Bootstrapping BGV ciphertexts with a wider choice of p and q

Emmanuela Orsini, Joop van de Pol, Nigel Smart

Department of Computer Science
University of Bristol

PKC 2015

Motivation

Motivation

- **In the cloud**
  - ◇ Private outsourcing of computation
  - ◇ Near-optimal private outsourcing of storage (single-server PIR) [G09,BV11b]
  - ◇ Verifiable outsourcing (delegation) [GGP11,CKV11,KKR13]
  - ◇ Private machine learning in the cloud [GLN12,HW13]

University of BRISTOL

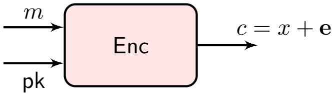
Motivation

- **In the cloud**
  - Private outsourcing of computation
  - Near-optimal private outsourcing of storage (single-server PIR) [G09,BV11b]
  - Verifiable outsourcing (delegation) [GGP11,CKV11,KKR13]
  - Private machine learning in the cloud [GLN12,HW13]
- **Secure Multiparty Computation**
  - Low-communication multiparty computation [AJLTVW12,LTV12,CLOPS13]
  - More efficient MPC [BDOZ11,DPSZ12,DKLPSS12]

Motivation

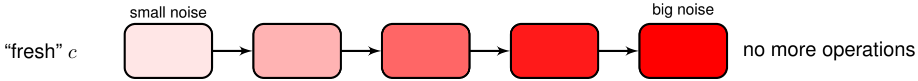
- **In the cloud**
  - Private outsourcing of computation
  - Near-optimal private outsourcing of storage (single-server PIR) [G09,BV11b]
  - Verifiable outsourcing (delegation) [GGP11,CKV11,KKR13]
  - Private machine learning in the cloud [GLN12,HW13]
- **Secure Multiparty Computation**
  - Low-communication multiparty computation [AJLTVW12,LTV12,CLOPS13]
  - More efficient MPC [BDOZ11,DPSZ12,DKLPSS12]
- **Primitives**
  - Succinct argument systems [GLR11,DFH11,BCCT11,BC12,BCCT12,BCGT13,...]
  - General functional encryption [GKPVZ12]
  - Indistinguishability obfuscation for all circuits [GGHRSW13]

How to construct an FHE scheme - Step I



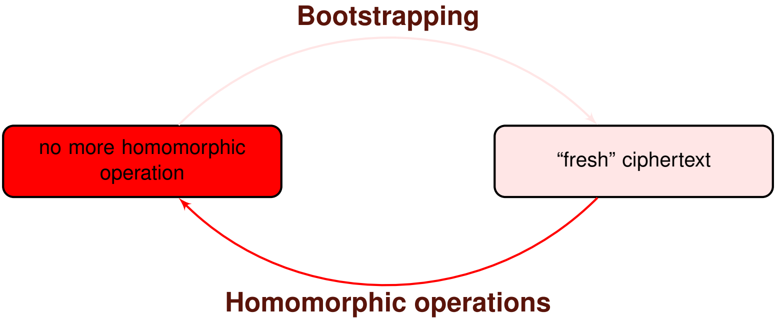
The ciphertext contains an error term $\mathbf{e}$ (noise)

- The noise increases with every homomorphic operation
- A correct decryption is guaranteed if the final noise magnitude is below a certain limit



- **Somewhat Homomorphic Encryption Scheme** : support a limited number of additions and multiplications

Bootstrapping - Step 2



- The bootstrapping step takes as input a ciphertext with a large noise and outputs a “fresh” ciphertext of the *same* plaintext
- It is the only known way of obtaining unbounded FHE

Bootstrapping

- Homomorphically computes the SHE decryption function on encrypted secret key

$$\boxed{c} \qquad \boxed{\mathsf{sk}} \longrightarrow \mathsf{Eval}_f\Big(\mathsf{Dec}(\,\cdot\,, c)\Big) \longrightarrow \boxed{c}$$

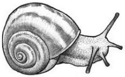
University of BRISTOL

Bootstrapping

◇ Homomorphically computes the SHE decryption function on encrypted secret key

$$\boxed{c} \qquad \boxed{\mathsf{sk}} \longrightarrow \mathsf{Eval}_f\Big(\mathsf{Dec}(\,\cdot\,, c)\Big) \longrightarrow \boxed{c}$$

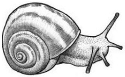
- Still the main bottleneck in FHE



Bootstrapping

◇ Homomorphically computes the SHE decryption function on encrypted secret key

$$\boxed{c} \qquad \boxed{\mathsf{sk}} \longrightarrow \mathsf{Eval}_f\Big(\mathsf{Dec}(\,\cdot\,, c)\Big) \longrightarrow \boxed{c}$$

