FHE over the Integers and Modular Arithmetic Circuits

Eunkyung Kim[1] Mehdi Tibouchi[2]

[1]Ewha Womans University, South Korea

[2]NTT Secure Platform Laboratories, Japan

WHEAT 2016, 2016–07–06

©2016 NTT Secure Platform Laboratories

Outline

©2016 NTT Secure Platform Laboratories

Outline

©2016 NTT Secure Platform Laboratories

FHE and binary message spaces

- Most FHE schemes introduced with message space $\mathbb{Z}/2\mathbb{Z}$
- Support the homomorphic evaluation of Boolean circuits
- In particular, in FHE "over the integers" [vDGHV10,...], ciphertexts usually look like:

$$c = pq + 2r + m, \qquad m \in \{0, 1\}$$

- Some variants with multiple slots (message space $(\mathbb{Z}/2\mathbb{Z})^m$) or extension fields ($\mathrm{GF}(2^m)$), but still binary

©2016 NTT Secure Platform Laboratories

How about non-binary message spaces?

- Could we replace 2 by some other value Q? (odd prime, say)
- We would then evaluate mod-Q arithmetic circuits instead of Boolean ones
- The most naive way works somewhat
- E.g. for FHE over the integers, use ciphertexts of the form:

$$c = pq + Qr + m, \qquad m \in \{0, \ldots, Q-1\}$$

- Addition and multiplication work fine mod Q: can evaluate low-degree polynomials mod Q on ciphertexts
- Can you get fully homomorphic encryption that way?

©2016 NTT Secure Platform Laboratories

The bootstrapping problem

- To get FHE from somewhat homomorphic encryption, we use bootstrapping: homomorphic evaluation of the decryption circuit
- Decryption (for ciphertexts above) looks like:

$$m = (c \bmod p) \bmod Q$$

- This has to be expressed as a low-depth mod-Q arithmetic circuit (squashing). Main hurdle: division c mod p
- In binary: write $1/p \approx \sum s_i y_i$ (y_i fixed precision public reals, all but one pseudorandom; s_i random secret bits). Division then becomes a large iterated addition:

$$\sum s_i(cy_i)$$

©2016 NTT Secure Platform Laboratories

The Nuida–Kurosawa approach

- Squashing mod Q: need to write a low-depth mod-Q arithmetic circuit for precise enough iterated addition
- Looked like a daunting task, so nobody touched it for many years, until Nuida–Kurosawa (EUROCRYPT 2015)
- They gave explicit mod-Q circuits for iterated addition; deduced an FHE scheme over the integers with message space $\mathbb{Z}/Q\mathbb{Z}$
- Only works for small Q (otherwise, squashed decryption circuit depth too large for bootstrappability)

©2016 NTT Secure Platform Laboratories

Outline

©2016 NTT Secure Platform Laboratories

Boolean circuits vs. arithmetic circuits

- Mod-Q arithmetic circuits can be efficiently simulated by Boolean circuits (size expansion factor polylogarithmic in Q) [vzGS91]
- In particular, easy to homomorphically evaluate mod-Q arithmetic circuits using FHE with binary message space:
  1. encrypt $m \in \mathbb{Z}/Q\mathbb{Z}$ bit by bit, as $\log_2 Q$ ciphertexts c_i
  2. convert the mod-Q arithmetic circuit to Boolean, by replacing $+$ and $\times$ gates by Boolean subcircuits doing those operations
- Therefore, FHE with non-binary message space at most an optimization

©2016 NTT Secure Platform Laboratories

Is the optimization worth it?

- So we asked ourselves the following question: is the mod-Q scheme in [NK15] (NK_Q) a good optimization compared to using Boolean circuits?
- For large Q, impossible:
  - overhead of NK_Q (in terms of ciphertext size & cost of bootstrapping) is $\mathrm{poly}(Q)$
  - converting a mod-Q circuit to Boolean, the overhead is only $\mathrm{polylog}(Q)$
- It could be worth it for small Q, though. Let's compare.
- For a level playing field, we compared NK_Q to its own binary version: $\mathrm{Convert\text{-}NK}_2$

©2016 NTT Secure Platform Laboratories

Outline

©2016 NTT Secure Platform Laboratories

A new FHE $\mathrm{Convert\text{-}NK}_2$ with $\mathcal{M} = \mathbb{Z}/Q\mathbb{Z}$

Let NK_2 be NK FHE with $\mathcal{M} = \mathbb{Z}/2\mathbb{Z}$, then $\mathrm{Convert\text{-}NK}_2$ scheme is described as follow:

