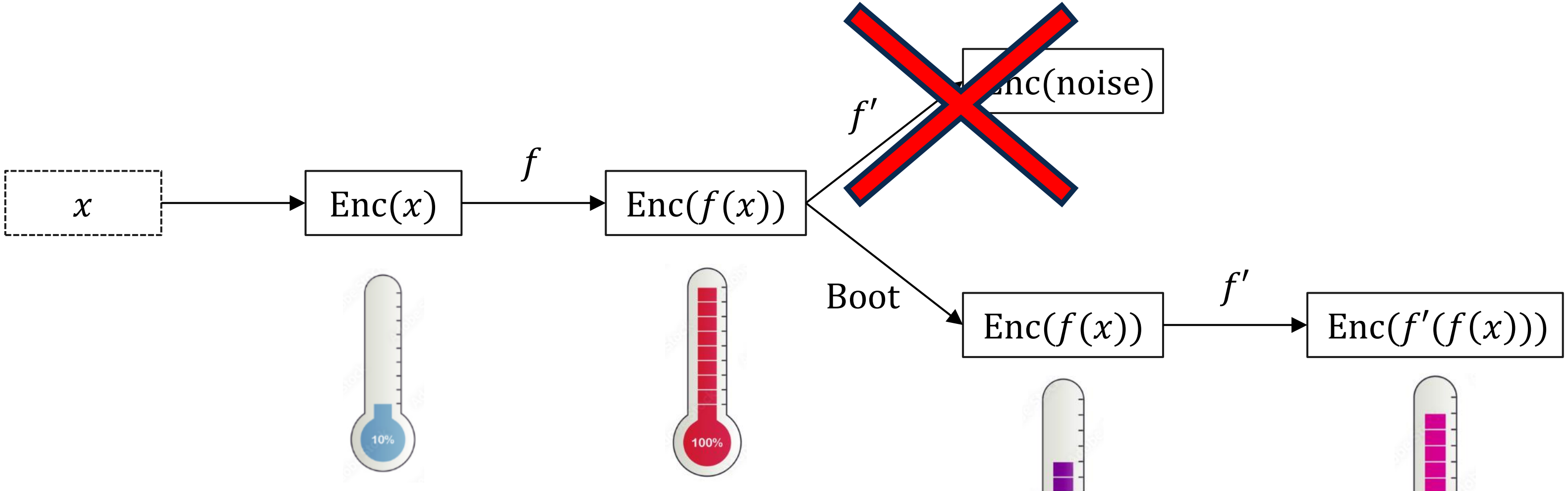
Relaxed Functional Bootstrapping: A New Perspective on BGV/BFV Bootstrapping

Zeyu Liu, Yunhao Wang
Yale University

Yale University
Computer Science

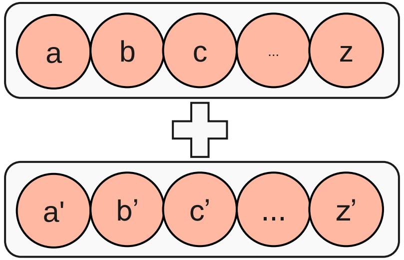
Fully Homomorphic Encryption & Bootstrapping

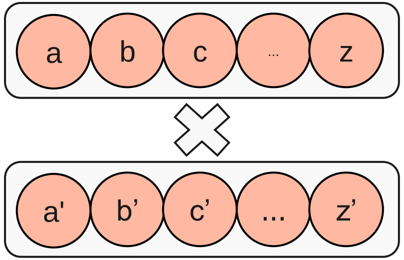


- Very inefficient
  - Bootstrapping circuit is complicated
  - Bootstrapping does nothing else
- Our work:
  - Define and build relaxed functional bootstrapping
    - Allows a more efficient bootstrapping circuit
    - Free function evaluation during bootstrapping