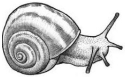


- Still the main bottleneck in FHE
- GOAL: Efficiency! Minimize depth d of decryption circuit

Bootstrapping

◇ Homomorphically computes the SHE decryption function on encrypted secret key

$$\boxed{c} \qquad \boxed{\mathsf{sk}} \longrightarrow \mathsf{Eval}_f\Big(\mathsf{Dec}(\,\cdot\,, c)\Big) \longrightarrow \boxed{c}$$

- Still the main bottleneck in FHE 

- GOAL: Efficiency! Minimize depth d of decryption circuit

- Intensive research area
  [AP13, BV14, AP14, HS14, HAO15, HS15, DM15]

Our result

We present a new bootstrapping technique with:

Our result

We present a new bootstrapping technique with:

- Small depth growth

Our result

We present a new bootstrapping technique with:

- Small depth growth
- Large choice of the parameters of the scheme

We present a new bootstrapping technique with:

- Small depth growth
- Large choice of the parameters of the scheme

Main tools:

◇ Matrix representation of rings

Our result

We present a new bootstrapping technique with:

- Small depth growth
- Large choice of the parameters of the scheme

Main tools:

- ◇ Matrix representation of rings
- ◇ Batch Computation

University of BRISTOL

Our result

We present a new bootstrapping technique with:

- Small depth growth
- Large choice of the parameters of the scheme

Main tools:

- ◇ Matrix representation of rings
- ◇ Batch Computation
- ◇ Ring-switching technique

The BGV ring-LWE-based somewhat homomorphic encryption scheme

We consider the BGV SHE scheme [BGV12]

The BGV ring-LWE-based somewhat homomorphic encryption scheme

We consider the BGV SHE scheme [BGV12]

- We use two rings (at some point we perform a ring-switching)
  - ◇ $R = Z[X]/\Phi_m(X)$
  - ◇ $\deg \Phi_m(X) = N$
  - ◇ $\mathsf{sk}^{(R)}$
  - ◇ $S = Z[X]/\Phi_{m'}(X)$
  - ◇ $\deg \Phi_{m'}(X) = n$
  - ◇ $\mathsf{sk}^{(S)}$

  S is a subring of R $(m'|m)$

The BGV ring-LWE-based somewhat homomorphic encryption scheme

We consider the BGV SHE scheme [BGV12]

- We use two rings (at some point we perform a ring-switching)
  - ◇ $R = Z[X]/\Phi_m(X)$
  - ◇ $\deg \Phi_m(X) = N$
  - ◇ $\mathsf{sk}^{(R)}$
  - ◇ $S = Z[X]/\Phi_{m'}(X)$
  - ◇ $\deg \Phi_{m'}(X) = n$
  - ◇ $\mathsf{sk}^{(S)}$

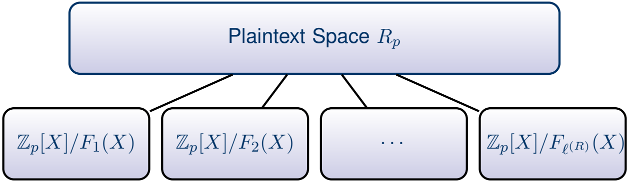
  S is a subring of R $(m'|m)$

- The scheme is parametrized by a sequence of decreasing moduli $q_L > q_{L-1} > \cdots > q_0 = q$, such that $Q = q_L = \prod_{i=0}^{L} p_i$.

  Fresh ciphertexts are defined in R_Q.

University of BRISTOL

Batch computation [SV11]

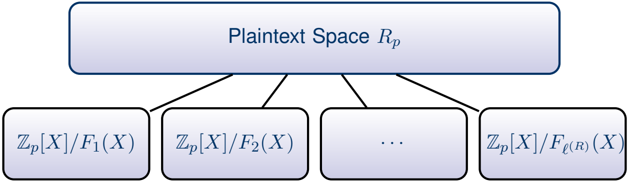


- Let p be a prime, coprime with m, and $R_p = R/pR = \mathbb{Z}_p[X]/\Phi_m(X)$
- We have $\ell^{(R)}$ isomorphisms

$$\psi_i : \mathbb{Z}_p[X]/F_i(X) \rightarrow \mathbb{F}_{p^{d^{(R)}}}, \quad i = 1, \ldots, \ell^{(R)},$$

$\Rightarrow$ we can represent $\ell^{(R)}$ plaintext elements of $\mathbb{F}_{p^{d^{(R)}}}$ as a single element in R_p.