- KeyGen(1^λ): $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{NK}_2.\mathsf{KeyGen}(1^\lambda)$
- Enc(pk, m): for $m \in \mathcal{M} = \mathbb{Z}/Q\mathbb{Z}$, write $m = (m_{n-1}, \cdots, m_0)$ $(n = \lceil \log(Q+1) \rceil)$ and encrypt m bitwise

$$\vec{c} = (c_{n-1}, \cdots, c_0) \text{ with } c_i \leftarrow \mathrm{NK}_2.\mathsf{Enc}(\mathsf{pk}, m_i)$$

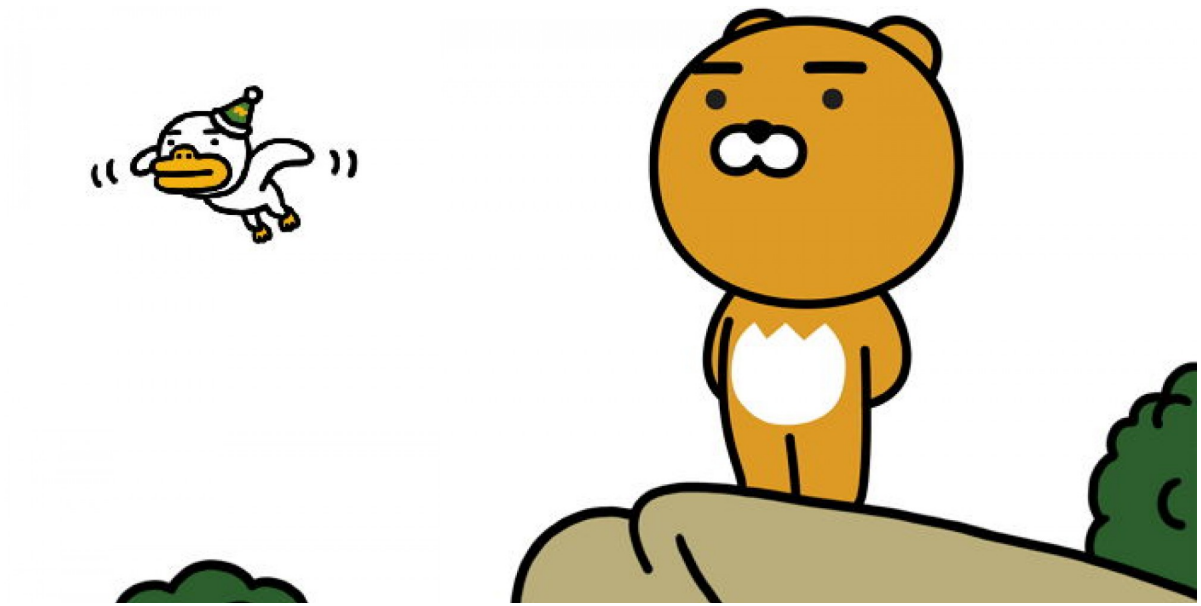
- Dec$(\mathsf{sk}, \vec{c})$: $m_i \leftarrow \mathrm{NK}_2.\mathsf{Dec}(\mathsf{sk}, c_i)$ and return

$$m = \sum_{i=0}^{n-1} m_i 2^i$$

- Eval: Use Boolean circuits which compute mod-Q addition and mod-Q multiplication

©2016 NTT Secure Platform Laboratories

In this work



We compared $\mathrm{Convert\text{-}NK}_2$ vs NK_Q; which is better?

©2016 NTT Secure Platform Laboratories

Convert-NK_2 vs NK_Q: Criteria for Comparison

1. Ciphertext size
   - γ_Q: $N_Q \in [1, 2^{\gamma_Q}) \cap \mathbb{Z}$
   - $\gamma_2' \approx \gamma_2 \log Q$: ciphertext of Convert-$\mathrm{NK}_2$ is n-tuple of ciphertexts of NK_2
2. Time complexity to execute one mod-Q multiplication
   - T_Q: time complexity of a single ciphertext refresh operation in NK_Q,
   - T_2': time complexity of carrying out a multiplication mod Q in Convert-NK_2

©2016 NTT Secure Platform Laboratories

Outline

©2016 NTT Secure Platform Laboratories

- $\rho = \omega(\lambda)$, to resist the attack by Chen and Nguyen [4] for the approximate GCD assumption.
- $\gamma > \eta^2/\rho$, to resist Howgrave-Graham's attack [11] for the approximate GCD assumption.
- $\eta = \Omega(\lambda^2)$ and $\gamma = (\sum_{i=1}^{k} h_i) \cdot \eta + \Omega(\lambda^2)$, to resist Lenstra's elliptic curve method [13] for factoring the integer N (the latter is to make the (approximate) bit length $\gamma - (\sum_{i=1}^{k} h_i) \cdot \eta$ of q_0 sufficiently large).
- $\gamma = \eta^2 \omega(\log \lambda)$, to resist the attack by Cohn and Heninger [5] and the attack using Lagarias algorithm [12] on the approximate GCD assumption. (This implies the condition $\gamma = \Omega(\lambda^3)$ arisen from the general number field sieve [2] for factoring N.)
- $\tau = \gamma + \omega(\log \lambda)$, in order to use the Leftover Hash Lemma in the security proof (see [3] for the details).