BGV/BFV FHE Scheme

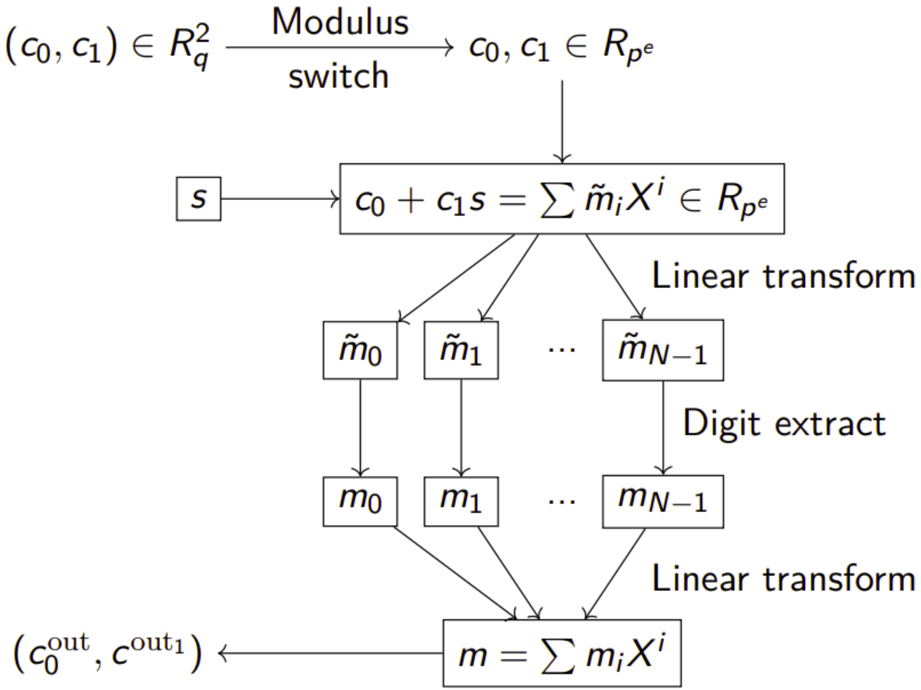
- Ring-LWE based
- Works over rings $R = \mathbb{Z}[X]/\phi_N(X)$ where $\phi_N(X)$ is the N-th cyclotomic polynomial
- Ciphertexts has form $(a, b) \in R_q^2$ for some large q
  - $b = as + e + \lfloor \frac{q}{p^r} m \rfloor$
- Plaintext space R_{p^r} for some p^r
  - $m \in R_{p^r}$
  - If N is a power of 2 and $p \bmod 2N \equiv 1$, plaintext space can be $\mathbb{Z}_p^N$ (encode $\vec{m} \in \mathbb{Z}_p^N$ into $m \in R_p$)
- Allows addition & multiplication
  - For $\mathbb{Z}_p^N$, operations are done element-wise





Past Works in BGV/BFV Bootstrapping

- Temporarily enlarge plaintext space
  - From p^r to p^e for some $e > r$
  - This restricts the choice of p



Relaxed Functional Bootstrapping Definition

- Plaintext space $\mathbb{Z}_p^N$ (operated element-wise)
  - Correctness only guaranteed for $X \subseteq \mathbb{Z}_p$
    - i.e., if $m \in X$, output is $f(m)$ for some pre-defined $f: X \to \mathbb{Z}_p$
  - Output noise budget $>$ input noise budget
    - If f is non-trivial, it can be interesting even if noise budget does not increase
  - Regular bootstrapping is a special case of ours
- Why reasonable?
  - For lots of applications, we know the input in advance.
  - Example 1: after a comparison, the result is always 0/1
  - Example 2: encode the data into X instead of $\mathbb{Z}_p$ for applications like PIR/PSI

Definition 3.1 (Correctness). The bootstrapping procedure is correct, if it satisfies the following: let $(\mathsf{pp} = (N, t, \mathcal{B}_{\mathsf{in}}, \mathcal{B}_{\mathsf{out}}, \mathcal{F}, \mathsf{pp}_{\mathsf{aux}}), \mathsf{sk}, \mathsf{btk}) \leftarrow \mathsf{Setup}(1^\lambda)$, for any function $f : \mathcal{X} \to \mathcal{Y} \in \mathcal{F}$ (where $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{Z}_t$ and $|\mathcal{X}| \geq |\mathcal{Y}| \geq 2$),[8] any honest input ciphertext ct with $\mathcal{B}(\mathsf{sk}, \mathsf{ct}) \geq \mathcal{B}_{\mathsf{in}}$, let $\mathsf{ct}' \leftarrow \mathsf{Boot}(\mathsf{pp}, \mathsf{btk}, f, \mathsf{ct})$, $\vec{m} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \in \mathbb{Z}_t^N$, $\vec{m}' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}') \in \mathbb{Z}_t^N$, it holds that:

$$\Pr\left[\begin{array}{ll} & \forall\, i \in [N],\ \boxed{\text{if } \vec{m}[i] \in \mathcal{X},\ f(\vec{m}[i]) = \vec{m}'[i]} \\ \wedge & \boxed{\mathcal{B}(\mathsf{sk}, \mathsf{ct}') \geq \mathcal{B}_{\mathsf{out}} > \mathcal{B}_{\mathsf{in}}} \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

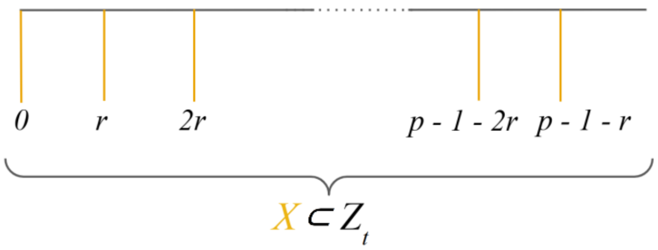
Relaxed Functional Bootstrapping Definition

- What if $m \notin X$?
  - In general, we do not care. This is the core source of our efficiency improvement.
  - However, we define ℓ-closeness
    - If $m \notin X$, output $f(m')$ where $m' \in X$ is one of the ℓ closest elements to m
    - Some of our constructions achieve this for free, some does not achieve it, and some achieves with costs
    - Can be useful for some applications like privacy-preserving machine learning

Definition 3.2 (ℓ-closeness)**.** The bootstrapping procedure is ℓ-close, if it satisfies the following: for the same quantifiers as correctness: for all $x \in \mathbb{Z}_t \setminus \mathcal{X}$, let $y_{x,1}, \ldots, y_{x,|\mathcal{Y}|}$ denote all the points in $\mathcal{Y}$ satisfying $\boxed{|f_x^{-1}(y_{x,1}) - x| \le |f_x^{-1}(y_{x,2}) - x| \le \cdots \le |f_x^{-1}(y_{x,|\mathcal{Y}|}) - x|}$[9][10] and $\mathcal{S}_x := \{y_{x,1}, \ldots, y_{x,\ell}\}$; it holds that for all $i \in [N]$, if $\vec{m}[i] \notin \mathcal{X}$:

$$\Pr\left[f(\vec{m}[i]) \in \mathcal{S}_{\vec{m}[i]}\right] > 1 - \mathsf{negl}(\lambda)^{11}$$

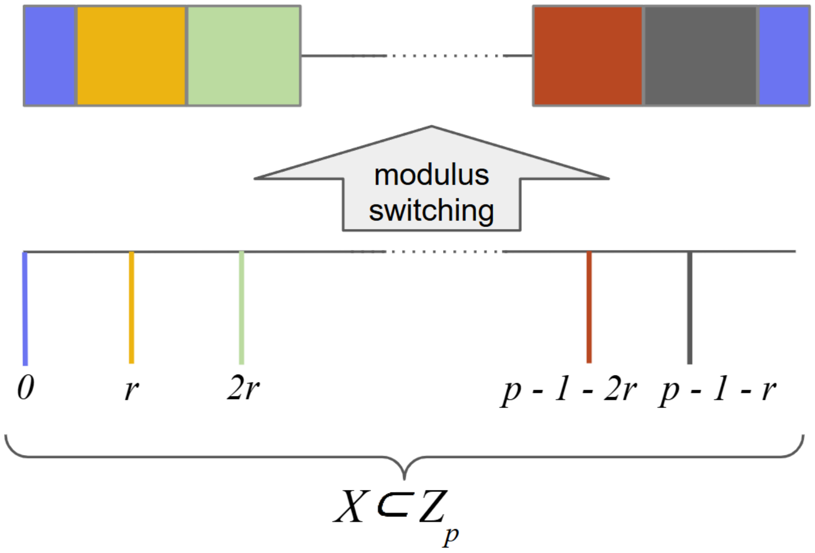
Our starting point (part 1)



