(Batch) Fully Homomorphic Encryption over Integers for Non-Binary Message Spaces

*Koji Nuida[1,2], Kaoru Kurosawa[3]

[1] National Institute of Advanced Industrial Science and Technology (AIST), Japan
[2] Japan Science and Technology Agency (JST) PRESTO Researcher
[3] Ibaraki University, Japan

EUROCRYPT 2015 (Sofia, Bulgaria) April 29, 2015

- Extending "FHE over integers" [van Dijk et al. EC'10] from plaintext space $\mathcal{M} = \{0, 1\}$ to **$\mathcal{M} = \mathbb{Z}_Q := \mathbb{Z}/Q\mathbb{Z}$ for any prime Q**
  - And "batch" version [Cheon et al. EC'13] from $\{0, 1\}^k$ to $\prod_i \mathbb{Z}_{Q_i}$ (omitted here)
- Reducing multiplicative degree of decryption **from $O(\lambda(\log \lambda)^2)$ to $O(\lambda)$**
- Concrete (not just asymptotic) bootstrappable condition for parameters

Contents

- Introduction
- Q-ary Half-Adder by Polynomial
- Low-Degree Q-ary Integer Addition
- Our Scheme

Contents

Fully Homomorphic Encryption (FHE)

- (Public key) encryption that enables anyone to evaluate any function on the plaintexts
- Example: For plaintext space $\mathbb{Z}_2$, a scheme with ciphertext operators $\oplus, \otimes$ satisfying

$$\mathrm{Enc}(m_1) \oplus \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 + m_2)$$
$$\mathrm{Enc}(m_1) \otimes \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 \cdot m_2)$$

  - Every function over $\mathbb{Z}_2$ can be written as a combination of $+$ and $\times$

- Ciphertext for $m \in \mathbb{Z}_2$: $c = pq + 2r + m$ (p: secret prime, r: random noise)
- $\mathrm{Dec}(c) = (c \bmod p) \bmod 2$ (if noise is small)
- Homomorphic $+$, $\times$: Just applying them to ciphertexts

Rough Sketch of FHE by [van Dijk et al. EC'10]

- Ciphertext for $m \in \mathbb{Z}_2$: $c = pq + 2r + m$ (p: secret prime, r: random noise)
- $\mathrm{Dec}(c) = (c \bmod p) \bmod 2$ (if noise is small)
- Homomorphic $+, \times$: Just applying them to ciphertexts
  - Noise grows, to be cancelled by bootstrapping (cf., [Gentry STOC'09])
- "Squashing" Dec circuit to reduce the degree

- $\mathrm{Dec}^*(c) = c + \lfloor \sum_{i=1}^{\Theta} s_i z_i \rceil \bmod 2$
  - $(s_1, \ldots, s_\Theta) \in \{0,1\}^\Theta$: **secret** vector
  - $z_i = (z_{i,0} \,.\, z_{i,1} \ldots z_{i,L})_2$: binary real numbers satisfying $\sum_{i=1}^{\Theta} s_i z_i \approx c/p$

Squashed Decryption for Bootstrapping

- $\mathrm{Dec}^*(c) = c + \lfloor \sum_{i=1}^{\Theta} s_i z_i \rceil \bmod 2$
  - $(s_1, \ldots, s_\Theta) \in \{0,1\}^\Theta$: **secret** vector
  - $z_i = (z_{i,0} \,.\, z_{i,1} \ldots z_{i,L})_2$: binary real numbers satisfying $\sum_{i=1}^{\Theta} s_i z_i \approx c/p$
- To sum up $s_i z_i = ((s_i z_{i,0}) \,.\, (s_i z_{i,1}) \ldots (s_i z_{i,L}))_2$, each digit $s_i z_{i,j} \in \mathbb{Z}_2$ is given in **encrypted** form