Batch computation [SV11]



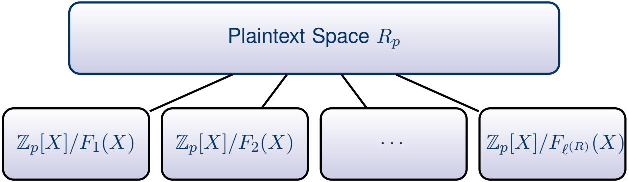
- Let p be a prime, coprime with m, and $R_p = R/pR = \mathbb{Z}_p[X]/\Phi_m(X)$
- We have $\ell^{(R)}$ isomorphisms

$$\psi_i : \mathbb{Z}_p[X]/F_i(X) \to \mathbb{F}_{p^{d^{(R)}}}, \quad i = 1, \ldots, \ell^{(R)},$$

$\Rightarrow$ we can represent $\ell^{(R)}$ plaintext elements of $\mathbb{F}_{p^{d^{(R)}}}$ as a single element in R_p.

- S_p splits into $\ell^{(S)}$ slots

Batch computation [SV11]



- Let p be a prime, coprime with m, and $R_p = R/pR = \mathbb{Z}_p[X]/\Phi_m(X)$
- We have $\ell^{(R)}$ isomorphisms

$$\psi_i : \mathbb{Z}_p[X]/F_i(X) \to \mathbb{F}_{p^{d^{(R)}}}, \quad i = 1, \ldots, \ell^{(R)},$$

$\Rightarrow$ we can represent $\ell^{(R)}$ plaintext elements of $\mathbb{F}_{p^{d^{(R)}}}$ as a single element in R_p.

- S_p splits into $\ell^{(S)}$ slots
- By the CRT, addition and multiplication correspond to SIMD operations on the slots $\Rightarrow$ we can process $\ell^{(R)}$ input values at once.

Switching modulus and ring

- ⋆ We encrypt at level L and perform homomorphic operations down to level zero with a single ring switching to improve efficiency

$$c_L = \mathsf{Enc}_{\mathsf{sk}}(\mu) \in R_Q = R_{q_L}$$

$$\downarrow$$

$$\vdots$$

$$\downarrow$$

$$c_{L_2+1} = \mathsf{Enc}_{\mathsf{sk}}(\mu) \in R_{q_{L_2+1}}$$

$$\downarrow$$

ring switching $$c_{L_2} = \mathsf{Enc}_{\mathsf{sk}'}(\mu') \in S_{L_2}$$

$$\downarrow$$

$$\vdots$$

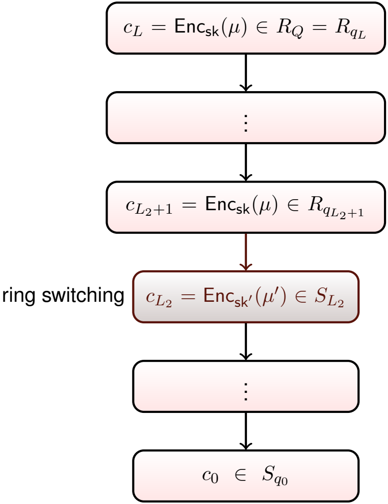
$$\downarrow$$

$$c_0 \in S_{q_0}$$

University of BRISTOL

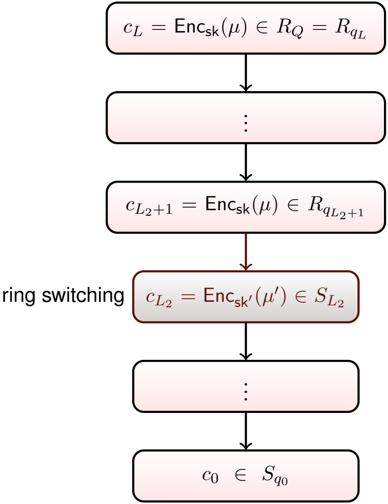
Switching modulus and ring

- We encrypt at level L and perform homomorphic operations down to level zero with a single ring switching to improve efficiency
- The ciphertexts before ring switching are associated to $\ell^{(R)}$ plaintext slots



Switching modulus and ring

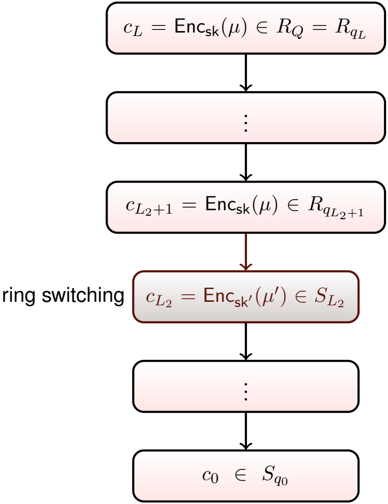
- ⋆ We encrypt at level L and perform homomorphic operations down to level zero with a single ring switching to improve efficiency
- ⋆ The ciphertexts before ring switching are associated to $\ell^{(R)}$ plaintext slots
- ⋆ With the ring switching the input ciphertext becomes associated with $\ell^{(R)}/\ell^{(S)}$ distinct ciphertexts associated to $\ell^{(S)}$ plaintext slots