- Define $X := [0, p - 1 - r, r]$ (i.e., $(0, r, 2r, \ldots, p - 1 - r)$)
  - r to-be-defined
  - Assume r divides p
    - Otherwise choose r to be the nearest value that divides p
    - Or let $X := [0, p - c - r, r]$ for some integer c such that $r | p - c$
    - Define f to be the identity function
  - i.e., output m if $m \in X$

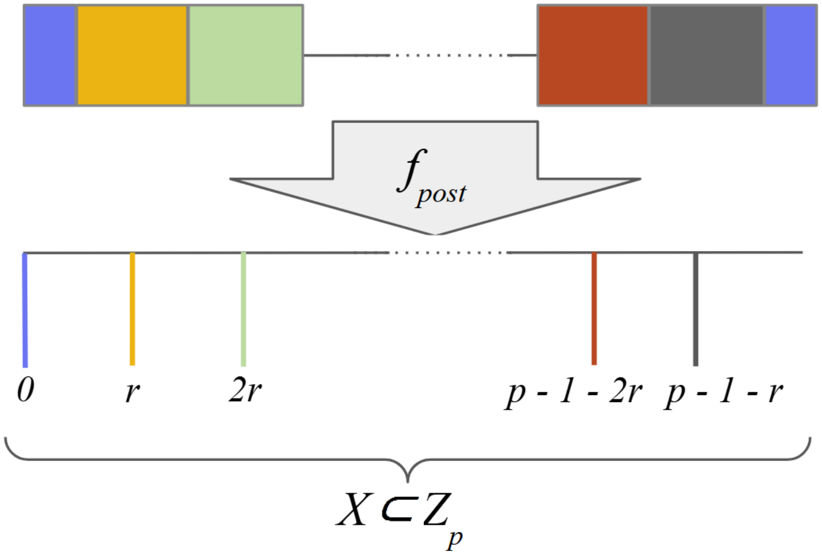
Our starting point (part 1)

- Given ciphertext $(a, b) \in R_q$ encrypting $m \in \mathbb{Z}_p$
- Modulus switching it to $(a', b') \in R_p$ encrypting $m \in \mathbb{Z}_p$
  - If $m \in X$, $m' \in (m - \frac{r}{2}, m + \frac{r}{2})$, where r is the modulus switching error

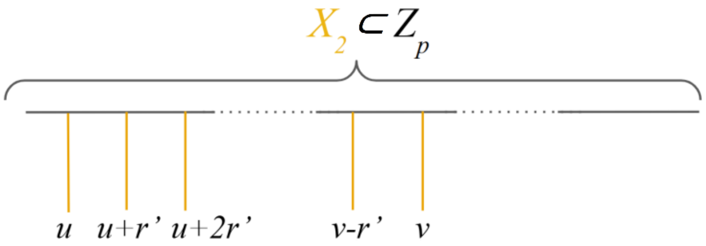


Our starting point (part 1)

- Homomorphically decrypt $(a', b') \in R_p$ to obtain m'
  - Compute $-as + b \in R_p$
  - This can be done either via linear transformation or homomorphic NTT
- Homomorphically compute a function f_{post}
  - Maps $(m - \frac{r}{2}, m + \frac{r}{2})$ to $m, \forall m \in X$
  - This can be done via a degree-$(p-1)$ function
- Correctness is achieved in a straightforward way
- 2-closeness
  - If $m \notin X$, after modulus switching $m' \in (m_1 - \frac{r}{2}, m_2 + \frac{r}{2})$ where $m_1 < m < m_2$ and $m_1, m_2 \in X$