- $\mathrm{Dec}^*(c) = c + \lfloor \sum_{i=1}^{\Theta} s_i z_i \rceil \bmod 2$
  - $(s_1, \ldots, s_\Theta) \in \{0,1\}^\Theta$: **secret** vector
  - $z_i = (z_{i,0} \,.\, z_{i,1} \ldots z_{i,L})_2$: binary real numbers satisfying $\sum_{i=1}^{\Theta} s_i z_i \approx c/p$
- To sum up $s_i z_i = ((s_i z_{i,0}) \,.\, (s_i z_{i,1}) \ldots (s_i z_{i,L}))_2$, each digit $s_i z_{i,j} \in \mathbb{Z}_2$ is given in **encrypted** form
- Carry function for binary **integer** addition should be given as **polynomial** on **finite field** $\mathbb{Z}_2$ to apply homomorphic operations

A Key Mathematical Fact

- For binary digits $x_1, \ldots, x_n$ represented by elements of $\mathbb{Z}_2$, their integer sum $(\ldots y_2 y_1 y_0)_2$ is given by

$$y_i = e_{2^i}(x_1, \ldots, x_n) \bmod 2$$

(cf., [Boyar et al. 2000])

  - where e_{2^i} is the elementary symmetric polynomial of degree 2^i (over $\mathbb{Z}_2$)

- Based on this, squashed decryption circuit is homomorphically evaluated

- For **Q-ary** digits x_1, x_2 represented by elements of $\mathbb{Z}_Q$, their integer sum $(y_1 y_0)_Q$ is given by

$$y_i = \varphi_i(x_1, x_2) \bmod Q$$

  - We constructed such a concrete polynomial φ_i of degree Q^i (over $\mathbb{Z}_Q$)
    - Note: $\varphi_0(x_1, x_2) = x_1 + x_2$ (in $\mathbb{Z}_Q$)
- Based on this, SHE over integers with $\mathcal{M} = \mathbb{Z}_Q$ [Cheon et al. EC'13] is made bootstrappable

Contents

Q-ary Half Adder

- For $x, y \in \mathbb{Z}_Q$, let $c, s \in \mathbb{Z}_Q$ satisfy

$$x + y = (c, s)_Q = c \cdot Q + s \text{ (as integers)}$$

- Building block of our bootstrapping algorithm

Q-ary Half Adder

- For $x, y \in \mathbb{Z}_Q$, let $c, s \in \mathbb{Z}_Q$ satisfy

$$x + y = (c, s)_Q = c \cdot Q + s \text{ (as integers)}$$

- Building block of our bootstrapping algorithm
- Note: $s = x + y$ (in $\mathbb{Z}_Q$)
- Problem: Find polynomial $c = f(x, y)$ (over $\mathbb{Z}_Q$)

Theorem We have

$$c = f(x,y) = \sum_{i=1}^{Q-1} \binom{x}{i}_Q \binom{y}{Q-i}_Q$$

where (for $a \in \mathbb{Z}$ and $b \in \mathbb{Z}_Q$)

$$\binom{a}{b}_Q := a(a-1)\cdots(a-b+1)\cdot\ (\ (b!)^{-1} \text{ in } \mathbb{Z}_Q\ )$$

Theorem We have

$$c = f(x,y) = \sum_{i=1}^{Q-1} \binom{x}{i}_Q \binom{y}{Q-i}_Q$$

where (for $a \in \mathbb{Z}$ and $b \in \mathbb{Z}_Q$)

$$\binom{a}{b}_Q := a(a-1)\cdots(a-b+1)\cdot\ (\ (b!)^{-1} \text{ in } \mathbb{Z}_Q\ )$$

- $\deg(f) = Q$ (**optimal**, and such f is unique)
- When $Q = 2$, $f(x,y) = xy = e_{2^1}(x,y)$ (known)

(Proof) (cf., Lucas' Theorem (1878))

$$c \equiv_{\bmod Q} \binom{x+y \text{ in } \mathbb{Z}}{Q}$$

$$= \binom{x}{0}\binom{y}{Q} + \binom{x}{1}\binom{y}{Q-1} + \cdots + \binom{x}{Q}\binom{y}{0}$$

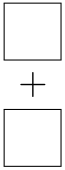
$$= \binom{x}{1}\binom{y}{Q-1} + \cdots + \binom{x}{Q-1}\binom{y}{1}$$

$$\equiv_{\bmod Q} \binom{x}{1}_Q \binom{y}{Q-1}_Q + \cdots + \binom{x}{Q-1}_Q \binom{y}{1}_Q$$

