HYBRID Key Switching

Reference1

- Revisiting Homomorphic Encryption Schemes for Finite Fields
  In Appendix

- Author : Andrey Kim, Yuriy Polyakov, and Vincent Zucca

- Advances in Cryptology–ASIACRYPT 2021

Reference2

- Better Bootstrapping for Approximate Homomorphic Encryption In Full-RNS decomposition Chapter

- Kyoohyung Han and Dohyeong Ki

- Cryptographers' Track at the RSA Conference, 2020

Base Key Switching

Let $\mathtt{ct}^A = (\boldsymbol{c}_0^A, \boldsymbol{c}_1^A)$ be a ciphertext encrypted modulo $Q \in \{Q_i\}_{i=0}^L$ under a public key $\mathtt{pk}_A$ whose associated secret key is $\mathtt{sk}_A = \boldsymbol{s}_A$, we have

$$\boldsymbol{c}_0^A + \boldsymbol{c}_1^A \cdot \boldsymbol{s}_A \equiv \boldsymbol{m} + t\boldsymbol{v} \bmod Q.$$

It is possible to transform $\mathtt{ct}_A$ into another cipertext $\mathtt{ct}_B$ which will decrypt under a secret key $\mathtt{sk}_B = \boldsymbol{s}_B$. The high level idea is to multiply $\boldsymbol{c}_1^A$ by an encryption of $\boldsymbol{s}_A$ under a public key associated to $\boldsymbol{s}_B$

$$\mathtt{ks}_{A \to B} = \left([\boldsymbol{s}_A + \boldsymbol{a} \cdot \boldsymbol{s}_B + t\boldsymbol{e}]_Q, -\boldsymbol{a} \right) \in \mathcal{R}_Q^2$$

with $\boldsymbol{a} \in\leftarrow \mathcal{U}_Q$ and $\boldsymbol{e} \leftarrow \chi_{\mathtt{err}}$. Then by setting

$$\mathtt{ct}^B = \left([\boldsymbol{c}_0^A + \boldsymbol{c}_1^A \cdot (\boldsymbol{s}_A + \boldsymbol{a} \cdot \boldsymbol{s}_B + t\boldsymbol{e})]_Q, [-\boldsymbol{c}_1^A \cdot \boldsymbol{a}]_Q \right) \in \mathcal{R}_Q^2,$$

s_A Encryption 하는 방식으로 진행

we would have

$$\begin{aligned} \boldsymbol{c}_0^B + \boldsymbol{c}_1^B \cdot \boldsymbol{s}_B &\equiv \boldsymbol{c}_0^A + \boldsymbol{c}_1^A \cdot (\boldsymbol{s}_A + \boldsymbol{a} \cdot \boldsymbol{s}_B + t\boldsymbol{e}) - \boldsymbol{c}_1^A \cdot \boldsymbol{a} \cdot \boldsymbol{s}_B \\ &\equiv \boldsymbol{c}_0^A + \boldsymbol{c}_1^A \cdot \boldsymbol{s}_A + t\boldsymbol{c}_1^A \cdot \boldsymbol{e} \\ &\equiv \boldsymbol{m} + t(\boldsymbol{v} + \boldsymbol{c}_1^A \cdot \boldsymbol{e}) \bmod Q, \end{aligned}$$

which is exactly what we wanted. Unfortunately, this cannot be done directly this way because the added noise $\boldsymbol{c}_1^A \cdot \boldsymbol{e}$ would be too high: $\|\boldsymbol{c}_1^A \cdot \boldsymbol{e}\|_\infty \leq \delta_{\mathcal{R}} Q B_{\mathtt{err}}/2 > \lfloor (Q-t)/2t \rfloor$. Therefore one has to find ways to reduce the size of the product $\boldsymbol{c}_1^A \cdot \boldsymbol{e}$.

Key switching operation을 실행한 후, Decryption을 하게 되면, Decryption 부분에서 coefficient가 m에 거의 근접해야 한다. 하지만, 일반적으로 복호화를 실행하면 $c_1^A * e$에 의해 이 값을 작은 값으로 판단하기는 섣부르다.