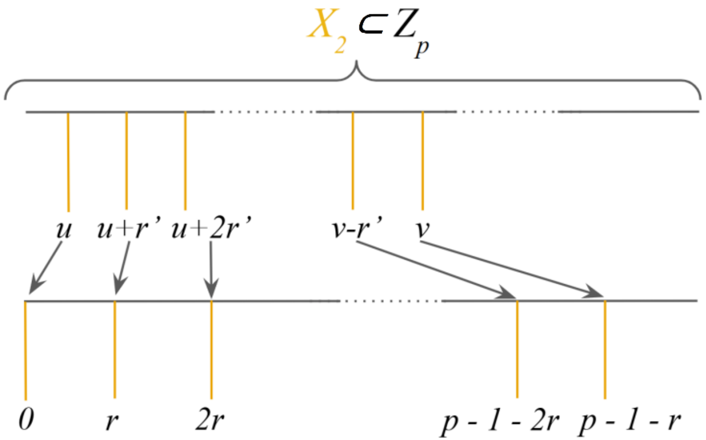


Our starting point (part 2)



- Define $X_2 \coloneqq [u, v, r']$ (i.e., $(u, u + r', u + 2r', \dots, v)$)
  - Arbitrary r'
  - Require $\frac{v - u}{r'} = \frac{p-1}{r}$
- Define f_2 to be an arbitrary function $f_2 \colon X_2 \to \mathbb{Z}_p$
  - i.e., output $f_2(m)$ if $m \in X_2$

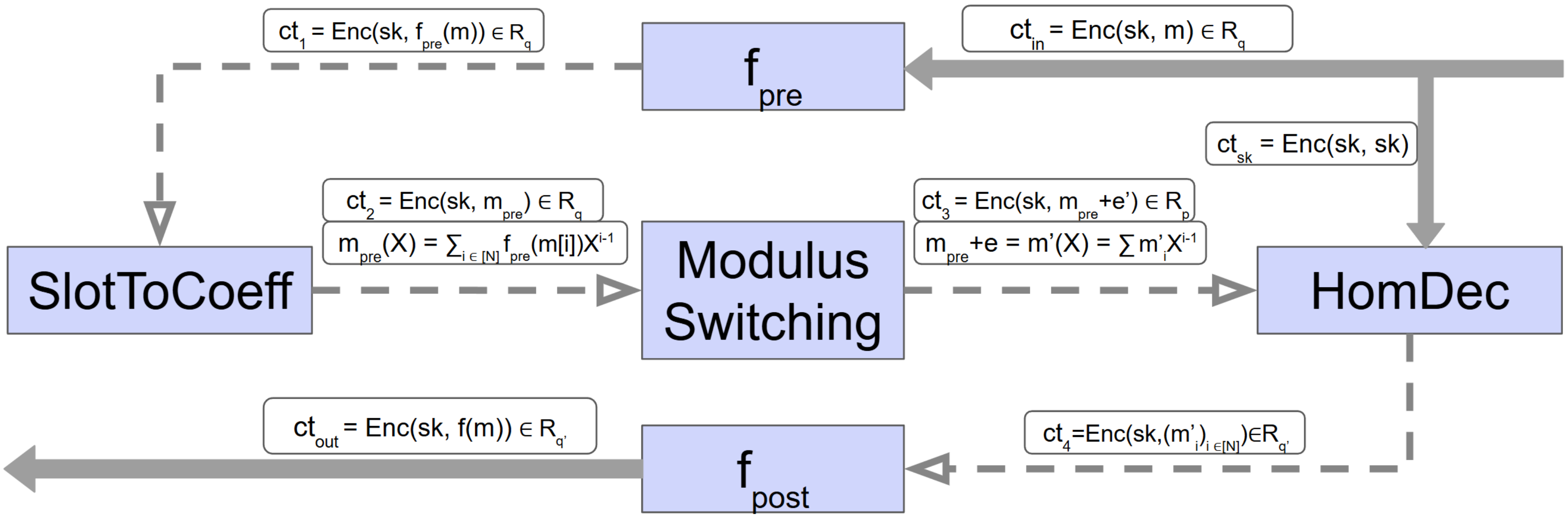
Our starting point (part 2)