University of BRISTOL

Switching modulus and ring

- ⋆ We encrypt at level L and perform homomorphic operations down to level zero with a single ring switching to improve efficiency
- ⋆ The ciphertexts before ring switching are associated to $\ell^{(R)}$ plaintext slots
- ⋆ With the ring switching the input ciphertext becomes associated with $\ell^{(R)}/\ell^{(S)}$ distinct ciphertexts associated to $\ell^{(S)}$ plaintext slots
- ◇ Bootstrap a number $(\ell^{(R)}/n)$ of ciphertexts in S_q in one shot.



Bootstrapping in more details

- The ciphertexts are elements $c = (c_0, c_1) \in R_q^2$

Bootstrapping in more details

- The ciphertexts are elements $c = (c_0, c_1) \in R_q^2$
- The decryption is an evaluation of a linear function D (dependent on c)

$$D_C(x) = \left((c_0 + x \cdot c_1) \mod q\right)$$

on the secret key sk mod q, followed by a reduction $\mod p$.

Bootstrapping in more details

- The ciphertexts are elements $c = (c_0, c_1) \in R_q^2$
- The decryption is an evaluation of a linear function D (dependent on c)
$$D_C(\mathsf{sk}) = \big((c_0 + \mathsf{sk} \cdot c_1) \mod q\big)$$
on the secret key sk mod q, followed by a reduction $\mod p$.

Bootstrapping in more details

- The ciphertexts are elements $c = (c_0, c_1) \in R_q^2$
- The decryption is an evaluation of a linear function D (dependent on c)

$$D_C(\mathsf{sk}) = \left((c_0 + \mathsf{sk} \cdot c_1) \mod q\right)$$

on the secret key sk mod q, followed by a reduction $\mod p$.

Bootstrapping: homomorphic evaluation decryption circuit:

Bootstrapping in more details

- The ciphertexts are elements $c = (c_0, c_1) \in R_q^2$
- The decryption is an evaluation of a linear function D (dependent on c)

$$D_C(\mathsf{sk}) = \left((c_0 + \mathsf{sk} \cdot c_1) \mod q\right)$$

on the secret key sk mod q, followed by a reduction $\mod p$.

Bootstrapping: homomorphic evaluation decryption circuit:

⋆ Given an encryption of the secret key sk, we can homomorphically evaluate D

Bootstrapping in more details

- The ciphertexts are elements $c = (c_0, c_1) \in R_q^2$
- The decryption is an evaluation of a linear function D (dependent on c)

$$D_C(\mathsf{sk}) = \left((c_0 + \mathsf{sk} \cdot c_1) \mod q\right)$$

on the secret key sk mod q, followed by a reduction $\mod p$.

Bootstrapping: homomorphic evaluation decryption circuit:

- ⋆ Given an encryption of the secret key sk, we can homomorphically evaluate D
- ⋆ Homomorphic evaluation of the $\mod p$ map

Bootstrapping in more details

Three main problems:

1. Homomorphically evaluate the $\bmod p$-map

$$
\begin{tikzcd}
\mathbb{Z}_q^+ \arrow[r, "\text{rep}"] \arrow[dr, "\bmod p"'] & \mathbb{G} \arrow[d, "\text{red}"] \\
& \mathbb{Z}_p
\end{tikzcd}
$$

University of BRISTOL

Bootstrapping in more details

Three main problems:

1. Homomorphically evaluate the $\bmod p$-map

$$\begin{tikzcd} \mathbb{Z}_q^+ \arrow[r, "\text{rep}"] \arrow[dr, red, "\bmod p"'] & \mathbb{G} \arrow[d, "\text{red}"] \\ & \mathbb{Z}_p \end{tikzcd}$$

2. Encode the sk and then using, a **dec-eval** function, create a set of ciphertexts encrypting the required input to red.

