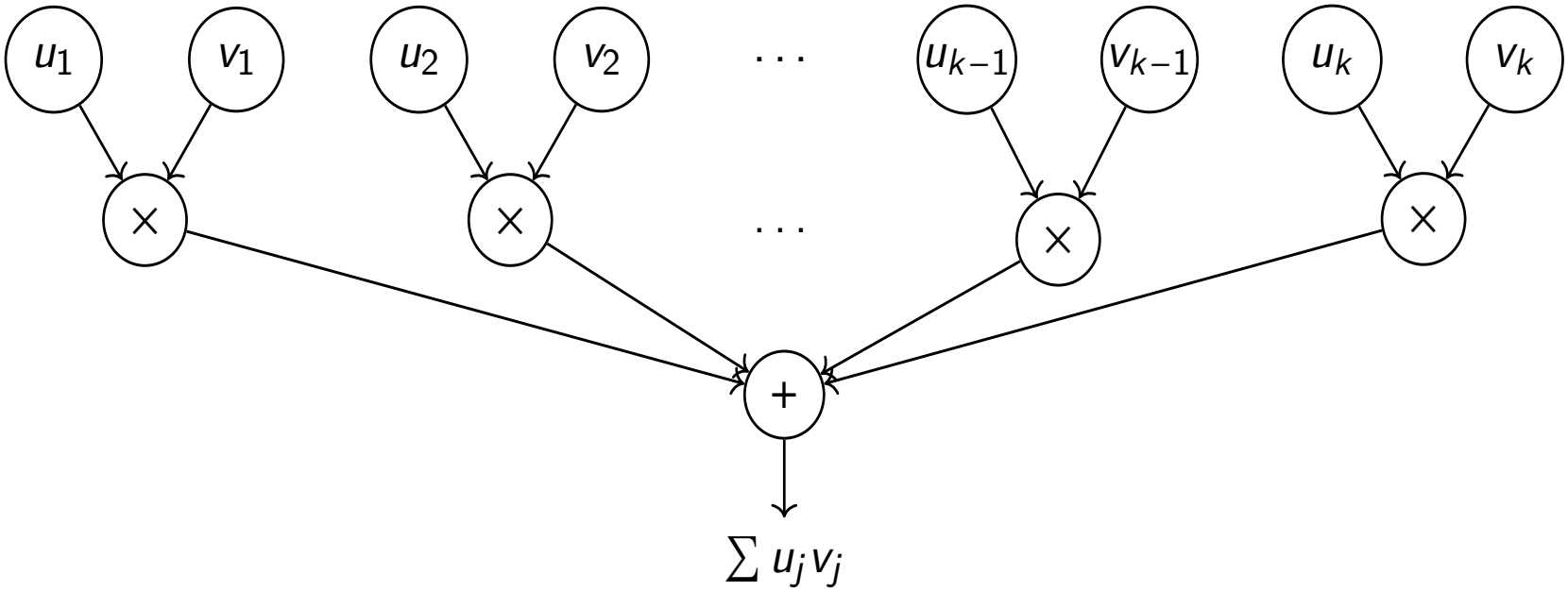


Figure: Homomorphic inner product for each q_i

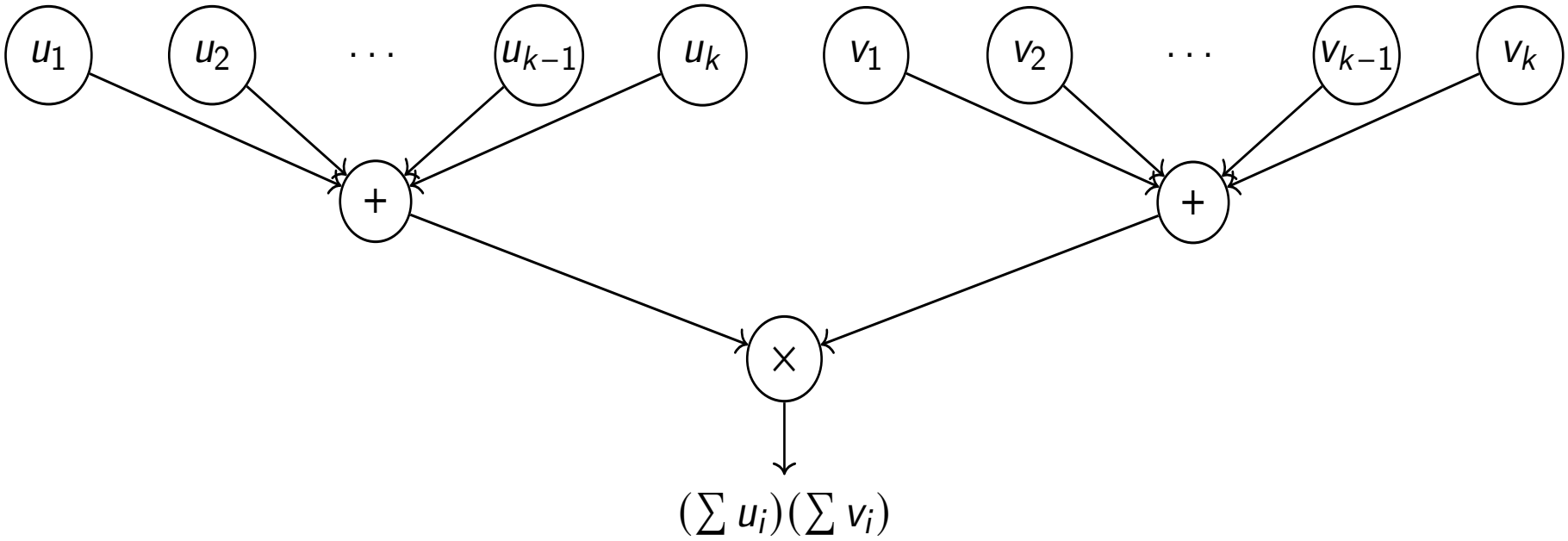


Figure: Homomorphic product of sums for each q_i

Thank You!