Key Recovery Attacks against NTRU-based Somewhat Homomorphic Encryption Schemes

Massimo Chenal Qiang Tang

University of Luxembourg
APSIA Group
SnT - Interdisciplinary Centre for Security, Reliability and Trust

September 11, 2015

Outline of the Talk

Outline

1. Quick overview of FHE
2. Previous work
3. Idea of our key recovery attacks
4. Details of attacks
5. Conclusion and future directions

Basic Definitions

We only assume bit-by-bit public-key encryption

Public Key Homomorphic Encryption Scheme

$\mathcal{E} = (\text{KeyGen}_{\mathcal{E}}, \text{Encrypt}_{\mathcal{E}}, \text{Decrypt}_{\mathcal{E}}, \text{Evaluate}_{\mathcal{E}})$, all run in poly. time.

$$\text{KeyGen}(\lambda) = (\text{sk}, \text{pk}), \text{Encrypt}(\text{pk}, m) = c$$

$$\text{Decrypt}(\text{sk}, c) = m', \text{Evaluate}(\text{pk}, C, (c_1, \ldots, c_r)) = c_e$$

Correct Homomorphic Decryption

$\mathcal{E}$ is correct for a given t-input circuit C if, $\forall (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(\lambda)$, $\forall m_1, \ldots, m_t \in \{0, 1\}$, $\forall \overline{c} = (c_1, \ldots, c_t)$ with $c_i \leftarrow \text{Encrypt}_{\mathcal{E}}(\text{pk}, m_i)$

$$\text{Decrypt}(\text{sk}, \text{Evaluate}(\text{pk}, C, \overline{c})) = C(m_1, \ldots, m_t)$$

Homomorphic Encryption

$\mathcal{E}$ homomorphic for a class $\mathcal{C}$ of circuits: correct for all circuits $C \in \mathcal{C}$
$\mathcal{E}$ fully homomorphic encryption (FHE) scheme: correct for all boolean circuits
$\mathcal{E}$ somewhat homomorphic encryption (SHE) scheme: limited # of op.

Basic Definitions

Security Definitions

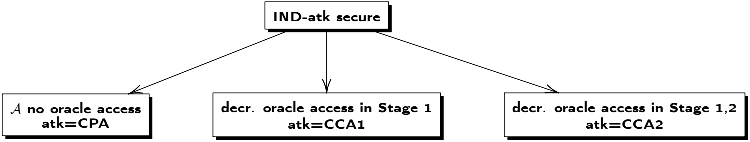
game between a challenger and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $(m_0, m_1) \leftarrow \mathcal{A}_1^{(\cdot)}(\mathsf{pk})$ /* Stage 1 */
- $b \leftarrow \{0, 1\}$
- $c^* \leftarrow \mathsf{Encrypt}(m_b, \mathsf{pk})$
- $b' \leftarrow \mathcal{A}_2^{(\cdot)}(c^*)$ /* Stage 2 */

If $b = b'$: $\mathcal{A}$ wins game with

$$\mathrm{Adv}_{\mathcal{A},\mathcal{E},\lambda}^{\text{IND-atk}} = |\Pr(b = b') - 1/2|$$

Scheme IND-atk secure if no poly. time $\mathcal{A}$ wins with non-negl. adv.



Quick overview of FHE based on hardness assumptions

- 1978: Rivest et al [RAD78]: is it possible to perform arbitrary operations on encrypted ciphertexts? (privacy homomorphism / FHE)
- 2009: Gentry [Gen09b]: yes!