3. Packed ciphertexts

Overview of our technique

1. Find a suitable representation of S_q as an algebraic group over $\mathbb{F}_p$
2. SIMD evaluation of **dec-eval** over $\mathbb{G}$
3. SIMD evaluation of red
4. Repacking ciphertexts

Overview of our technique

1. Find a suitable representation of S_q as an algebraic group over $\mathbb{F}_p$
2. SIMD evaluation of **dec-eval** over $\mathbb{G}$
3. SIMD evaluation of red
4. Repacking ciphertexts

We give two different instantiations:

- Polynomial representation
- Elliptic curve based version

Step 1: Polynomial representation (1)

Find an $\mathbb{F}_p$-representation $\mathbb{G}$ for $\mathbb{Z}_q^+$:

$$\begin{tikzcd}
\mathbb{Z}_q^+ \arrow[r, "\text{rep}"] \arrow[dr, "\bmod p"'] & \mathbb{G} \arrow[d] \\
& \mathbb{Z}_p
\end{tikzcd}$$

Step 1: Polynomial representation (1)

Find an $\mathbb{F}_p$-representation $\mathbb{G}$ for $\mathbb{Z}_q^+$:

- By the CRT we have a group embedding

$$\text{rep} : \begin{cases} \mathbb{Z}_q^+ & \longrightarrow \ \mathbb{G} = \prod_{i=1}^t \mathbb{F}^*_{p^{k_i}} \\ a & \longmapsto \ (g_1^{a_1}, \ldots, g_t^{a_t}) \end{cases}$$

  for some k_i, where $a_i = a \pmod{e_i}$, $q = \prod_{i=1}^t e_i$

$$\begin{tikzcd} \mathbb{Z}_q^+ \arrow[r, "\text{rep}"] \arrow[dr, "\bmod p"'] & \mathbb{G} \arrow[d] \\ & \mathbb{Z}_p \end{tikzcd}$$

- One **add** in $\mathbb{Z}_q^+$ translates into $M = \frac{1}{2}\sum_{i=1}^t k_i \cdot (k_i + 1)$ **mult** in $\mathbb{F}_p$;
  each element in $\mathbb{G}$ requires $E = \sum_{i=1}^t k_i$ elements in $\mathbb{F}_p$ to represent it.

Step 1: Polynomial representation (1)

Find an $\mathbb{F}_p$-representation $\mathbb{G}$ for $\mathbb{Z}_q^+$:

- By the CRT we have a group embedding

$$\text{rep}: \begin{cases} \mathbb{Z}_q^+ & \longrightarrow \mathbb{G} = \prod_{i=1}^{t} \mathbb{F}^*_{p^{k_i}} \\ a & \longmapsto (g_1^{a_1}, \ldots, g_t^{a_t}) \end{cases}$$

  for some k_i, where $a_i = a \pmod{e_i}$, $q = \prod_{i=1}^{t} e_i$

$$\begin{tikzcd} \mathbb{Z}_q^+ \arrow[r, "\text{rep}"] \arrow[dr, "\bmod p"'] & \mathbb{G} \arrow[d, "\text{red}"] \\ & \mathbb{Z}_p \end{tikzcd}$$

- One **add** in $\mathbb{Z}_q^+$ translates into $M = \frac{1}{2}\sum_{i=1}^{t} k_i \cdot (k_i + 1)$ **mult** in $\mathbb{F}_p$; each element in $\mathbb{G}$ requires $E = \sum_{i=1}^{t} k_i$ elements in $\mathbb{F}_p$ to represent it.

A reduction $\mathbb{G} \to \mathbb{Z}_p$ can be defined by *algebraically* from the coefficient representation of $\mathbb{G}$ to $\mathbb{F}_p$. (Step 3)

Step 1: Polynomial representation - Extending maps

- Let $\tau = \mathsf{red} \circ \mathsf{rep}$

$$\begin{tikzcd} \mathbb{Z}_q^+ \arrow[r, "\mathsf{rep}"] \arrow[dr, "\mathrm{mod}\ p"'] & \mathbb{G} \arrow[d, "\mathsf{red}"] \\ & \mathbb{Z}_p \end{tikzcd}$$

Step 1: Polynomial representation - Extending maps

- Let $\tau = \text{red} \circ \text{rep}$
- By linearity we extend the maps:

$$\widehat{\text{rep}} : (S_q^+) \longrightarrow \mathbb{G}^n$$

and

$$\hat{\tau} : (S_q^+) \longrightarrow \mathbb{F}_p^n$$

$$\begin{tikzcd} \mathbb{Z}_q^+ \arrow[r, "\text{rep}"] \arrow[dr, "\bmod p"'] & \mathbb{G} \arrow[d, "\text{red}"] \\ & \mathbb{Z}_p \end{tikzcd}$$

Step 1: Polynomial representation - Extending maps

- Let $\tau = \mathsf{red} \circ \mathsf{rep}$
- By linearity we extend the maps:

$$\hat{\mathsf{rep}} : (S_q^+) \longrightarrow \mathbb{G}^n$$

and

$$\hat{\tau} : (S_q^+) \longrightarrow \mathbb{F}_p^n$$

$$\begin{tikzcd}
\mathbb{Z}_q^+ \arrow[r, "\mathsf{rep}"] \arrow[dr, "\mathrm{mod}\ p"'] & \mathbb{G} \arrow[d, "\mathsf{red}"] \\
 & \mathbb{Z}_p
\end{tikzcd}$$