Figure : Conditions on parameters from [NK15]

©2016 NTT Secure Platform Laboratories

Choice of parameters in NK FHE

Q is treated as constant

- $\rho = \Theta(\lambda \log\log\log \lambda)$: size of noise
- $\eta = \Theta(\lambda^2 \log\log \lambda)$: size of secret prime
- $\gamma = \Theta(\lambda^4 \log^2 \lambda)$: size of ciphertexts
- $L = \lceil \log_Q \lambda \rceil + 2$: the number of precision after Q-ary point in z_i
- $\boldsymbol{\Theta} = \Theta((\lambda \log \lambda)^4)$: the number of sparse elements s_i

In a nutshell, we want to compare the case $Q > 2$ with $Q = 2$, so it is important not to ignore Q as constant.

©2016 NTT Secure Platform Laboratories

Choice of parameters in NK FHE

Q is treated as constant

- $\rho = \Theta(\lambda \log\log\log \lambda)$: size of noise
- $\eta = \Theta(\lambda^2 \log\log \lambda)$: size of secret prime
- $\gamma = \Theta(\lambda^4 \log^2 \lambda)$: size of ciphertexts
- $L = \lceil \log_Q \lambda \rceil + 2$: the number of precision after Q-ary point in z_i
- $\boldsymbol{\Theta} = \Theta((\lambda \log \lambda)^4)$: the number of sparse elements s_i

In a nutshell, we want to compare the case $Q > 2$ with $Q = 2$, so it is important not to ignore Q as constant.

©2016 NTT Secure Platform Laboratories

Dependence of parameters on Q

- In $\mathrm{NK}_Q.\mathsf{KeyGen}$, we have $v_i = pq_i + Qr_i + s_i$ and

$$\log |v_i \bmod p| = \log |Qr_i + s_i| \leq \log Q + \rho = O(\rho)$$

- Squashed decryption circuit can be computed within in degree $Q^{L_Q+2} \approx Q^3\lambda$ ($L_Q \approx \log_Q \lambda$)

In order to make $\mathrm{NK}_Q.\mathsf{Eval}(\mathsf{pk}, \mathrm{NK}_Q.\mathsf{Dec}, v_i, c)$ works correctly,

$$\eta_Q = (\text{noise size}) \cdot \Theta(\text{degree of } \mathsf{Dec}) = \Theta(\rho Q^3 \lambda)$$

Thus, $\eta \propto Q^3$, and hence $\gamma \propto Q^6$ since $\gamma \propto \eta^2$

©2016 NTT Secure Platform Laboratories

Choice of parameters with consideration of Q

We have parameters depending on Q

- $\eta_Q = \Theta(Q^3\lambda^2 \log\log\lambda)$: size of secret prime
- $\gamma_Q = \Theta(Q^6\lambda^4 \log^2\lambda)$: size of ciphertexts
- $L_Q = \lceil \log_Q \lambda \rceil + 2$: the number of precision after Q-ary point in z_i

and not depending on Q

- $\rho = \Theta(\lambda \log\log\log\lambda)$: size of noise
- $\boldsymbol{\Theta} = \Theta((\lambda\log\lambda)^4)$: the number of sparse elements s_i

©2016 NTT Secure Platform Laboratories

Ciphertext size of $\mathrm{Convert\text{-}NK}_2$ is smaller than NK_Q

- γ_Q: ciphertext size of NK_Q
- γ_2': ciphertext size of $\mathrm{Convert\text{-}NK}_2$

Proposition

For a given security parameter λ and odd prime $Q > 2$, we have

$$\frac{\gamma_2'}{\gamma_Q} = \Theta\left(\frac{\log Q}{Q^6}\right)$$

©2016 NTT Secure Platform Laboratories

Sketch of proof

- Ciphertext space of NK_Q is $\mathbb{Z}/N_Q\mathbb{Z}$ and $N_Q \in [1, 2^{\gamma}_{Q}) \cap \mathbb{Z}$
- $\gamma_Q = \Theta(Q^6 \lambda^4 \log^2 \lambda)$
- Ciphertext space of $\mathrm{Convert\text{-}NK}_2$ is $(\mathbb{Z}/N_2\mathbb{Z})^{\log Q}$
- $\gamma'_2 = \log Q \cdot \Theta(2^6 \lambda^4 \log^2 \lambda) = \Theta(\log Q \lambda^4 \log^2 \lambda)$
- $\dfrac{\gamma'_2}{\gamma_Q} = \dfrac{\Theta(\log Q \lambda^4 \log^2 \lambda)}{\Theta(Q^6 \lambda^4 \log^2 \lambda)} = \Theta\left(\dfrac{\log Q}{Q^6}\right)$

