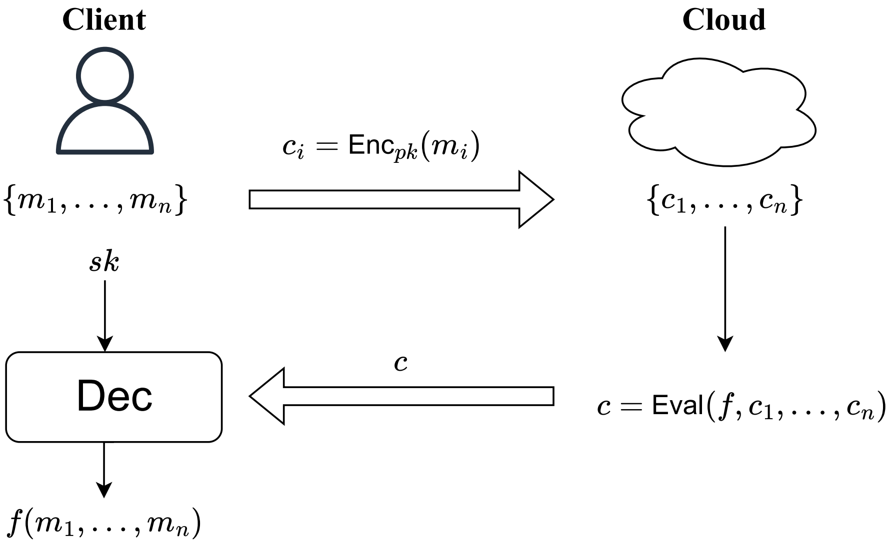
KU LEUVEN

On the overflow and p-adic theory applied to homomorphic encryption

Jacob Blindenbach[1,2], Jung Hee Cheon[3,4], Gamze Gürsoy[1,2], **Jiayi Kang**[5]
[1]Columbia University, [2]NYGC, [3]SNU, [4]CryptoLab, and [5]KU Leuven

CSCML 2024, December 19

Homomorphic Encryption (HE)



Overflow in HE

- The HE plaintext space $\mathcal{P}$ and the message space $\mathcal{M}$ of the client may not be the same.
- The client needs to encode a message into the plaintext space

$$\begin{aligned} \text{Encode} : \mathcal{M} &\longrightarrow \mathcal{P} \\ m &\longrightarrow \text{Encode}(m), \end{aligned}$$

  whose reverse procedure is Decode.
- When $|\mathcal{M}| > |\mathcal{P}|$, *overflow* is a natural phenomenon when performing arithmetics $(\mathcal{M}, +, \times)$ from HE.
- Following [CLPX18, HDRS23], we consider $\mathcal{P} = \mathbb{Z}/q\mathbb{Z}$.

Avoiding Overflows or Tolerating Overflows?

- For the message space $\mathcal{M} = \mathbb{Z}$ or $\mathbb{Q}$,

$$|\mathcal{M}| = \infty > |\mathbb{Z}/q\mathbb{Z}| = q \quad \Longrightarrow \text{overflow}$$

- Previous works [CLPX18, HDRS23] suggest to avoid overflows
  - This leads to larger FHE parameters
- Our work discusses two possibilities of tolerating overflows.

1. Pseudo-overflows do not affect the correctness of the final output, hence do not need to be avoided.
2. When $\mathcal{M} = \mathbb{Z}_p$ (the collection of p-adic integers), the overflow error could be bounded to a desired p-adic precision.

Pseudo-overflows

- If inputs and final outputs are well-bounded, intermediate results can go arbitrarily large without affecting the correctness of the final output.
  - This follows from our lattice interpretation of decoding.

Example

Let $a = 8.3$ and $b = 17$. In computing $f(a, b) = a + b - 16$ using $\mathcal{P} = \mathbb{Z}/(3^{10}\mathbb{Z})$,

- The intermediate result of $f_1(a, b) = a + b$ is too large to be decoded correctly

$$\text{Decode} \circ f_1 \circ \text{Encode}(a, b) = -\frac{10}{233} \neq f_1(a, b) = \frac{253}{10}$$

- The final result is however correct

$$\text{Decode} \circ f \circ \text{Encode}(a, b) = \frac{93}{10} = f(a, b).$$

Overflows in the p-adic arithmetic

- Consider $\mathcal{M} = \mathbb{Z}_p$ being the collection of p-adic integers.
  - Different from Euclidean norm, p-adic norms are ultra-metric

$$|a+b|_p \leq \max\{|a|_p, |b|_p\}, \ \forall a, b \in \mathbb{Q}.$$

  - For $\mathcal{P} = \mathbb{Z}/(p^r\mathbb{Z})$, the overflow error is always bounded by p^{-r} in the p-adic norm.

Example

Recall $\text{Decode} \circ f_1 \circ \text{Encode}(a,b) = -\frac{10}{233} \neq f_1(a,b) = \frac{253}{10}$. Their 3-adic representations are

$$(-\frac{10}{233})_3 = .1000010220120\cdots$$
$$(\frac{253}{10})_3 = .1000010220022\cdots,$$

hence the overflow error is $|\text{Decode} \circ f_1 \circ \text{Encode}(a,b) - f_1(a,b)|_3 = 3^{-10}$.

Implementation and Performance

- Our p-adic encoding and decoding is implemented as a wrapper to the `HElib` library in https://github.com/G2Lab/padicBGV.

$\mathfrak{n}$	$\log_2 Q$	b	t	D_n	D_o	D	$\lvert e\rvert_2$	Method
2^{14}	435	257	—	15	14	14	0	[CLPX18]
		2^{16}	—	11	11	11	0	[HDRS23]
		—	2^8	15	—	**15**	2^{-8}	Ours
2^{15}	890	2^{16}	—	23	16	16	0	[CLPX18]
		2^{16}	—	23	15	15	0	[HDRS23]
		—	2^8	32	—	**32**	2^{-8}	Ours

Table: Comparison of the maximum multiplicative depth D of supported circuits in [CLPX18], [HDRS23] and our p-adic encoding to BGV for input size $L = 2^8$

Conclusion and future works

- Overflows can be tolerated in two aspects
  - pseudo-overflows do not affect the correctness
  - for p-adic arithmetic, the overflow error is small in the p-adic norm
- Under the same ciphertext parameters, tolerating p-adic errors supports circuits up to 2x deeper
- For future works, further investigations of p-adic applications with privacy concerns would be valuable to apply our methods

Thank you for your attention!

ia.cr/2024/1353