- We want to bootstrap $\ell(R)/n$ ciphertexts, hence we define

$$\overline{\mathsf{rep}} : (S_q^+)^{\ell(R)/n} \longrightarrow \mathbb{G}^{\ell(R)}$$

and

$$\overline{\tau} : (S_q^+)^{\ell(R)/n} \longrightarrow \mathbb{F}_p^{\ell(R)}$$

$\overline{\mathsf{red}}$ is the SIMD evaluation of red on the image of $\overline{\mathsf{rep}}$ in $\mathbb{G}^{\ell(R)}$

Step 2 - Evaluating the decryption equation

We can then rewrite the decryption equation of our $\ell^{(R)}/n$ ciphertexts:

Step 2 - Evaluating the decryption equation

We can then rewrite the decryption equation of our $\ell^{(R)}/n$ ciphertexts:

$$\left(\left(c_0^{(j)} + \mathsf{sk}^{(S)} \cdot c_1^{(j)} \pmod{q}\right) (\mod p)\right)_{j=1}^{\ell^{(R)}/n}$$

Step 2 - Evaluating the decryption equation

We can then rewrite the decryption equation of our $\ell^{(R)}/n$ ciphertexts:

$$\begin{aligned}
&\left(\left(c_0^{(j)} + \mathsf{sk}^{(S)} \cdot c_1^{(j)} \pmod q\right) (\mod p)\right)_{j=1}^{\ell^{(R)}/n} \\
= \quad & \overline{\mathsf{red}}\left(\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)\right)
\end{aligned}$$

Step 2 - Evaluating the decryption equation

We can then rewrite the decryption equation of our $\ell^{(R)}/n$ ciphertexts:

$$
\begin{aligned}
&\left(\left(c_0^{(j)} + \mathsf{sk}^{(S)} \cdot c_1^{(j)} \pmod q\right) (\mod p)\right)_{j=1}^{\ell^{(R)}/n} \\
= \quad & \overline{\mathsf{red}}\left(\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)\right) \\
= \quad & \overline{\mathsf{red}}\left(\overline{\mathsf{rep}}\left(\mathbf{x}\right)\right),
\end{aligned}
$$

where $\mathbf{x}$ is the vector consisting of S_q elements

$$c_0^{(j)} + \mathsf{sk}^{(S)} \cdot c_1^{(j)},\ j = 1, \ldots, \ell^{(R)}/n$$

Step 2 - Evaluating the decryption equation

We can then rewrite the decryption equation of our $\ell^{(R)}/n$ ciphertexts:

$$\begin{aligned}
&\left(\left(c_0^{(j)} + \mathsf{sk}^{(S)} \cdot c_1^{(j)} \pmod q\right) (\mod p)\right)_{j=1}^{\ell^{(R)}/n} \\
= \quad & \overline{\mathsf{red}}\left(\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)\right) \\
= \quad & \overline{\mathsf{red}}\left(\overline{\mathsf{rep}}\left(\mathbf{x}\right)\right),
\end{aligned}$$

where $\mathbf{x}$ is the vector consisting of S_q elements

$$c_0^{(j)} + \mathsf{sk}^{(S)} \cdot c_1^{(j)},\ j = 1, \ldots, \ell^{(R)}/n$$

⋆ At this point $\overline{\mathsf{red}}$ is just an algebraic function

Step 2 - Evaluating the decryption equation

We can then rewrite the decryption equation of our $\ell^{(R)}/n$ ciphertexts:

$$\begin{aligned}
&\left(\left(c_0^{(j)} + \mathsf{sk}^{(S)} \cdot c_1^{(j)} \pmod q\right) (\mod p)\right)_{j=1}^{\ell^{(R)}/n} \\
= \quad &\overline{\mathsf{red}}\left(\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)\right) \\
= \quad &\overline{\mathsf{red}}\left(\overline{\mathsf{rep}}\left(\mathbf{x}\right)\right),
\end{aligned}$$

where $\mathbf{x}$ is the vector consisting of S_q elements

$$c_0^{(j)} + \mathsf{sk}^{(S)} \cdot c_1^{(j)},\ j = 1, \ldots, \ell^{(R)}/n$$

- ⋆ At this point $\overline{\mathsf{red}}$ is just an algebraic function
- ⋆ We need to homomorphically evaluate $\overline{\mathsf{rep}}(\mathbf{x})$

University of BRISTOL

$$\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)$$

MATRIX REPRESENTATION

- We can associate an element $b \in S_q$ to an $n \times n$ matrix $\mathbf{M}_b$ over $\mathbb{Z}_q$ such that the vector

$$\mathbf{c} = \mathbf{M}_b \cdot \mathbf{a}$$

is the coefficient vector of c where $c = a \cdot b$.