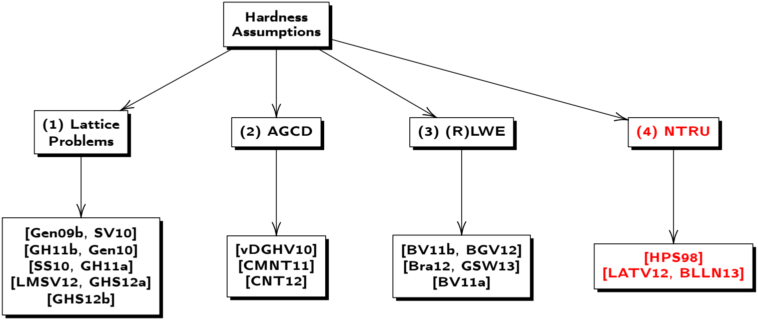


Figure : Hardness assumptions and relevant papers

Homomorphic Encryption and IND-CPA,CCA

- All known SHE and FHE schemes: IND-CPA secure
- No SHE and FHE scheme can be IND-CCA2
- With Gentry's approach, FHE scheme cannot be IND-CCA1 secure
- Open problem: investigate SHE schemes with IND-CCA1 security (Gentry [Gen09b])

Homomorphic Encryption and IND-CPA,CCA

- All known SHE and FHE schemes: IND-CPA secure
- No SHE and FHE scheme can be IND-CCA2
- With Gentry's approach, FHE scheme cannot be IND-CCA1 secure
- Open problem: investigate SHE schemes with IND-CCA1 security (Gentry [Gen09b])

IND-CPA ✔	IND-CCA1 ?	IND-CCA2 ✗
Every scheme	Not known	No scheme

Key Recovery Attacks - Our Contribution

Our contribution

1. key recovery attack for SHE schemes in [LATV12, BLLN13]
2. SHE schemes in (4) above are not IND-CCA1 secure
3. conclusion: with results from [LMSV12, ZPS12, CT14], most existing SHE schemes (except [LMSV12]) suffer from key recovery attacks, so not IND-CCA1 secure

Key Recovery Attacks - The Idea

General line of work

- Premise: decryption oracle reveals one bit at a time or a polynomial in $Z_2[x]/(x^n + 1)$
- Idea: we submit to decryption oracle specifically-chosen 'ciphertexts' in order to get 1 bit of information for each coefficient of sk
- recover sk by gradually reducing (halving) the key space

Key Recovery Attack against SHE [LATV12]

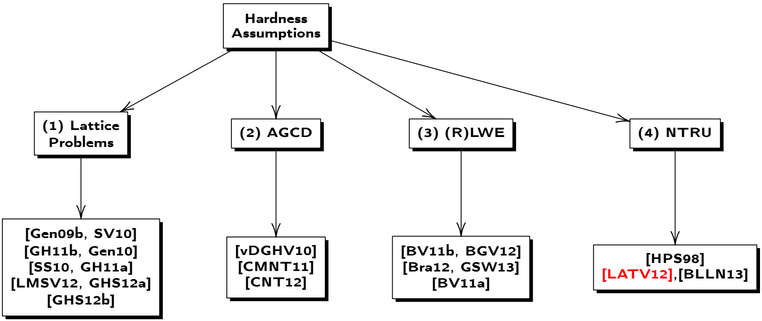


Figure : Hardness assumptions and relevant papers

Key Recovery Attack against [LATV12]

The [LATV12] SHE scheme (informal)

$\mathcal{M} = \mathbb{Z}_2$, $R := \mathbb{Z}[x]/(x^n + 1)$

KeyGen(λ) :

- $[\cdots]$
- sk $:= f \in R$

Encrypt(pk, m):

- sample $s, e \leftarrow \chi$
- output ciphertext $c := hs + 2e + m \in R_q$

Decrypt(sk, c):

- let $\mu = f \cdot c \in R_q$
- output $\mu' := \mu \bmod 2$

Comparison with [DGM15]

- attack exists in [DGM15], but require $6(t^2 + t) < q$ and $B^2 < \frac{q}{36t^2}$ (conditions not assumed in [LATV12])
- our attack: works for all parameters. More efficient than [DGM15]:

Our Attack	Attack from [DGM15]
$\lfloor \log_2 B \rfloor + n$	$n \cdot \lceil \log_2 B \rceil + n$

- n: power of 2; $B \ll q$ bound on coefficient of χ; $t \geq 2$ integer

Key Recovery Attack against [LATV12]

KeyGen(λ) :

- sk := $s(x) = s_0 + s_1 x + s_2 x^2 + \cdots + s_{n-1}x^{n-1} \in R_q$

Encrypt(pk, m):

- output ciphertext $c(x) \in R_q$

Decrypt(sk, $c(x)$):

- output $s(x) \cdot c(x) \in R_q \bmod 2$

Key recovery attack - The Idea

- determine the parity of each coefficient $s_i \in (-q/2, q/2]$
- determine $|s_i|$ by gradually halving the interval in which it lies
- at some point, $|s_i|$ belongs to some interval with at most two consecutive integers
- $|s_i|$ deduced by its known parity
- last step: query the oracle decryption at most n times in order to recover the sign of the coefficients s_i, for $i = 1, 2, \ldots, n-1$, relative to the (unknown) sign of s_0
- two possible candidate secret keys $s_1(x)$ and $s_2(x) = -s_1(x)$
- find whether $s(x) = s_1(x)$ or $s(x) = s_2(x)$ with extra oracle query

Key Recovery Attack against [LATV12] - Details

Preliminary Step

- submit to dec. oracle $c(x) = 1 \in R_q$
- oracle returns $D(c(x) = 1) = s(x) \bmod 2 = \sum_{i=0}^{n-1}(s_i \bmod 2)x^i$
- $\Rightarrow$ we learn parity of s_i, $i = 0, 1, \ldots, n-1$

Step 1

- submit to dec. oracle $c(x) = 2 \in R_q$
- oracle returns $D(c(x) = 2) = (2s(x) \in R_q) \bmod 2 = \sum_{i=0}^{n-1} [(2s_i \bmod q) \bmod 2]\, x^i$
- Now, $\forall i \in [0, n-1]$ we have

$$\frac{-q+1}{2} \le s_i \le \frac{q-1}{2}, \text{ and so } -q+1 \le 2s_i \le q-1 \qquad \text{(A)}$$

$\forall i$, two cases to distinguish:

Case A_1: $(2s_i \bmod q) \bmod 2 = 0$.
Then, condition (A) implies that
$\frac{-q+1}{2} \le 2s_i \le \frac{q-1}{2}$, i.e.
$\frac{-q+1}{4} \le s_i \le \frac{q-1}{4}$

$$-q+1 \le 4s_i \le q-1 \qquad \text{(A1)}$$

Case B_1: $(2s_i \bmod q) \bmod 2 = 1$.
Then, condition (A) implies that
$\frac{q-1}{2} + 1 \le 2|s_i| \le q-1$, i.e.
$\frac{q+1}{4} \le |s_i| \le \frac{q-1}{2}$

$$q+1 \le 4|s_i| \le 2q-2 \qquad \text{(B1)}$$

Key Recovery Attack against [LATV12] - Details

Step 2

- submit to dec. oracle $c(x) = 4 \in R_q$
- oracle returns $D(c(x) = 4) = [s(x) \cdot 4]_q \bmod 2 = \sum_{i=0}^{n-1} [[4s_i]_q \bmod 2]\, x^i$
- Now, $\forall i$, four cases to distinguish:

Case A_2**:** In Step 1 case A_1 held, and $[4s_i]_q \bmod 2 = 0$. Then, condition (A1) implies that $\frac{-q+1}{2} \leq 4s_i \leq \frac{q-1}{2}$, i.e. $\frac{-q+1}{8} \leq s_i \leq \frac{q-1}{8}$

$$-q+1 \leq 8s_i \leq q-1 \qquad \text{(A2)}$$

Case B_2**:** In Step 1 case A_1 held, and $[4s_i]_q \bmod 2 = 1$. Then, condition (A1) implies that $\frac{q-1}{2} + 1 \leq 4|s_i| \leq q-1$, i.e. $\frac{q+1}{8} \leq |s_i| \leq \frac{q-1}{4}$