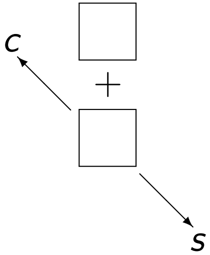
Contents

- Introduction
- Q-ary Half-Adder by Polynomial
- Low-Degree Q-ary Integer Addition
- Our Scheme

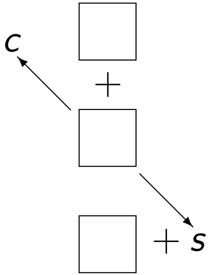
Q-ary Addition: Single-Digit



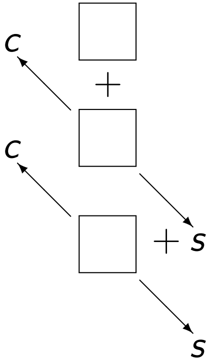
Q-ary Addition: Single-Digit

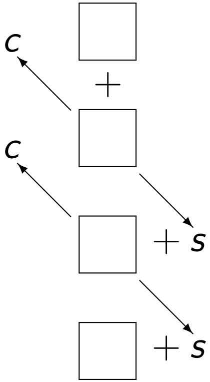


Q-ary Addition: Single-Digit

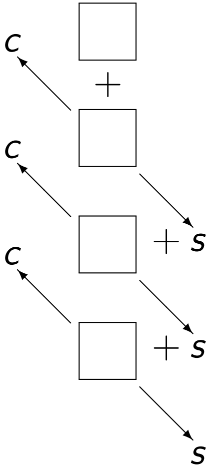


Q-ary Addition: Single-Digit

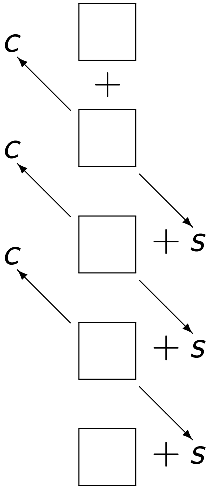



c
$+$
c
$+$ s
$+$ s

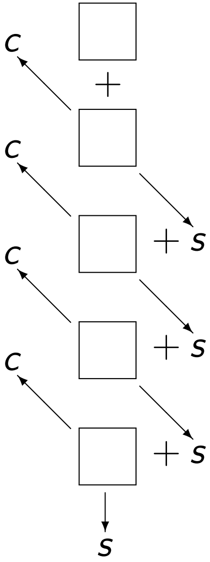
Q-ary Addition: Single-Digit



Q-ary Addition: Single-Digit

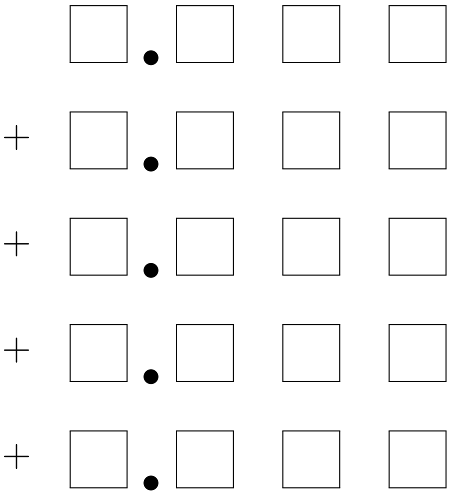


Q-ary Addition: Single-Digit



Q-ary Addition: Multi-Digits

$(\bmod Q)$



Q-ary Addition: Multi-Digits

$(\bmod Q)$

$$\boxed{c}\,\bullet\,\boxed{c}\;\;\boxed{c}$$

$$+\;\boxed{c}\,\bullet\,\boxed{c}\;\;\boxed{c}$$

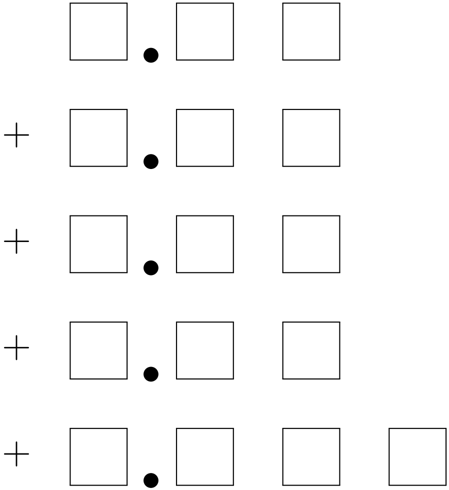
$$+\;\boxed{c}\,\bullet\,\boxed{c}\;\;\boxed{c}$$