Step 2: Homomorphic evaluation of $\overline{\mathsf{rep}}(\mathbf{x})$

$$\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)$$

MATRIX REPRESENTATION

- ◇ We can associate an element $b \in S_q$ to an $n \times n$ matrix $\mathbf{M}_b$ over $\mathbb{Z}_q$ such that the vector

$$\mathbf{c} = \mathbf{M}_b \cdot \mathbf{a}$$

is the coefficient vector of c where $c = a \cdot b$.

- $c_1^{(j)} \to \mathbf{M}_{c_1^{(j)}}, j = 1, \ldots, \ell^{(R)}/n$

Step 2: Homomorphic evaluation of $\overline{\mathsf{rep}}(\mathbf{x})$

$$\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)$$

- $\mathbf{M}_{c_1^{(j)}} = \sum_{k=0}^{\lceil \log q / \log p \rceil} p^k \cdot \mathbf{M}_1^{(j,k)}, \quad j = 1, \ldots, \ell^{(R)}/n$

University of BRISTOL

Step 2: Homomorphic evaluation of $\overline{\mathsf{rep}}(\mathbf{x})$

$$\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)$$

- $\mathbf{M}_{c_1^{(j)}} = \sum_{k=0}^{\lceil \log q / \log p \rceil} p^k \cdot \mathbf{M}_1^{(j,k)}, \quad j = 1, \ldots, \ell^{(R)}/n$
- Setting $\sum_j \mathbf{M}_1^{(j,k)} = \mathbf{M}_1^{(k)}$

$$\overline{\mathsf{rep}}\left(c_0^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)}\right) \cdot \prod_{k=0}^{\lceil \log q / \log p \rceil} \overline{\mathsf{rep}}\left(p^k \cdot \underline{\mathsf{sk}}^{(S)}, \ldots, p^k \cdot \underline{\mathsf{sk}}^{(S)}\right)^{\mathbf{M}_1^{(k)}}$$

University of BRISTOL

Step 2: Homomorphic evaluation of $\overline{\mathsf{rep}}(\mathbf{x})$

$$\overline{\mathsf{rep}}\left(c_0^{(1)} + \mathsf{sk}^{(S)} \cdot c_1^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)} + \mathsf{sk}^{(S)} \cdot c_1^{(\ell^{(R)}/n)}\right)$$

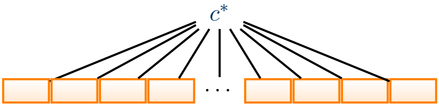
- $\mathbf{M}_{c_1^{(j)}} = \sum_{k=0}^{\lceil \log q / \log p \rceil} p^k \cdot \mathbf{M}_1^{(j,k)}, \quad j = 1, \ldots, \ell^{(R)}/n$
- Setting $\sum_j \mathbf{M}_1^{(j,k)} = \mathbf{M}_1^{(k)}$

$$\overline{\mathsf{rep}}\left(c_0^{(1)}, \ldots, c_0^{(\ell^{(R)}/n)}\right) \cdot \prod_{k=0}^{\lceil \log q / \log p \rceil} \overline{\mathsf{rep}}\left(p^k \cdot \underline{\mathsf{sk}}^{(S)}, \ldots, p^k \cdot \underline{\mathsf{sk}}^{(S)}\right)^{\mathbf{M}_1^{(k)}}$$

◇ $\mathsf{Enc}\left(\overline{\mathsf{rep}}(p^k \cdot \underline{\mathsf{sk}}^{(S)}, \ldots, p^k \cdot \underline{\mathsf{sk}}^{(S)})\right)$, for $k = 0, \ldots, \lceil \log q / \log p \rceil$

University of BRISTOL

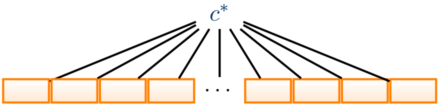
Step 4: Repacking



$\ell^{(R)}$ slots encoding the coefficients of the ciphertexts we are bootstrapping

- ⋆ We need to extract these coefficients to produce a ciphertext (or a set of ciphertexts) which encode the same data.

Step 4: Repacking



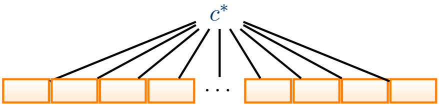
$\ell^{(R)}$ slots encoding the coefficients of the ciphertexts we are bootstrapping

⋆ We need to extract these coefficients to produce a ciphertext (or a set of ciphertexts) which encode the same data.