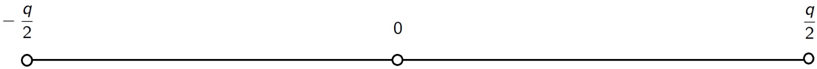
$$q+1 \leq 8|s_i| \leq 2q-2 \qquad \text{(B2)}$$

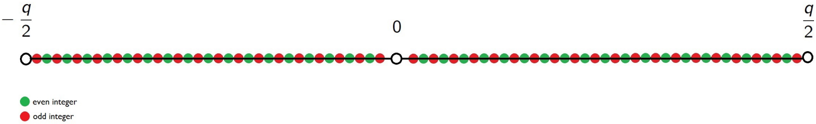
Case C_2**:** In Step 1 case B_1 held, and $[4s_i]_q \bmod 2 = 0$. Then, condition (B1) implies that $q+1+\frac{q-1}{2} \leq 4|s_i| \leq 2q-2$, i.e. $\frac{3q+1}{8} \leq |s_i| \leq \frac{q-1}{2}$

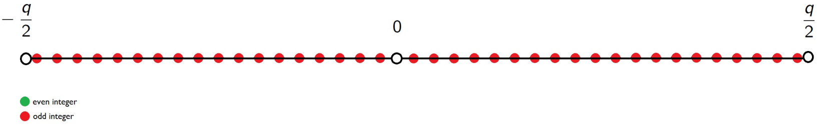
$$3q+1 \leq 8|s_i| \leq 4q-4 \qquad \text{(C2)}$$

Case D_2**:** In Step 1 case B_1 held, and $[4s_i]_q \bmod 2 = 1$. Then, condition (B1) implies that $q+1 \leq 4|s_i| \leq \frac{3q-1}{2}$, i.e. $\frac{q+1}{4} \leq |s_i| \leq \frac{3q-1}{8}$

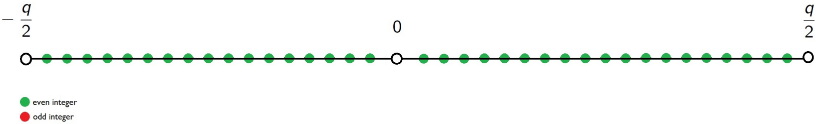
$$2q+2 \leq 8|s_i| \leq 3q-1 \qquad \text{(D2)}$$