©2016 NTT Secure Platform Laboratories

Outline

©2016 NTT Secure Platform Laboratories

Basic binary operation

- k *bit* + k *bit*: 2 AND for each carry, and total $2k$ AND
- k *bit* $\times$ l *bit* for $(k \leq l)$: $2l(k + l)$ AND using so-called "two-out-of-three" technique

©2016 NTT Secure Platform Laboratories

Boolean circuit for $\bmod Q$ multiplication

input $m, m' \in \mathbb{Z}/Q\mathbb{Z}$, Pre-computed $Q' = \lceil \frac{2^K}{Q} \rceil$ with $K > 2n$ and $n \approx \log Q$

output $m \cdot m' \bmod Q$

1. $m \cdot m'$ $(n\,bit \times n\,bit)$
2. $(mm') \cdot Q'$ $(2n\,bit \times (K-n)\,bit)$
3. $Q \cdot \lfloor \frac{mm'Q'}{2^k} \rfloor$ $(n\,bit \times n\,bit)$
4. $mm' - Q\lfloor \frac{mm'Q'}{2^k} \rfloor$ $(2n\,bit + 2n\,bit)$

©2016 NTT Secure Platform Laboratories

Boolean circuit for $\bmod Q$ multiplication

input $m, m' \in \mathbb{Z}/Q\mathbb{Z}$, Pre-computed $Q' = \lceil \frac{2^K}{Q} \rceil$ with $K > 2n$ and $n \approx \log Q$

output $m \cdot m' \bmod Q$

1. $m \cdot m'$ $(n\,bit \times n\,bit)$
2. $(mm') \cdot Q'$ $(2n\,bit \times (K-n)\,bit)$
3. $Q \cdot \lfloor \frac{mm'Q'}{2^k} \rfloor$ $(n\,bit \times n\,bit)$
4. $mm' - Q\lfloor \frac{mm'Q'}{2^k} \rfloor$ $(2n\,bit + 2n\,bit)$

Total Number of AND gates

$$2(2n(n+n)) + 2(K-n)(2n+K-n) + 2(2n) \approx \mathbf{14}\log^2 \mathbf{Q}$$

©2016 NTT Secure Platform Laboratories

- t_Q: time complexity of one mod-N_Q multiplication
- T_Q: ($\sharp$ of mults in $\mathrm{NK}_Q.\mathrm{Dec}$)$\times t_Q$
- T_2': ($\sharp$ of AND gate in mod-Q mult Boolean circuit)$\times T_2$

Proposition

For a given security parameter λ and odd prime $Q > 2$, we have

$$\frac{T_2'}{T_Q} = O\left(\frac{\log^4 Q}{Q^7}\right)$$

©2016 NTT Secure Platform Laboratories

Sketch of proof

- $t_Q = \log N_Q \log\log N_Q \approx \log N_Q = \gamma_Q = \Theta(Q^6 \lambda^4 \log^2 \lambda)$
- $$\begin{aligned} T_Q &= 4Q\Theta L_Q^2 \cdot t_Q \\ &= \frac{4Q\Theta \log^2 \lambda}{\log^2 Q} \Theta(Q^6 \lambda^4 \log^2 \lambda) \\ &= \Theta\left(\frac{Q^7 \Theta \lambda^4 \log^4 \lambda}{\log Q}\right) \end{aligned}$$
- $$\begin{aligned} T_2' &= 14 \log^2 Q \cdot 4 \cdot 2\Theta L_2^2 \cdot t_2 \\ &= 8\Theta \log^2 \lambda \Theta(2^6 \lambda^4 \log^2 \lambda) \\ &= \Theta(\log^2 Q \Theta \lambda^4 \log^4 \lambda) \end{aligned}$$

©2016 NTT Secure Platform Laboratories

Sketch of proof

- $t_Q = \Theta(Q^6 \lambda^4 \log^2 \lambda)$
- $T_Q = \Theta\left(\frac{Q^7 \boldsymbol{\Theta} \lambda^4 \log^4 \lambda}{\log Q}\right)$
- $T_2' = \Theta(\log^2 Q \boldsymbol{\Theta} \lambda^4 \log^4 \lambda)$

Therefore

$$\frac{T_2'}{T_Q} = \frac{\Theta(\log^2 Q \boldsymbol{\Theta} \lambda^4 \log^4 \lambda)}{\Theta\left(\frac{Q^7 \boldsymbol{\Theta} \lambda^4 \log^4 \lambda}{\log Q}\right)} = \Theta\left(\frac{\log^4 Q}{Q^7}\right)$$

©2016 NTT Secure Platform Laboratories

Thank you for your attention

Questions?

©2016 NTT Secure Platform Laboratories