CKKS Error Estimation and Parameters

1 Preliminaries

Let χ is a discrete Gaussian, usually of standard deviation $\sigma = 3.2$. We denote by $\mathcal{ZO}(\rho)$ the distribution where 0 is sampled with probability ρ, and ± 1 are sampled with probability $\rho/2$. We denote the secret key distribution by S. This is usually the unifrom ternary distribution.

We use the notation R_Q to denote the ring $\mathbb{Z}[x]/(\Phi_N, Q)$, where $\phi_N = x^N + 1$. $[\cdot]_Q$ denotes modular reduction $\mod Q$ (usually centered around 0).

2 The CKKS scheme

$\texttt{SecretKeyGen}(\lambda)$: Sample $s \leftarrow S$ and output $\texttt{sk} = (1, s)$.

$\texttt{PublicKeyGen}(\texttt{sk})$: For $\texttt{sk} = (1, s)$, sample $a \leftarrow R_Q$ uniformly at random and $e \leftarrow \chi$. Output $\texttt{pk} = ([-as + e]_Q, a)$.

$\texttt{Encrypt}(\texttt{pk}, m)$: For the message $m \in R$. Let $\texttt{pk} = (p_0, p_1)$, sample $v \leftarrow S$ and $e_1, e_2 \leftarrow \chi$. Output $\texttt{ct} = ([m + p_0 v + e_1]_Q, [p_1 v + e_2]_Q)$.

$\texttt{Decrypt}(\texttt{sk}, \texttt{ct})$: Let $\texttt{ct} = (c_0, c_1)$. Output $m' = [c_0 + c_1 s]_Q$.

Let $\texttt{ct}$ be a fresh ciphertext encrypted under the public key $\texttt{pk}$, where we have $\texttt{pk} = ([-as + e]_Q, a)$. Then, decrypting $\texttt{ct}$ yields

$$\begin{aligned}
\texttt{Decrypt}(\texttt{ct}, \texttt{sk}) &= c_0 + sc_1 \pmod{Q} \\
&= m + p_0 c + e_1 + svp_1 + se_2 \\
&= m + ve + e_1 + se_2.
\end{aligned}$$

Recall $e, e_0, e_1 \leftarrow \chi$. The ephemeral key v here is drawn from the same distribution as the secret key $\texttt{sk} = (1, s)$, but sometimes it can be sampled from a slightly different distribution. This can for example be the distribution $\mathcal{ZO}(\rho)$.

3 CKKS Error Estimation

The number of error bits in fresh ciphertext c_1 encrypting message m_1 using error distribution $N(0, \sigma^2 I_N)$ and secret distribution $\{-1, 0, 1\}$ is given by (Using Central Limit Theory (CLT) and results from [CCH+22]).

$$\epsilon_1 = \frac{1}{2}\log N(\rho^2_{fresh} + \frac{1}{12}) + logH_c(\alpha, N), \tag{1}$$

where

$$\rho^2_{fresh} = (\frac{4}{3}N + 1)\sigma^2$$
$$H_c(\alpha, N) = (-\log(1 - (1-\alpha)^{\frac{2}{N}})^{\frac{1}{2}}.$$

α represents the failure probability or error tolerance.

Similarly, ciphertext c_2 encrypting message m_2 using error distribution $N(0, \sigma'^2 I_N)$ and same secret distribution can be written as

$$\epsilon_2 = \frac{1}{2}\log N(\rho'^2_{fresh} + \frac{1}{12}) + logH_c(\alpha, N), \tag{2}$$

where

$$\rho'^2_{fresh} = (\frac{4}{3}N + 1)\sigma'^2.$$

3.1 Addition

Adding these two ciphertext results into ciphertext with error bits

$$\epsilon_1 + \epsilon_2 = \frac{1}{2}\log N(\rho''^2_{fresh} + \frac{1}{6}) + log2H_c(\alpha, N) \tag{3}$$

with new error $N(0, \rho''^2_{fresh} I_N)$, where

$$\rho''^2_{fresh} = (\frac{4}{3}N + 1)(\sigma^2 + \sigma'^2)$$

3.2 Multiplication by constant

: Multiplying the ciphertext with a constant λ results in a ciphertext with new error $N(0, \rho^2_{mulconst} I_N)$ where