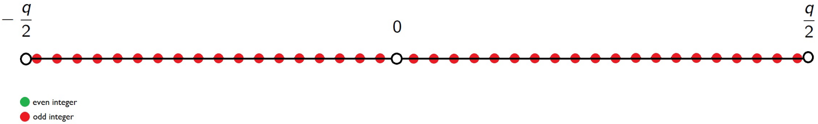

$-\frac{q}{2}$
0
$\frac{q}{2}$


$-\frac{q}{2}$
0
$\frac{q}{2}$
even integer
odd integer

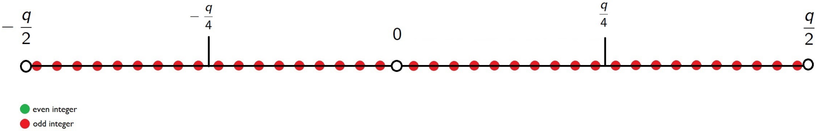


$-\frac{q}{2}$
0
$\frac{q}{2}$
even integer
odd integer

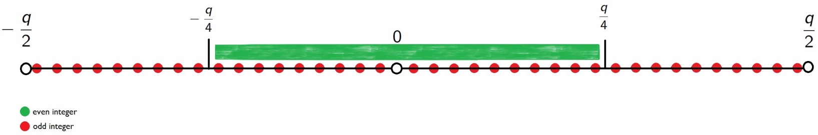

$-\frac{q}{2}$
0
$\frac{q}{2}$
even integer
odd integer



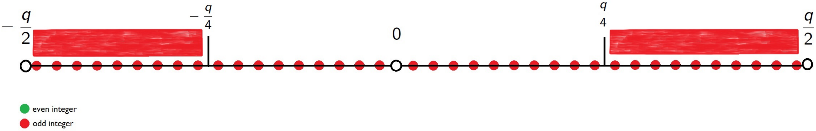
$-\frac{q}{2}$
0
$\frac{q}{2}$
even integer
odd integer



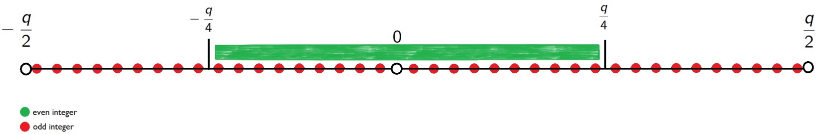
$-\frac{q}{2}$
$-\frac{q}{4}$
0
$\frac{q}{4}$
$\frac{q}{2}$
even integer
odd integer



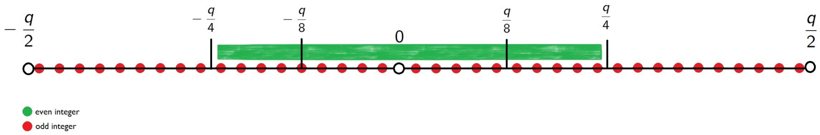
$-\frac{q}{2}$
$-\frac{q}{4}$
0
$\frac{q}{4}$
$\frac{q}{2}$
even integer
odd integer

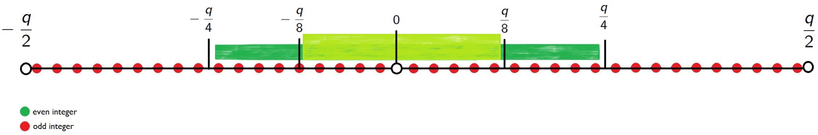


$-\frac{q}{2}$
$-\frac{q}{4}$
0
$\frac{q}{4}$
$\frac{q}{2}$
even integer
odd integer

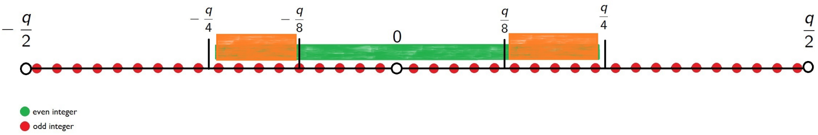


$-\frac{q}{2}$
$-\frac{q}{4}$
0
$\frac{q}{4}$
$\frac{q}{2}$
even integer
odd integer

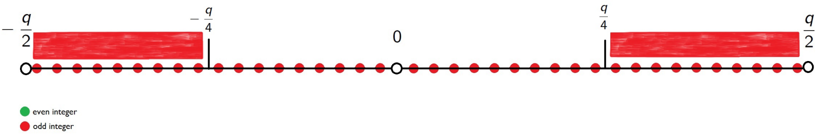

$-\frac{q}{2}$
$-\frac{q}{4}$
$-\frac{q}{8}$
0
$\frac{q}{8}$
$\frac{q}{4}$
$\frac{q}{2}$
even integer
odd integer



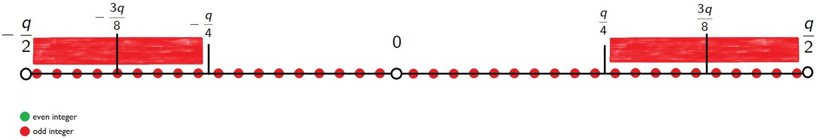
$-\frac{q}{2}$
$-\frac{q}{4}$
$-\frac{q}{8}$
0
$\frac{q}{8}$
$\frac{q}{4}$
$\frac{q}{2}$
even integer
odd integer