$$+\;\boxed{c}\,\bullet\,\boxed{c}\;\;\boxed{c}$$

$$+\;\boxed{s}\,\bullet\,\boxed{s}\;\;\boxed{s}\;\;\boxed{s}$$

Q-ary Addition: Multi-Digits

$\pmod{Q}$



Q-ary Addition: Multi-Digits

$$\pmod{Q} \qquad \boxed{c} \bullet \boxed{c}$$

$$+ \quad \boxed{c} \bullet \boxed{c}$$

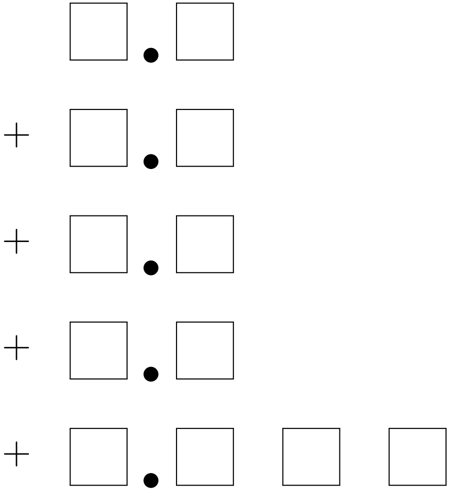
$$+ \quad \boxed{c} \bullet \boxed{c}$$

$$+ \quad \boxed{c} \bullet \boxed{c}$$

$$+ \quad \boxed{s} \bullet \boxed{s} \quad \boxed{s} \quad \boxed{}$$

Q-ary Addition: Multi-Digits

$(\bmod Q)$



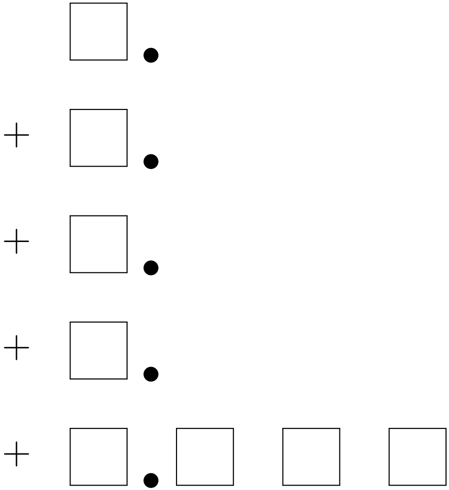
Q-ary Addition: Multi-Digits

$(\bmod Q)$

$$
\begin{array}{rllll}
 & \boxed{c}\,\bullet & & & \\
+ & \boxed{c}\,\bullet & & & \\
+ & \boxed{c}\,\bullet & & & \\
+ & \boxed{c}\,\bullet & & & \\
+ & \boxed{s}\,\bullet & \boxed{s} & \boxed{} & \boxed{}
\end{array}
$$

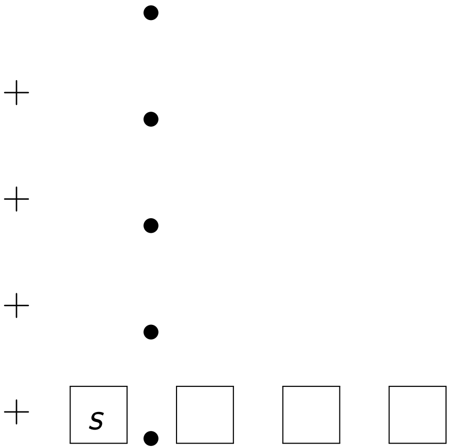
Q-ary Addition: Multi-Digits

$(\bmod Q)$



Q-ary Addition: Multi-Digits

$(\bmod Q)$



Q-ary Addition: Multi-Digits

$\pmod{Q}$

$$
\begin{array}{ccccc}
 & & \bullet & & \\
+ & & \bullet & & \\
+ & & \bullet & & \\
+ & & \bullet & & \\
+ & \boxed{d_0} & \bullet\ \boxed{d_1} & \boxed{\cdots} & \boxed{d_L}
\end{array}
$$

d_i: polynomial of degree Q^{L-i}
in the original digits

Contents

SHE Scheme

(Essentially the same as [Cheon et al. EC'13])