$$\rho^2_{mulconst} = ||\lambda||_2^2(\frac{4}{3}N + 1)\sigma^2 \tag{4}$$

3.3 Multiplication

: Multiplication of two ciphertext results into ciphertext with error of the following form

$$B_{final\ error} = \Delta^{-1}(B_{mult} + B_{ks}) + B_{round} \tag{5}$$

$$
\begin{aligned}
B_{mult} &= N(0, N\rho^2_{fresh}\rho'^2_{fresh} I_N) \\
B_{ks} &= N(0, \eta^2_{ks} I_N) \\
B_{round} &= N(0, \eta^2_{round} I_N)
\end{aligned}
$$

where

$$
\begin{aligned}
\rho^2_{fresh} &= (\frac{4}{3}N + 1)\sigma^2 \\
\rho'^2_{fresh} &= (\frac{4}{3}N + 1)\sigma'^2 \\
\eta^2_{ks} &= \frac{1}{12}p^{-2}q_l^2 N\sigma^2 + 1_{p\nmid q_l}(\frac{N}{18} + \frac{1}{12}) \\
&= \frac{1}{12}N\sigma^2 \qquad [usually \quad p^{-2}q_l^2 \approx 1] \\
\eta^2_{round} &= \frac{N}{18} + \frac{1}{12}
\end{aligned}
$$

The final error of multiplication of two ciphertext can be written down as

$$
\begin{aligned}
B_{final\ error} &= \Delta^{-1}(N(0, N\rho^2_{fresh}\rho'^2_{fresh} I_N) + N(0, \eta^2_{ks} I_N)) + N(0, \eta^2_{round} I_N) \\
&= N(0, \Delta^{-2}N(\rho^2_{fresh}\rho'^2_{fresh} + \frac{1}{12}\sigma^2) I_N) + N(0, (\frac{N}{18} + \frac{1}{12}) I_N) \\
&= N\bigg(0, \Big(\Delta^{-2}N(\rho^2_{fresh}\rho'^2_{fresh} + \frac{1}{12}\sigma^2) + \frac{N}{18} + \frac{1}{12}\Big) I_N\bigg) \\
&= N(0, \rho^2_{mult\ error} I_N) \qquad (6)
\end{aligned}
$$

where

$$
\rho^2_{mult\ error} = \Delta^{-2}N(\rho^2_{fresh}\rho'^2_{fresh} + \frac{1}{12}\sigma^2) + \frac{N}{18} + \frac{1}{12}
$$

4 CKKS parameter

Here is the list of parameter values suggested by the homomorphic encryption standar to retain different security levels.

N	security level	$\log q$	uSVP	dec	dual
1024	128	25	132.6	165.5	142.3
	192	17	199.9	284.1	222.2
	256	13	262.6	423.1	296.6
2048	128	51	128.6	144.3	133.4
	192	35	193.5	231.9	205.2
	256	27	257.1	327.8	274.4
4096	128	101	129.6	137.4	131.5
	192	70	193.7	213.6	198.8
	256	54	259.7	295.2	270.6
8192	128	202	129.8	130.7	128.0
	192	141	192.9	202.5	196.1
	256	109	258.3	276.6	263.1
16384	128	411	128.2	129.5	129.0
	192	284	192.0	196.8	193.7
	256	220	257.2	265.8	260.7
32768	128	827	128.1	128.7	128.4
	192	571	192.0	194.1	193.1
	256	443	256.1	260.4	260.4

Table 1: The differnt parameters shown in the table represents the following: N is the ring dimension, security level is the bit security provided by the parameters equivalent to that of AES bit security. $\log q$ is the number of bits in the modulus q. $uSVP$ represents the bit security against unique shortest vector attack, *dec* represents the bit security against decoding attack and *dual* represnts the bit security against dual attack.

References

[CCH+22] Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. On the precision loss in approximate homomorphic encryption. Cryptology ePrint Archive, Paper 2022/162, 2022. https://eprint.iacr.org/2022/162.