

## Section 6

# Key Switching

# Remember Proxy Re-encryption?

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Primary key:  $(pk, sk)$
- Secondary key:  $(pk_1, sk_1)$
- Re-encryption key:  $rk = Enc(pk_1, sk[1..k])$
- Input ciphertext  $c = Enc(pk, m)$
- Decryption function  $f_c(sk) = Dec(sk, c)$

## Question

*What is the result of the following computation?*

$Eval(pk_1, f_c, rk)$

# Remember Proxy Re-encryption?

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Primary key:  $(pk, sk)$
- Secondary key:  $(pk1, sk1)$
- Re-encryption key:  $rk = Enc(pk1, sk[1..k])$
- Input ciphertext  $c = Enc(pk, m)$
- Decryption function  $f_c(sk) = Dec(sk, c)$

## Question

*What is the result of the following computation?*

$$Eval(pk1, f_c, rk)$$

## Question

*Can you implement proxy re-encryption using LWE?*

# LWE-based Proxy Re-encryption?

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

$$sk[1..n] \in \mathbb{Z}_q^n$$

$$sk'[1..n] \in \mathbb{Z}_q^n$$

$$Enc(sk, msg) = LWE(sk, msg * pow2row) = (A[], b[])$$

$$rk[i] = Enc(sk', sk[i])$$

$$Dec'(sk, (A, b))[j] = b[j] - \sum_i sk[i] * A[i, j] \\ \approx msg * pow2row$$

$$Dec(sk, (A, b)) = decode(Dec'(sk, (A, b)))$$

## Question

*Can you compute  $Dec'$  homomorphically?*

*Does it give you a proxy re-encryption scheme?*

# LWE-based Proxy Re-encryption

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

$$\text{Enc}(\text{sk}, \text{msg}) = \text{LWE}(\text{sk}, \text{msg} * \text{pow2row})$$

$$\text{rk}[i] = \text{Enc}(\text{sk}', \text{sk}[i])$$

$$\text{Dec}'(\text{sk}, (A, b)) = b[j] - \sum_i \text{sk}[i] * A[i, j]$$

Goal: homomorphically evaluate the function

$$f_{A,b}(\text{sk}) = \text{Dec}'(\text{sk}, (A, b))$$

$$\text{Eval}(f_{A,b}, \text{rk}) = ?$$

# LWE-based Proxy Re-encryption

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

$$\text{Enc}(\text{sk}, \text{msg}) = \text{LWE}(\text{sk}, \text{msg} * \text{pow2row})$$

$$\text{rk}[i] = \text{Enc}(\text{sk}', \text{sk}[i])$$

$$\text{Dec}'(\text{sk}, (A, b)) = b[j] - \sum_i \text{sk}[i] * A[i, j]$$

Goal: homomorphically evaluate the function

$$f_{A,b}(\text{sk}) = \text{Dec}'(\text{sk}, (A, b))$$

$$\text{Eval}(f_{A,b}, \text{rk}) = ?$$

Solution:  $\text{Eval}(f_{A,b}, \text{rk}) = \text{ct}$

$$\text{ct}[j] = \text{Const}(b[j]) - \sum_i \text{CMul}(\text{rk}[i], A[i, j])$$

# Key Switching

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Generalize proxy re-encryption:
  - $sk, sk'$  may have different dimensions and moduli
  - $Enc(sk, \cdot), Enc'(sk', \cdot)$  may use different plaintext moduli and message encodings
- Example
  - Message space  $msg: \mathbb{Z}_p$
  - Ciphertext modulus  $q$
  - $sk[1..n], sk'[1..n] \in \mathbb{Z}_q^n$
  - $Enc(sk, m) = LWE(sk, (q/p)*msg) \bmod q$
  - Evaluation key:  $rk[i] = Enc(sk', sk[i])$
- Do you see any problem?

# Key Switching

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

Source scheme:

$$\text{msg} : \mathbb{Z}_p$$

$$\text{sk}[1..n] \in \mathbb{Z}_q^n$$

$$\text{Enc}(\text{sk}, \text{msg}) = \text{LWE}(\text{sk}, \frac{q}{p}\text{msg}) = (a[], b) \bmod q$$

Target scheme:

$$\text{msg}' : \mathbb{Z}_q$$

$$\text{sk}'[1..n'] \in \mathbb{Z}_q^{n'}$$

$$\text{Enc}'(\text{sk}', \text{msg}') = \text{LWE}(\text{sk}', \text{msg}' * \text{pow2row})$$