Base Key Switching

- Key Switching을 실행한 이후 Decryption을 실행하였을 때,
- Coefficient가 m에 가까워지도록 설정

BV Key Switching

- 계수들을 Base ω를 기준으로 Decomposition을 활용하는 방법

$$\mathcal{D}_{\omega,Q}(\boldsymbol{a}) = \left([\boldsymbol{a}]_\omega, \left[\left\lfloor \frac{\boldsymbol{a}}{\omega} \right\rfloor\right]_\omega, \dots, \left[\left\lfloor \frac{\boldsymbol{a}}{\omega^{\ell_{\omega,Q}-1}} \right\rfloor\right]_\omega\right) \in \mathcal{R}_\omega^{\ell_{\omega,Q}}$$

$$\mathcal{P}_{\omega,Q}(\boldsymbol{a}) = \left([\boldsymbol{a}]_Q, [\boldsymbol{a}\omega]_Q, \dots, [\boldsymbol{a}\omega^{\ell_{\omega,Q}-1}]_Q\right) \in \mathcal{R}_Q^{\ell_{\omega,Q}}$$

D의 길이 : $log_w Q$ * degree of a

Lemma B.1 *For any* $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{R}^2$, $\langle \mathcal{D}_{\omega,Q}(\boldsymbol{a}), \mathcal{P}_{\omega,Q}(\boldsymbol{b}) \rangle \equiv \boldsymbol{a} \cdot \boldsymbol{b} \bmod Q$.

Therefore if we use a key-switching key

$$\mathtt{ks}^{\mathtt{BV}}_{A \to B} = \left(\left[\mathcal{P}_{\omega,Q_L}(\boldsymbol{s}_A) + \vec{\boldsymbol{a}} \cdot \boldsymbol{s}_B + t\,\vec{\boldsymbol{e}}\right]_{Q_L}, -\vec{\boldsymbol{a}}\right) \in \mathcal{R}_{Q_L}^{\ell_{\omega,Q_L}} \times \mathcal{R}_{Q_L}^{\ell_{\omega,Q_L}}$$

s_A를 지수배로 증가

with $\vec{\boldsymbol{a}} \in\leftarrow \mathcal{U}_Q^{\ell_{\omega,Q}}$ and $\vec{\boldsymbol{e}} \leftarrow \chi_{\mathtt{err}}^{\ell_{\omega,Q}}$, we can compute

$$\mathtt{ct}_B = \left(\left[\boldsymbol{c}_0^A + \langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \mathtt{ks}^{\mathtt{BV}}_{A\to B,0} \rangle\right]_Q, \left[\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \mathtt{ks}^{\mathtt{BV}}_{A\to B,1} \rangle\right]_Q\right).$$

Thanks to the linearity of the inner product we obtain in this case

$$\begin{aligned}
\boldsymbol{c}_0^B + \boldsymbol{c}_1^B \cdot \boldsymbol{s}_B &\equiv \boldsymbol{c}_0^A + \langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \mathtt{ks}^{\mathtt{BV}}_{A\to B,0} \rangle + \langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \mathtt{ks}^{\mathtt{BV}}_{A\to B,1} \rangle \cdot \boldsymbol{s}_B \\
&\equiv \boldsymbol{c}_0^A + \langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \mathcal{P}_{\omega,Q}(\boldsymbol{s}_A) \rangle + t\,\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \vec{\boldsymbol{e}} \rangle \\
&\equiv \boldsymbol{c}_0^A + \boldsymbol{c}_1^A \cdot \boldsymbol{s}_A + t\,\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \vec{\boldsymbol{e}} \rangle \\
&\equiv \boldsymbol{m} + t\left(\boldsymbol{v} + \langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \vec{\boldsymbol{e}} \rangle\right) \bmod Q
\end{aligned}$$

with

$$\|\boldsymbol{v}_{\mathtt{BV}}\|_\infty = \|\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \vec{\boldsymbol{e}} \rangle\|_\infty \le \sum_{i=0}^{\ell_{\omega,Q}-1} \left\| \left[\left\lfloor \frac{\boldsymbol{c}_1^A}{\omega^i} \right\rfloor\right]_\omega \cdot \boldsymbol{e}_i \right\|_\infty \le \frac{\ell_{\omega,Q}\delta_{\mathcal{R}}\omega B_{\mathtt{err}}}{2}.$$

복호화 하는 과정에서 다항식이 Decomposition 한 개수만큼 늘어나, Computation Complexity 가 증가

BV Key Switching

B.2.1 Brakerski-Vaikuntanathan

The BV technique can be adapted to RNS by decomposing the values according to each residue as done in [6]

$$\mathcal{D}_{Q_i}(\boldsymbol{a}) = \left(\left[\boldsymbol{a} \left(\frac{Q_i}{q_0} \right)^{-1} \right]_{q_0}, \ldots, \left[\boldsymbol{a} \left(\frac{Q_i}{q_i} \right)^{-1} \right]_{q_i} \right) \in \mathcal{R}^{i+1}$$

$$\mathcal{P}_{Q_i}(\boldsymbol{a}) = \left(\left[\boldsymbol{a} \frac{Q_i}{q_0} \right]_{Q_i}, \ldots, \left[\boldsymbol{a} \frac{Q_i}{q_i} \right]_{Q_i} \right) \in \mathcal{R}_{Q_i}^{i+1}$$