- Now it is possible that $r' \leq r$
  - Direct modulus switching may cause error
- First homomorphically evaluate f_{pre}
  - $f_{pre}: X_2 \to X$
  - $f_{pre}(x) := r \cdot (x - u) \cdot r'^{-1}$
  - Only one level of plaintext multiplication
  - Can be merged with SlotToCoeff, so essentially for free
- Then performs modulus switching and homomorphic decryption
- Needs a new post-processing function, since f_2 is no longer an identity function
  - $f_{post,2} = f_2(f_{pre}^{-1}(r \cdot \mathrm{round}(x/r)))$
  - Again, it can be interpolated as a degree $p - 1$ function

Yale University
Computer Science

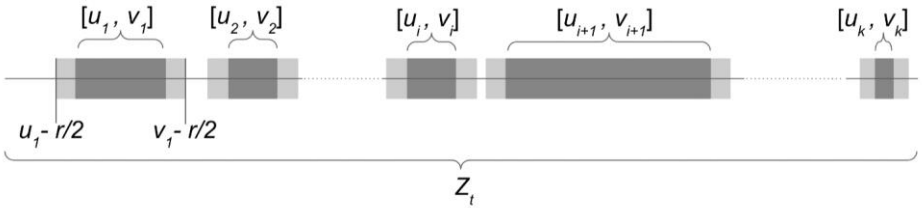
Our general framework



More fine-grained constructions -- multiple points

- Define $X_3 \coloneqq \{y_1, \dots, y_z\}$
  - Arbitrary points, but z $< \frac{p-1}{r}$
- Define f_3 to be an arbitrary function $f_3 \colon X_3 \to \mathbb{Z}_p$
- Naturally, we can define $f_{pre,3}$ to be a map from X_3 to X_1 (the first z elements)
  - However, if $z \ll \frac{p-1}{r}$, a mapping $f_{pre,3'}(x) \coloneqq w \cdot x$ for some $w \in \mathbb{Z}_p^*$ is already sufficient to obtain a result $X_3' \coloneqq f_{pre,3'}(X_3)$ such that every two points in X_3' is separated by r
- $f_{post,3}$ is similar to $f_{post,2}$
  - But only degree $\approx z \cdot r$ instead of $p - 1$

More fine-grained constructions -- multiple ranges



- Define $X_4 \coloneqq \{[u_1, v_1], \ldots, [u_k, v_k]\}$
  - k well-separated ranges
  - i.e., $|u_i - v_j| \geq r, \forall\, i, j \in [k]$
- Define f_4 to be an arbitrary function $\mathrm{f}_4\colon X_4 \to \mathbb{Z}_p$ s.t.
  - Mapping one range to one point
  - i.e., $f_4(x) = y_j$ if $x \in [u_j, v_j]$
- $f_{pre,4}$ is simply an identity function
- $f_{post,4}$ is similar to $f_{post,3}$
  - Except that it now maps $[u_i + \frac{r}{2}, v_i + \frac{r}{2}]$ to y_i
  - It has degree $\approx |X_4| + k \cdot r$