- Plaintext space $\mathcal{M} = \mathbb{Z}_Q$
- Ciphertexts are in modulo $N = pq_0$ (p **secret**)

(Essentially the same as [Cheon et al. EC'13])

- Plaintext space $\mathcal{M} = \mathbb{Z}_Q$
- Ciphertexts are in modulo $N = pq_0$ (p **secret**)
- Public key: N, $x' = \mathrm{Enc}(1)$, many $x_\xi = \mathrm{Enc}(0)$
- $\mathrm{Enc}^*(m) = m \cdot x' + \sum_{\xi \in T} x_\xi$ (for random T)
- $\mathrm{Dec}^*(c) = (c \bmod p) \bmod Q$
- Homomorphic $+, \times$ are usual $+, \times$ for integers

- **Secret** vector: $\vec{s} = (s_1, \ldots, s_\Theta) \in \{0,1\}^\Theta$, $\mathrm{weight}(\vec{s}) = \theta$. $v_\ell = \mathrm{Enc}(s_\ell)$ are made **public**
- Random **public** integers $0 \leq u_\ell < Q^{\kappa+1}$ with

$$\sum_{\ell=1}^{\Theta} s_\ell u_\ell \equiv_{\bmod Q^{\kappa+1}} \lfloor Q^\kappa \cdot (p \bmod Q)/p \rceil$$

- $\mathrm{Dec}^*(c)$: Computes $z_\ell = (z_{\ell,0} \,.\, z_{\ell,1} \ldots z_{\ell,L})_Q$ with $z_\ell \approx c u_\ell / Q^\kappa \bmod Q$, and outputs

$$m := c - \lfloor \sum_{\ell=1}^{\Theta} s_\ell z_\ell \rceil$$

Bootstrapping

- Recall: $v_\ell = \mathrm{Enc}(s_\ell)$, $z_\ell = (z_{\ell,0} \,.\, z_{\ell,1} \ldots z_{\ell,L})_Q$,

$$\mathrm{Dec}^*(c) = c - \lfloor \textstyle\sum_{\ell=1}^{\Theta} s_\ell z_\ell \rceil$$

Bootstrapping

- Recall: $v_\ell = \mathrm{Enc}(s_\ell)$, $z_\ell = (z_{\ell,0} . z_{\ell,1} \dots z_{\ell,L})_Q$,
$$\mathrm{Dec}^*(c) = c - \lfloor \sum_{\ell=1}^{\Theta} s_\ell z_\ell \rceil$$
- Computes $v^*_{\ell,\xi} := z_{\ell,\xi} \cdot v_\ell$ for $\xi = 1, \dots, L$
  - Intuition: $v^*_\ell = (v^*_{\ell,0} . v^*_{\ell,1} \dots v^*_{\ell,L})_Q$ is digit-wise encryption of $s_\ell z_\ell$

Bootstrapping

- Recall: $v_\ell = \mathrm{Enc}(s_\ell)$, $z_\ell = (z_{\ell,0} \,.\, z_{\ell,1} \ldots z_{\ell,L})_Q$,
$$\mathrm{Dec}^*(c) = c - \lfloor \sum_{\ell=1}^{\Theta} s_\ell z_\ell \rceil$$
- Computes $v^*_{\ell,\xi} := z_{\ell,\xi} \cdot v_\ell$ for $\xi = 1, \ldots, L$
  - Intuition: $v^*_\ell = (v^*_{\ell,0} \,.\, v^*_{\ell,1} \ldots v^*_{\ell,L})_Q$ is digit-wise encryption of $s_\ell z_\ell$
- Homomorphically computes $w = (w_0 \,.\, w_1 \ldots w_L)_Q = \sum_{\ell=1}^{\Theta} v^*_\ell \bmod Q$
  - $\mathrm{Dec}(w_1) \in \{0, Q-1\}$, so $\lfloor \mathrm{Dec}(w) \rceil = \mathrm{Dec}(w_0) - \mathrm{Dec}(w_1) \bmod Q$