This method was later improved by Halevi, Polyakov, and Shoup [23] who noticed that one could move the $[(Q_i/q_j)^{-1}]_{q_i}$ factors from $\mathcal{D}_{Q_i}(\boldsymbol{c}_1^A)$ to $\mathcal{P}_{Q_i}$, saving therefore $(i+1)$ vector-scalar multiplications.

Lemma B.2 *For any* $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{R}^2$, $\langle \mathcal{D}_{Q_i}(\boldsymbol{a}), \mathcal{P}_{Q_i}(\boldsymbol{b}) \rangle \equiv \boldsymbol{a} \cdot \boldsymbol{b} \bmod Q_i$.

32

$$\begin{aligned} \boldsymbol{c}_0^B + \boldsymbol{c}_1^B \cdot \boldsymbol{s}_B &\equiv \boldsymbol{c}_0^A + \left\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \mathsf{ks}^{\mathsf{BV}}_{A\to B,0} \right\rangle + \left\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \mathsf{ks}^{\mathsf{BV}}_{A\to B,1} \right\rangle \cdot \boldsymbol{s}_B \\ &\equiv \boldsymbol{c}_0^A + \left\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \mathcal{P}_{\omega,Q}(\boldsymbol{s}_A) \right\rangle + t \left\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \vec{\boldsymbol{e}} \right\rangle \\ &\equiv \boldsymbol{c}_0^A + \boldsymbol{c}_1^A \cdot \boldsymbol{s}_A + t \left\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \vec{\boldsymbol{e}} \right\rangle \\ &\equiv \boldsymbol{m} + t \left(\boldsymbol{v} + \left\langle \mathcal{D}_{\omega,Q}(\boldsymbol{c}_1^A), \vec{\boldsymbol{e}} \right\rangle \right) \bmod Q \end{aligned}$$

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

$Q \in \{Q_i\}_{i=0}^{L}$

Decomposition -> $m + t\left(v + \left\| \frac{q_j}{Q_i} \cdot e \right\|\right)$ 형태가 되어, c_1^A의 크기를 줄였다.

(i+1)개의 원소만큼 Decomposition ->복호화 할 때 inner product 단계에서 연산 횟수 quadratic 형태로 이루어진다.

Decomposition을 수행하는 원소의 개수를 줄이는 과정이 필요

GHS Key Switching

- Base Key Switching에서 s_A 큰 P값을 곱하여 $c_1^A * e$를 작게 만드는 방법

B.1.2 Gentry-Halevi-Smart

Another way to reduce the size of the added noise $\boldsymbol{c}_1^A \cdot \boldsymbol{e}$ was proposed by Gentry, Halevi, and Smart in [22]. Their idea was to temporarily extend the size of Q with another modulus P and modify the key-switching key by shifting $\boldsymbol{s}_A$ of P

$$\mathtt{ks}^{\mathtt{GHS}}_{A\to B} = \left([P\boldsymbol{s}_A + \boldsymbol{a}\cdot\boldsymbol{s}_B + t\boldsymbol{e}]_{QP}, -\boldsymbol{a}\right) \in \mathcal{R}^2_{QP}.$$

Then one can perform the product with the key-switching key modulo QP and obtain:

$$\tilde{\mathtt{ct}}_B = \left([\boldsymbol{c}_1^A \cdot (P\boldsymbol{s}_A + \boldsymbol{a}\cdot\boldsymbol{s}_B + t\boldsymbol{e})]_{QP}, [-\boldsymbol{c}_1^A\cdot\boldsymbol{a}]_{QP}\right) \in \mathcal{R}^2_{QP},$$

$$\mathtt{ct}_B = \left(\left[\boldsymbol{c}_0^A + \frac{\tilde{\boldsymbol{c}}_0^B + \boldsymbol{\delta}_0}{P}\right]_Q, \left[\frac{\tilde{\boldsymbol{c}}_1^B + \boldsymbol{\delta}_1}{P}\right]_Q\right),$$