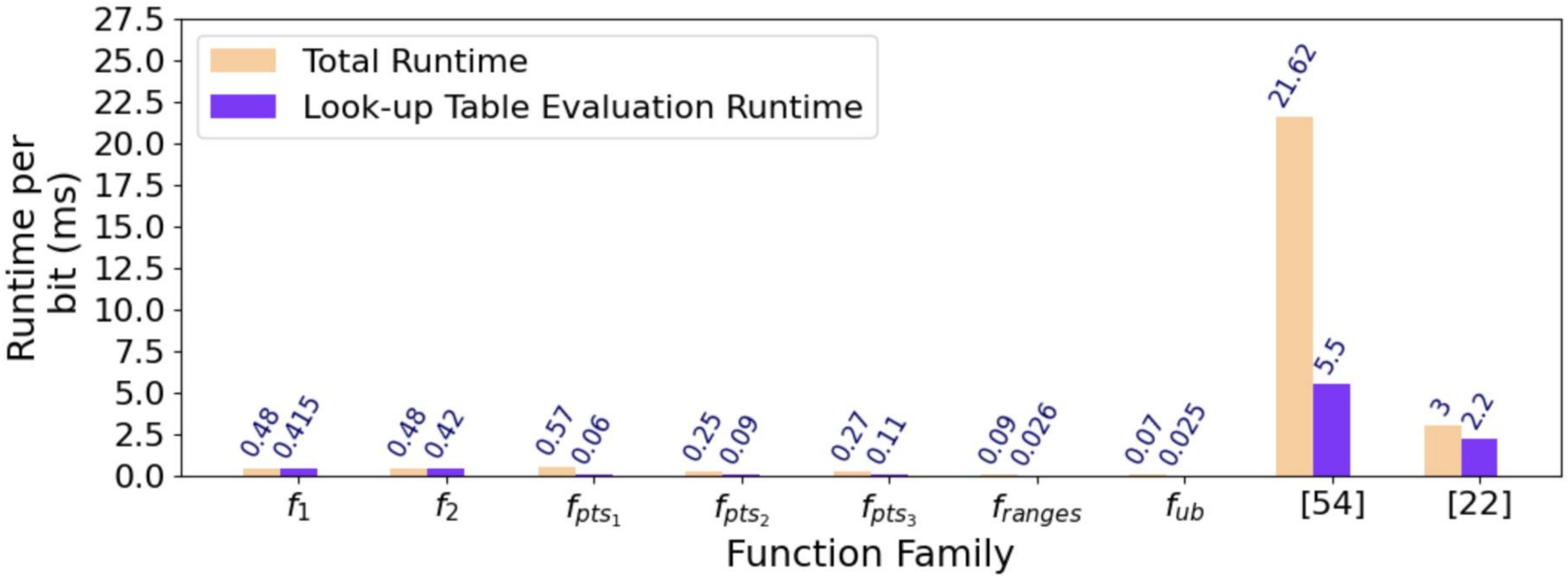
More fine-grained constructions -- unbalanced ranges

- Define $X_5 \coloneqq \{[u_1, v_1], [u_2, v_2]\}$
  - 2 well-separated ranges
  - One being much larger than the other, i.e. $v_2 - u_2 \gg v_1 - u_1$
  - Can be extended to k ranges but preferably one range larger than the others combined
- Define f_5 to be an arbitrary function $\mathrm{f}_5\colon X_5 \to \mathbb{Z}_p$ s.t.
  - Mapping one range to one point
- $f_{pre,5}$ is again simply an identity function
- $f_{post,5}$ first checks if $x \in [u_1, v_1]$
  - If so, maps to y_1, o.w., maps to y_2
  - $f_{post,5}(x) \coloneqq (\prod_{i \in \left[u_1 - \frac{r}{2}, v_1 + \frac{r}{2}\right]} (x - i)^{p-1} \cdot (y_2 - y_1) + y_1)$
  - This can be done in $\approx v_1 - u_1 + \log(p) + r$