Evaluation:

$$\text{ek}[i] = \text{Enc}'(\text{sk}', \text{sk}[i])$$

$$\text{KeySwitch}(\text{ek}, (a[], b)) =$$

$$\text{Const}(b) - \sum_i \text{CMul}(a[i], \text{ek}[i])$$

# Correctness

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

$$\text{msg} : \mathbb{Z}_p; \text{sk}[1..n] \in \mathbb{Z}_q^n$$

$$\text{msg}' : \mathbb{Z}_q; \text{sk}'[1..n'] \in \mathbb{Z}_q^{n'}$$

$$\text{Enc}(\text{sk}, \text{msg}) = \text{LWE}(\text{sk}, \frac{q}{p} \text{msg}) = (a[], b) \bmod q$$

$$\text{Enc}'(\text{sk}', \text{msg}') = \text{LWE}(\text{sk}', \text{msg}' * \text{pow2row})$$

$$\text{ek}[i] = \text{Enc}'(\text{sk}', \text{sk}[i])$$

$$\text{KeySwitch}(\text{ek}, (a[], b))$$

$$= \text{Const}(b) - \sum_i \text{CMul}(a[i], \text{ek}[i])$$

# Correctness

$$\text{msg} : \mathbb{Z}_p; \text{sk}[1..n] \in \mathbb{Z}_q^n$$
$$\text{msg}' : \mathbb{Z}_q; \text{sk}'[1..n'] \in \mathbb{Z}_q^{n'}$$

$$\text{Enc}(\text{sk}, \text{msg}) = \text{LWE}(\text{sk}, \frac{q}{p} \text{msg}) = (a[], b) \bmod q$$

$$\text{Enc}'(\text{sk}', \text{msg}') = \text{LWE}(\text{sk}', \text{msg}' * \text{pow2row})$$

$$\text{ek}[i] = \text{Enc}'(\text{sk}', \text{sk}[i])$$

$$\text{KeySwitch}(\text{ek}, (a[], b))$$

$$= \text{Const}(b) - \sum_i \text{CMul}(a[i], \text{ek}[i])$$

$$= \text{Const}(b) - \sum_i \text{CMul}(a[i], \text{Enc}'(\text{sk}', \text{sk}[i]))$$

# Correctness

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

$$\text{msg} : \mathbb{Z}_p; \text{sk}[1..n] \in \mathbb{Z}_q^n$$

$$\text{msg}' : \mathbb{Z}_q; \text{sk}'[1..n'] \in \mathbb{Z}_q^{n'}$$

$$\text{Enc}(\text{sk}, \text{msg}) = \text{LWE}(\text{sk}, \frac{q}{p} \text{msg}) = (a[], b) \bmod q$$

$$\text{Enc}'(\text{sk}', \text{msg}') = \text{LWE}(\text{sk}', \text{msg}' * \text{pow2row})$$

$$\text{ek}[i] = \text{Enc}'(\text{sk}', \text{sk}[i])$$

$$\text{KeySwitch}(\text{ek}, (a[], b))$$

$$= \text{Const}(b) - \sum_i \text{CMul}(a[i], \text{ek}[i])$$

$$= \text{Const}(b) - \sum_i \text{CMul}(a[i], \text{Enc}'(\text{sk}', \text{sk}[i]))$$

$$= \text{LWE}(\text{sk}', b - \sum_i a[i] * \text{sk}[i])$$

$$= \text{LWE}(\text{sk}', \frac{q}{p} \text{msg} + e)$$

$$= \text{Enc}(\text{sk}', \text{msg})$$

# Remarks

- Source and Target schemes may use different moduli

- $Enc'(sk', msg') = LWE(sk', \frac{q}{p} msg' * \text{pow2row})$

# Remarks

- Source and Target schemes may use different moduli
  - $\text{Enc}'(\text{sk}', \text{msg}') = \text{LWE}(\text{sk}', \frac{q}{p} \text{msg}' * \text{pow2row})$
- Input ciphertext may use compact (matrix) LWE
  - $\text{Enc}(\text{SK}, \text{msg}[]) = \text{LWE}(\text{SK}, \frac{q}{p} \text{msg}[])$
  - $\text{RK}' = \text{Enc}'(\text{SK}', \text{SK})$

# Remarks

- Source and Target schemes may use different moduli
  - $\text{Enc}'(\text{sk}', \text{msg}') = \text{LWE}(\text{sk}', \frac{q}{p} \text{msg}' * \text{pow2row})$
- Input ciphertext may use compact (matrix) LWE
  - $\text{Enc}(\text{SK}, \text{msg}[]) = \text{LWE}(\text{SK}, \frac{q}{p} \text{msg}[])$
  - $\text{RK}' = \text{Enc}'(\text{SK}', \text{SK})$
- Key Switching:
  - Input:  $\text{Enc}(\text{sk}, \text{msg} : \text{mod } p) : \text{mod } q$
  - Switching Key:  $\text{Enc}'(\text{sk}', \text{sk} : \text{mod } q) : \text{mod } q'$
  - Output:  $\text{Enc}(\text{sk}', \text{msg} : \text{mod } p) : \text{mod } q'$