with $\boldsymbol{\delta}_i = t[-t^{-1}\tilde{\boldsymbol{c}}_i^B]_P$, satisfies

$$\begin{aligned}
\boldsymbol{c}_0^B + \boldsymbol{c}_1^B\cdot\boldsymbol{s}_B &\equiv \boldsymbol{c}_0^A + \frac{\boldsymbol{c}_1^A\cdot(P\boldsymbol{s}_A + \boldsymbol{a}\cdot\boldsymbol{s}_B + t\boldsymbol{e}) + \boldsymbol{\delta}_0}{P} + \frac{-\boldsymbol{c}_1^A\cdot\boldsymbol{a} + \boldsymbol{\delta}_1}{P}\cdot\boldsymbol{s}_B \\
&\equiv \boldsymbol{c}_0^A + \frac{\boldsymbol{c}_1^A\cdot(P\boldsymbol{s}_A + \boldsymbol{a}\cdot\boldsymbol{s}_B + t\boldsymbol{e}) + \boldsymbol{\delta}_0}{P} + \left(-\frac{\boldsymbol{c}_1^A\cdot\boldsymbol{a} + \boldsymbol{\delta}_1}{P}\right)\cdot\boldsymbol{s}_B \\
&\equiv \boldsymbol{c}_0^A + \boldsymbol{c}_1^A\cdot\boldsymbol{s}_A + t\frac{\boldsymbol{c}_1^A\cdot\boldsymbol{e}}{P} + \frac{\boldsymbol{\delta}_0 + \boldsymbol{\delta}_1\cdot\boldsymbol{s}_B}{P} \\
&\equiv \boldsymbol{m} + t\left(\boldsymbol{v} + \frac{\boldsymbol{c}_1^A\cdot\boldsymbol{e}}{P} + \frac{\boldsymbol{\delta}_0 + \boldsymbol{\delta}_1\cdot\boldsymbol{s}_B}{tP}\right) \bmod Q.
\end{aligned}$$

with

$$\|\boldsymbol{v}_{\mathtt{GHS}}\|_\infty = \left\|\frac{\boldsymbol{c}_1^A\cdot\boldsymbol{e}}{P} + \frac{\boldsymbol{\delta}_0 + \boldsymbol{\delta}_1\cdot\boldsymbol{s}_B}{tP}\right\|_\infty \le \frac{\delta_{\mathcal{R}} Q B_{\mathtt{err}}}{2P} + \frac{1 + \delta_{\mathcal{R}} B_{\mathtt{key}}}{2}.$$

연산횟수 적지만, P가 엄청 큰 값이어야 하므로, P를 나눠줄 때 multiplicative depth가 크게 감소한다.

GHS Key Switching

smaller complexity for key-switching is required. However, since the security of the scheme depends on the largest ciphertext modulus $\prod_{i=0}^{k-1} p_i \cdot \prod_{i=0}^{L} q_i$, the bit size of $\prod_{i=0}^{k-1} p_i \cdot \prod_{i=0}^{L} q_i$ should be fixed when we assume the same security level.

Security Level에 따라 P*Q의 사이즈가 고정되기 때문에, Key Switching을 할 때, Precision을 높이기 위해 P를 증가시키면 Q가 작아진다.

이는 Multiplicative Depth를 낮추는 효과를 낳는다.

Table 1: Cost model = BKZ.sieve

distribution	n	security level	logq	uSVP	dec	dual
uniform	1024	128	29	131.2	145.9	161.0
		192	21	192.5	225.3	247.2
		256	16	265.8	332.6	356.7
	2048	128	56	129.8	137.9	148.2
		192	39	197.6	217.5	233.7
		256	31	258.6	294.3	314.5
	4096	128	111	128.2	132.0	139.5
		192	77	194.7	205.5	216.4
		256	60	260.4	280.4	295.1
	8192	128	220	128.5	130.1	136.3
		192	154	192.2	197.5	205.3
		256	120	256.5	267.3	277.5
	16384	128	440	128.1	129.0	133.9
		192	307	192.1	194.7	201.0
		256	239	256.6	261.6	269.3
	32768	128	880	128.8	129.1	133.6
		192	612	193.0	193.9	198.2
		256	478	256.4	258.8	265.1

P*Q = logQ

P가 차지하는 bit수, depth가 차지하는 총 bit수 <= q의 비트 수를 만족하여야 한다.