Benchmarks

Function Family	Input Domain	# of slots	Ciphertext Modulus	Output Noise Budget	Total Runtime (sec)	**Runtime per slot (ms)**	**Runtime per bit (ms)**
Identity function f_1 over $[0, t-1-r, r]$, *Algorithm* 1	$[0, 65536, 128]$	32768	830	181	142.5	**4.35**	**0.48**
$f_2 : [u, v, r'] \to \mathcal{Y}$ $u, v, r' \in \mathbb{Z}_t$, $\mathcal{Y} \subset \mathbb{Z}_t$, *Algorithm* 2	$[0, 1022, 2]$				142.4	**4.34**	**0.48**
$f_{\mathsf{pts}_1} : \mathcal{X} \to \mathcal{Y}$, $\mathcal{X}, \mathcal{Y} \subset \mathbb{Z}_t$, $\lvert\mathcal{X}\rvert = \lvert\mathcal{Y}\rvert = 2$, *Algorithm* 4	$\{0, 32768\}$		590	198	18.7	**0.57**	**0.57**
$f_{\mathsf{pts}_2} : \mathcal{X} \to \mathcal{Y}$, $\mathcal{X}, \mathcal{Y} \subset \mathbb{Z}_t$, $\lvert\mathcal{X}\rvert = \lvert\mathcal{Y}\rvert = 8$, *Algorithm* 4, without pre-scale on $\mathcal{X}$	$\{57004, 46969, 21931, 39030, 59092, 9965, 30013, 58301\}$		650	194	24.8	**0.76**	**0.25**
$f_{\mathsf{pts}_3} : \mathcal{X} \to \mathcal{Y}$, $\mathcal{X}, \mathcal{Y} \subset \mathbb{Z}_t$, $\lvert\mathcal{X}\rvert = \lvert\mathcal{Y}\rvert = 8$, *Algorithm* 4 with pre-scale on $\mathcal{X}$				181	26.3	**0.80**	**0.27**
$f_{\mathsf{ranges}}(m) = y_i$ if $m \in [u_i, v_i]$, $u_i, v_i, y_i \in \mathbb{Z}_t, i \in [k], k \geq 2$, *Algorithm* 5	Two ranges: $[-63, 63]$ & $[32704, 32831]$		630	205	22.5	**0.69**	**0.09**
$f_{\mathsf{ub}}(m) = y_i$ if $m \in [u_i, v_i]$, $u_i, v_i, y_i \in \mathbb{Z}_t, i \in [2]$, *Algorithm* 6	Two ranges: $[-63, 63]$ & $\mathbb{Z}_{65537} \setminus [-127, 127]$		1070	180	34.3	**1.04**	**0.07**
Regular BFV bootstrapping [59] 128-bit security	$\mathbb{Z}_{257}$	128	881	507	22.0	**173.0**	**21.62**
Regular BFV bootstrapping [25] 66-bit security	$\mathbb{Z}_{127^2}$	2268	1134	330	95.0	**42.0**	**3.00**
Regular BFV bootstrapping [14] 126-bit security	$\mathbb{Z}_{257^2}$	128	806	245	42.0	**328.0**	**20.50**

Benchmarks


Total Runtime
Look-up Table Evaluation Runtime
Runtime per bit (ms)
Function Family
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.48
0.415
0.48
0.42
0.57
0.06
0.25
0.09
0.27
0.11
0.09
0.026
0.07
0.025
21.62
5.5
3
2.2
f_1
f_2
f_{pts_1}
f_{pts_2}
f_{pts_3}
f_{ranges}
f_{ub}
[54]
[22]

Applications

- Oblivious Permutation
  - [FLLP24] proposed a way to do homomorphic permutation
  - Given a database of N bits, the server randomly permutes it without knowing the exact permutation
    - The server performs Thorp shuffle homomorphically, using random bits encrypted under FHE

Oblivious Permutation

1	2	3	4	5	6	7	8

Oblivious Permutation

1	2	3	4	5	6	7	8

Oblivious Permutation

1	2	3	4

5	6	7	8

1	5

Oblivious Permutation

1	2	3	4


5	6	7	8

1	5	6	2

Oblivious Permutation

- This gives a permutation, but not yet random
- Repeat this process $k = O(\lambda)$ times
  - Gives a random permutation except with $1 - \mathrm{negl}(\lambda)$ probability
  - Concretely, $k \cong 400$
- For simplicity, assume these bits are easily samplable under FHE
  - [FLLP] achieves this by building an FHE-friendly PRG, ~0.3ms/bit

1	2	3	4

5	6	7	8

1	5	6	2	5	6	7	8

Oblivious Permutation

- Suitable application for our relaxed bootstrapping
  - ~400 levels
  - Fix some valid input set X, encode every $\log(|X|)$ bits into X
  - The permutation circuit only involves swapping between elements (i.e., input output both in X)
- Using our bootstrapping, the runtime is $> 100\times$ faster than prior works for the bootstrapping part
  - It has extra benefit of allowing more slots, thus in general more efficient

Applications

- Oblivious Permutation
- PIR/PSI/Fuzzy PSI (with computation)
- Secure machine learning
  - Closeness can be preferred
- Of independent interest, our techniques can be used to improve batched FHEW/TFHE bootstrapping

Thank you!

- Open questions
  - Additional function families
  - Other more efficient constructions
  - More applications
- Paper: https://eprint.iacr.org/2024/172.pdf