# Remarks

- Source and Target schemes may use different moduli

- $Enc'(sk', msg') = LWE(sk', \frac{q}{p} msg' * pow2row)$

- Input ciphertext may use compact (matrix) LWE

- $Enc(SK, msg[]) = LWE(SK, \frac{q}{p} msg[])$

- $RK' = Enc'(SK', SK)$

- Key Switching:

- Input:  $Enc(sk, msg: \text{mod } p): \text{mod } q$

- Switching Key:  $Enc'(sk', sk: \text{mod } q): \text{mod } q'$

- Output:  $Enc(sk', msg: \text{mod } p): \text{mod } q'$

- Input/Output can use arbitrary encoding, e.g.,

- Input:  $Enc(sk, msg) = LWE(sk, msg * pow2row)$

- Output:  $Enc(sk', msg) = LWE(sk', msg * pow2row)$

# Sub-key Switching

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Application: reduce key size  $SK \rightarrow SK'$
- Always:  $SK, SK'$  must have the same number of rows
- Often  $SK$  is a “sub-matrix” of  $SK = [SK', SK'']$
- Switching Key

$$\begin{aligned} [RK', RK''] &= \text{Enc}'(SK', SK) \\ &= \text{Enc}'(SK', [SK' \mid SK'']) \\ &= [\text{Enc}'(SK', SK') \mid \text{Enc}'(SK', SK'')] \end{aligned}$$

- But  $RK'$  is publicly known! (remember circular security?)
- Can use a smaller switching key  $RK'' = \text{Enc}'(SK', SK'')$

## Question

*Does it work? What if  $SK'' = []$ ? Then,  $RK'' = []$  and  $SK = SK'$ !  
Is it trivial? Is it useful?*

# Modulus switching

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Subkey switching from  $SK$  to  $SK' = SK$  can still be useful to change the ciphertext modulus from  $q$  to  $q'$
- So far we used the simplifying assumption that  $p|q$
- Switching from  $q$  to  $q'$  requires a switching key with
  - plaintext modulus  $q$
  - ciphertext modulus  $q'$
  - but if  $q|q'$ , this only allows to increase the modulus
- (Sub-)Key Switching works also for  $p \nmid q$ 
  - but introduces a “small” rounding error
  - for subkey switching the rounding error is proportional to  $SK$
  - switching to a smaller modulus requires “small” key  $SK$

# Subkey and Modulus Switching

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Subkey switching
  - Input:  $ct = \text{Enc}([SK', SK''], m)$  and  $RK = \text{Enc}'(SK', SK'')$
  - $\text{SubkeySwitch}(RK, ct) = ct'$  such that  $\text{Dec}(SK', ct') = m$

# Subkey and Modulus Switching

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Subkey switching

- Input:  $ct = \text{Enc}([SK', SK''], m)$  and  $RK = \text{Enc}'(SK', SK'')$
- $\text{SubkeySwitch}(RK, ct) = ct'$  such that  $\text{Dec}(SK', ct') = m$

## Question

*Give explicit description of SubkeySwitch algorithm*

# Subkey and Modulus Switching

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Subkey switching

- Input:  $ct = \text{Enc}([SK', SK''], m)$  and  $RK = \text{Enc}'(SK', SK'')$
- $\text{SubkeySwitch}(RK, ct) = ct'$  such that  $\text{Dec}(SK', ct') = m$

## Question

*Give explicit description of  $\text{SubkeySwitch}$  algorithm*

- Modulus switching

- Assume  $SK$  has small entries
- Input:  $ct = \text{Enc}(SK, m) \bmod q$  and nothing else
- $\text{ModSwitch}(ct) = ct' \bmod q'$  such that  $\text{Dec}(SK, ct') = m$

# Subkey and Modulus Switching

CSE208:  
Advanced  
Cryptography

Daniele  
Micciancio

Introduction

Defining FHE

Bootstrapping

LWE

Linearity

Key Switching

Multiplication

FHE!!

Ring LWE

- Subkey switching

- Input:  $ct = \text{Enc}([SK', SK''], m)$  and  $RK = \text{Enc}'(SK', SK'')$
- $\text{SubkeySwitch}(RK, ct) = ct'$  such that  $\text{Dec}(SK', ct') = m$

## Question

*Give explicit description of SubkeySwitch algorithm*

- Modulus switching

- Assume  $SK$  has small entries
- Input:  $ct = \text{Enc}(SK, m) \bmod q$  and nothing else
- $\text{ModSwitch}(ct) = ct' \bmod q'$  such that  $\text{Dec}(SK, ct') = m$

## Question

*Give explicit description of ModSwitch algorithm*