- Recall: $v_\ell = \mathrm{Enc}(s_\ell)$, $z_\ell = (z_{\ell,0} \,.\, z_{\ell,1} \ldots z_{\ell,L})_Q$,
$$\mathrm{Dec}^*(c) = c - \lfloor \textstyle\sum_{\ell=1}^{\Theta} s_\ell z_\ell \rceil$$
- Computes $v^*_{\ell,\xi} := z_{\ell,\xi} \cdot v_\ell$ for $\xi = 1, \ldots, L$
  - Intuition: $v^*_\ell = (v^*_{\ell,0} \,.\, v^*_{\ell,1} \ldots v^*_{\ell,L})_Q$ is digit-wise encryption of $s_\ell z_\ell$
- Homomorphically computes $w = (w_0 \,.\, w_1 \ldots w_L)_Q = \sum_{\ell=1}^{\Theta} v^*_\ell \bmod Q$
  - $\mathrm{Dec}(w_1) \in \{0, Q-1\}$, so $\lfloor \mathrm{Dec}(w) \rceil = \mathrm{Dec}(w_0) - \mathrm{Dec}(w_1) \bmod Q$
- Outputs $c^* = (c \bmod Q) - (w_0 - w_1) \bmod N$

Multiplicative Degree of Bootstrapping

- In $(w_0 \,.\, w_1 \dots w_L)_Q = \sum_{\ell=1}^{\Theta} v_\ell^* \bmod Q$, w_i is a polynomial in $v_{1,0}^*, \dots, v_{\Theta,L}^*$ of degree $Q^{L-i} \leq Q^L$

- In $(w_0 \,.\, w_1 \ldots w_L)_Q = \sum_{\ell=1}^{\Theta} v_\ell^* \bmod Q$, w_i is a polynomial in $v_{1,0}^*, \ldots, v_{\Theta,L}^*$ of degree $Q^{L-i} \leq Q^L$
- Our parameter choice (following the previous work) yields $L = \lceil \log_Q \theta \rceil + 2$ and $\theta = \lambda$.
- $\deg(\mathrm{Dec}^*) \leq Q^{\log_Q \lambda + 3} = Q^3 \cdot \lambda = O(\lambda)$

- In $(w_0 . w_1 \dots w_L)_Q = \sum_{\ell=1}^{\Theta} v_\ell^* \bmod Q$, w_i is a polynomial in $v_{1,0}^*, \dots, v_{\Theta,L}^*$ of degree $Q^{L-i} \le Q^L$
- Our parameter choice (following the previous work) yields $L = \lceil \log_Q \theta \rceil + 2$ and $\theta = \lambda$.
- $\deg(\mathrm{Dec}^*) \le Q^{\log_Q \lambda + 3} = Q^3 \cdot \lambda = O(\lambda)$
- Note: In "(digit-wise sum) + (three-for-two trick)" method in the previous work, the former part already uses polynomials of degree $O(\lambda)$
  - The latter part increases the degree further

Parameter Choice

- Our choice of bootstrappable parameters yields:
  - Public key size: $\Theta(\lambda^8(\log \lambda)^6)$ bits
  - Secret key size: $\Theta(\lambda^4(\log \lambda)^4)$ bits
  - Ciphertext size: $\Theta(\lambda^4(\log \lambda)^2)$ bits
  - ($\Theta(\lambda \log \log \log \lambda)$-bit noise in Enc)
- An explicit condition for bootstrappable parameters is given in the proceedings

Summary

- We extended FHE over integers and its batch version from binary plaintexts to Q-ary ones
  - By explicitly constructing polynomials for carry functions in Q-ary addition
- Multiplicative degree of decryption is reduced from $O(\lambda(\log\lambda)^2)$ to $O(\lambda)$
- We also gave concrete (not just asymptotic) bootstrappable condition for parameters

Summary

- We extended FHE over integers and its batch version from binary plaintexts to Q-ary ones
  - By explicitly constructing polynomials for carry functions in Q-ary addition
- Multiplicative degree of decryption is reduced from $O(\lambda(\log \lambda)^2)$ to $O(\lambda)$
- We also gave concrete (not just asymptotic) bootstrappable condition for parameters

Thank you for your attention!