CKKS Relinearization (GHS)

- $Dec(c)Dec(c') = \ (c_0{+}c_1 s)(c_0' + c_1' s) = \ c_0 c_0' + (c_0 c_1' + c_1 c_0')s + c_1 c_1' s^2$

$$= \ \left(c_0 c_0',\ (c_0 c_1' + c_1 c_0')\right) + (\frac{1}{P} * c_1 c_1')((-a_0 s + P s^2, a_0) = evk)$$

더욱 효과적으로 Relinearization을 할 수 있는 방법이 무엇인지에 대해 연구

-> HYBRID Key Switching 제안

HYBRID Key Switching

```
HYBRID key switching takes a number d that's defined modulo Q,
and performs 4 steps:
1 - Digit decomposition:
    Split d into dnum digits - the size of each digit is roughly
    ceil(sizeof(Q)/dnum)
2 - Extend ciphertext modulus from Q to Q*P
    Here P is a product of special primes
3 - Multiply extended component with key switching key
4 - Decrease the ciphertext modulus back down to Q
```

BV

$$
\begin{aligned}
c_0^B + c_1^B \cdot s_B &\equiv c_0^A + \langle \mathcal{D}_{\omega,Q}(c_1^A), \mathrm{ks}^{\mathrm{BV}}_{A\to B,0}\rangle + \langle \mathcal{D}_{\omega,Q}(c_1^A), \mathrm{ks}^{\mathrm{BV}}_{A\to B,1}\rangle \cdot s_B \\
&\equiv c_0^A + \langle \mathcal{D}_{\omega,Q}(c_1^A), \mathcal{P}_{\omega,Q}(s_A)\rangle + t\,\langle \mathcal{D}_{\omega,Q}(c_1^A), \vec{e}\rangle \\
&\equiv c_0^A + c_1^A \cdot s_A + t\,\langle \mathcal{D}_{\omega,Q}(c_1^A), \vec{e}\rangle \\
&\equiv m + t\left(v + \langle \mathcal{D}_{\omega,Q}(c_1^A), \vec{e}\rangle\right) \bmod Q
\end{aligned}
$$

with

$$
\|v_{\mathtt{BV}}\|_\infty = \left\|\langle \mathcal{D}_{\omega,Q}(c_1^A), \vec{e}\rangle\right\|_\infty \leq \sum_{i=0}^{\ell_{\omega,Q}-1} \left\| \left[\left[\frac{c_1^A}{\omega^i}\right]\right]_\omega \cdot e_i \right\|_\infty \leq \frac{\ell_{\omega,Q}\delta_{\mathcal{R}}\omega B_{\mathrm{err}}}{2}.
$$

HYBRID

$$
c_0^B + c_1^B \cdot s_B \equiv m + t\left(v + \frac{\langle \mathcal{D}_{\omega,Q}(c_1^A), \vec{e}\rangle}{P} + \frac{\delta_0 + \delta_1 \cdot s_B}{tP}\right) \bmod Q,
$$

with

$$
\|v_{\mathtt{Hybrid}}\|_\infty = \left\| \frac{\langle \mathcal{D}_{\omega,Q}(c_1^A), \vec{e}\rangle}{P} + \frac{\delta_0 + \delta_1 \cdot s_B}{tP} \right\|_\infty \leq \frac{\ell_{\omega,Q}\delta_{\mathcal{R}}\omega B_{\mathrm{err}}}{2P} + \frac{1 + \delta_{\mathcal{R}} B_{\mathrm{key}}}{2}
$$

GHS

$$
\begin{aligned}
c_0^B + c_1^B \cdot s_B &\equiv c_0^A + \frac{c_1^A \cdot (P s_A + a \cdot s_B + te) + \delta_0}{P} + \frac{-c_1^A \cdot a + \delta_1}{P} \cdot s_B \\
&\equiv c_0^A + \frac{c_1^A \cdot (P s_A + a \cdot s_B + te) + \delta_0}{P} + \left(-\frac{c_1^A \cdot a + \delta_1}{P}\right) \cdot s_B \\
&\equiv c_0^A + c_1^A \cdot s_A + t\frac{c_1^A \cdot e}{P} + \frac{\delta_0 + \delta_1 \cdot s_B}{P} \\
&\equiv m + t\left(v + \frac{c_1^A \cdot e}{P} + \frac{\delta_0 + \delta_1 \cdot s_B}{tP}\right) \bmod Q.
\end{aligned}
$$

$$
\|v_{\mathtt{GHS}}\|_\infty = \left\| \frac{c_1^A \cdot e}{P} + \frac{\delta_0 + \delta_1 \cdot s_B}{tP} \right\|_\infty \leq \frac{\delta_{\mathcal{R}} Q B_{\mathrm{err}}}{2P} + \frac{1 + \delta_{\mathcal{R}} B_{\mathrm{key}}}{2}.
$$