Different ways to perform this task:

- Technique from [AP13]
- Otherwise the Full Replication algorithm from [HS14].

Step 4: Repacking



$\ell^{(R)}$ slots encoding the coefficients of the ciphertexts we are bootstrapping

⋆ We need to extract these coefficients to produce a ciphertext (or a set of ciphertexts) which encode the same data.

Different ways to perform this task:

- Technique from [AP13]
- Otherwise the Full Replication algorithm from [HS14].

Note that we could produce

◇ $\ell^{(R)}/n$ ciphertexts each of which encodes one of the original plaintexts

◇ a single ciphertext which encodes all of them.

Conclusion

- ⋆ A new bootstrapping technique based on the $\mathbb{F}_p$-representation of the additive group $(\mathbb{Z}_q, +)$
  - ◇ Polynomial instantiation
  - ◇ Elliptic curve based version

Conclusion

- A new bootstrapping technique based on the $\mathbb{F}_p$-representation of the additive group $(\mathbb{Z}_q, +)$
  - Polynomial instantiation
  - Elliptic curve based version
- Using a polynomial representation:

Conclusion

- ⋆ A new bootstrapping technique based on the $\mathbb{F}_p$-representation of the additive group $(\mathbb{Z}_q, +)$
  - ◇ Polynomial instantiation
  - ◇ Elliptic curve based version
- ⋆ Using a polynomial representation:
  - ◇ Step 2: Homomorphically evaluate $\overline{\mathsf{rep}}(\mathbf{x})$
    $\rightarrow$ multiplicative depth $O(\log_2(\ell^{(R)}))$

Conclusion

- $\star$ A new bootstrapping technique based on the $\mathbb{F}_p$-representation of the additive group $(\mathbb{Z}_q, +)$
  - $\diamond$ Polynomial instantiation
  - $\diamond$ Elliptic curve based version
- $\star$ Using a polynomial representation:
  - $\diamond$ Step 2: Homomorphically evaluate $\overline{\mathsf{rep}}(\mathbf{x})$
    $\rightarrow$ multiplicative depth $O(\log_2(\ell^{(R)}))$
  - $\diamond$ Step 3: SIMD evaluation of $\overline{\mathsf{red}}$
    $\rightarrow O(\log_2 p + \log_2 \log_2 q)$

Conclusion

- A new bootstrapping technique based on the $\mathbb{F}_p$-representation of the additive group $(\mathbb{Z}_q, +)$
  - Polynomial instantiation
  - Elliptic curve based version

- Using a polynomial representation:
  - Step 2: Homomorphically evaluate $\overline{\mathsf{rep}}(\mathbf{x})$
    $\rightarrow$ multiplicative depth $O(\log_2(\ell^{(R)}))$
  - Step 3: SIMD evaluation of $\overline{\mathsf{red}}$
    $\rightarrow O(\log_2 p + \log_2 \log_2 q)$
  - Step 4: Repacking step
    $\rightarrow O(\log_2 \log_2 \ell^{(R)})$

University of BRISTOL

Conclusion

- ⋆ A new bootstrapping technique based on the $\mathbb{F}_p$-representation of the additive group $(\mathbb{Z}_q, +)$
  - ◇ Polynomial instantiation
  - ◇ Elliptic curve based version

- ⋆ Using a polynomial representation:
  - ◇ Step 2: Homomorphically evaluate $\overline{\mathsf{rep}}(\mathbf{x})$
    $\rightarrow$ multiplicative depth $O(\log_2(\ell^{(R)}))$
  - ◇ Step 3: SIMD evaluation of $\overline{\mathsf{red}}$
    $\rightarrow O(\log_2 p + \log_2 \log_2 q)$
  - ◇ Step 4: Repacking step
    $\rightarrow O(\log_2 \log_2 \ell^{(R)})$

University of BRISTOL

Conclusion

- ⋆ A new bootstrapping technique based on the $\mathbb{F}_p$-representation of the additive group $(\mathbb{Z}_q, +)$
  - ◇ Polynomial instantiation
  - ◇ Elliptic curve based version

- ⋆ Using a polynomial representation:
  - ◇ Step 2: Homomorphically evaluate $\overline{\mathsf{rep}}(\mathbf{x})$
    $\rightarrow$ multiplicative depth $O(\log_2(\ell^{(R)}))$
  - ◇ Step 3: SIMD evaluation of $\overline{\mathsf{red}}$
    $\rightarrow O(\log_2 p + \log_2 \log_2 q)$
  - ◇ Step 4: Repacking step
    $\rightarrow O(\log_2 \log_2 \ell^{(R)})$

THANK YOU!