HYBRID Key Switching

- HYBRID key switching takes a number d that's defined modulo Q, and performs 4 steps:
- 1. Digit decomposition: Split d into dnum digits - the size of each digit is roughly ceil(sizeof(Q)/dnum)
- 2. Extend ciphertext modulus from Q to Q*P Here P is a product of special primes
- 3. Multiply extended component with key switching key
- 4. Decrease the ciphertext modulus back down to Q

HYBRID Key Switching

B.2.3 Hybrid

For `Hybrid` key-switching in RNS we use the same methodology and tools as for `BV` and `GHS` techniques. We start by decomposing $\mathbf{c}_1^A$ in d_{num} digits $\tilde{Q}_0, \dots \tilde{Q}_{d_{\text{num}}-1}$, where each digit is the product of α moduli $\tilde{Q}_j = q_{\alpha j} \cdots q_{\alpha(j+1)-1}$ for $\alpha = \lceil (L+1)/d_{\text{num}} \rceil$. Therefore the key-switching key will be:

$$\mathbf{ks}_{A\to B}^{\texttt{RNS-Hybrid}} = \left([P\vec{\mathcal{P}}_{Q_i}(\mathbf{s}_A) + \vec{\mathbf{a}}\cdot \mathbf{s}_B + t\,\vec{\mathbf{e}}]_{PQ_i}, -\vec{\mathbf{a}}\right) \in \mathcal{R}_{PQ_i}^{d_{\text{num}}} \times \mathcal{R}_{PQ_i}^{d_{\text{num}}},$$

with

$$\vec{\mathcal{P}}_{Q_i}(\mathbf{s}_B) = \left(\left[\mathbf{s}_B \frac{Q_i}{\tilde{Q}_0}\right]_{Q_i}, \dots, \left[\mathbf{s}_B\frac{Q_i}{\tilde{Q}_{d_{\text{num}}-1}}\right]_{Q_i}\right) \in \mathcal{R}_{Q_i}^{d_{\text{num}}}.$$

Remark B.3 *Note that the trick used in `HPS` for `BV` key switching equally applies to hybrid key switching, hence the decomposition into d_{num} digits can be obtained for free (without the scalar multiplications).*

Then each digit is extended from $\tilde{\mathcal{Q}}_j = \{q_{\alpha j}, \dots, q_{\alpha(j+1)-1}\}$ to $\mathcal{P} \cup \mathcal{Q}_i$ which causes an overflow $\mathbf{u}_j \tilde{Q}_j$, where $\|\mathbf{u}_j\|_\infty \le (\alpha-1)/2$. As in GHS, the second source of errors comes from the conversion from $\mathcal{P}$ to $\mathcal{Q}_i$ to perform the modulus switching. In this case the overflow will remain the same as in GHS $\|\mathbf{u}'\| \le (k-1)/2$.

Therefore by denoting $\tilde{Q} = \max_{0\le j\le d_{\text{num}}-1}\{\tilde{Q}_j\}$ the noise added by the hybrid key-switching in RNS is bounded by

$$\|\boldsymbol{v}_{\text{RNS-Hybrid}}\|_\infty \le \frac{\alpha d_{\text{num}} \delta_{\mathcal{R}} \tilde{Q} B_{\text{err}}}{2P} + \frac{k + k\delta_{\mathcal{R}} B_{\text{key}}}{2}$$

Thus overall one can take $P \approx \tilde{Q}$ i.e. $k \approx \alpha$.

Remark B.4 *Note that for `BGV` while the moduli q_i must be chosen between 20 and 60 bits depending on the targeted application, the moduli p_i can be chosen of maximal size ≈ 60 bits which should reduce k and hence the computational complexity.*

In OpenFHE

If multiplicative depth is > 3, then dnum = 3 digits are used.

If multiplicative depth is 3, then dnum = 2 digits are used.

If multiplicative depth is < 3, then dnum is set to be equal to multDepth+1

정리

- 1. BV -> Key Switching Base에서 문제인 c_1^A의 계수를 decomposition
- 2. GHS -> BV와는 달리 Decomposition을 하지 않고 큰 값 P로 Dividing
- 3. Hybrid -> BV + GHS, Decomposition의 요소를 줄임 + 적당한 P를 Diving 함으로서 BV, GHS의 단점을 보완했다.