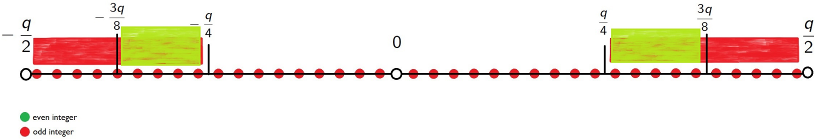


$-\frac{q}{2}$
$-\frac{q}{4}$
$-\frac{q}{8}$
0
$\frac{q}{8}$
$\frac{q}{4}$
$\frac{q}{2}$
even integer
odd integer

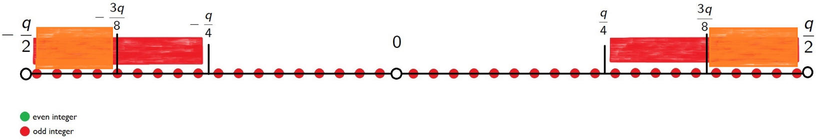


$-\frac{q}{2}$
$-\frac{q}{4}$
0
$\frac{q}{4}$
$\frac{q}{2}$
even integer
odd integer


$-\frac{q}{2}$
$-\frac{3q}{8}$
$-\frac{q}{4}$
0
$\frac{q}{4}$
$\frac{3q}{8}$
$\frac{q}{2}$
even integer
odd integer



$-\frac{q}{2}$
$-\frac{3q}{8}$
$-\frac{q}{4}$
0
$\frac{q}{4}$
$\frac{3q}{8}$
$\frac{q}{2}$
even integer
odd integer


$-\frac{q}{2}$
$-\frac{3q}{8}$
$-\frac{q}{4}$
0
$\frac{q}{4}$
$\frac{3q}{8}$
$\frac{q}{2}$
even integer
odd integer

Key Recovery Attack against [LATV12] - Details

Generalizing

- continue, and we find $s'_i := |s_i| \in [a_i, a_i + 1] \subseteq [0, \frac{q-1}{2}]$, for $i = 0, 1, \ldots, n-1$
- $|s_i|$ can assume at most only two (consecutive) values
- known parity $\Rightarrow$ determine $|s_i|$
- to achieve this we need $\lfloor \log_2 q \rfloor$ steps

Last step

- Left to find out whether $s_i \cdot s_j < 0$ or $s_i \cdot s_j > 0$, $\forall i, j$ with $s_i, s_j \neq 0$
- Let s_m be the first non-zero coefficient: we will obtain two possible candidates of sk, one with $s_m > 0$ and one with $s_m < 0$
- trivial oracle dec. query to determine which one is the correct sk
- omit details

Key Recovery Attack against SHE [BLLN13]

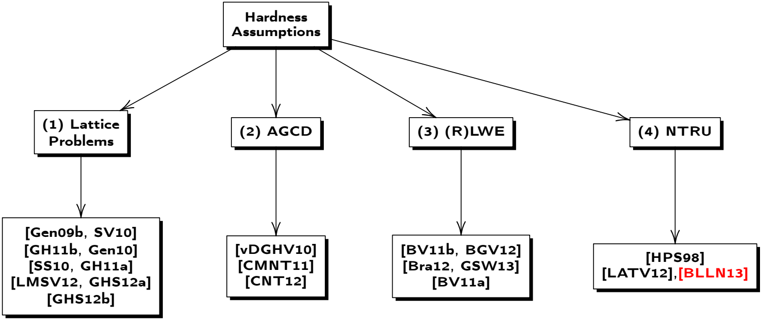


Figure : Hardness assumptions and relevant papers

Key Recovery Attack against [BLLN13]

Parameters Setup

- $\mathcal{M} = R/tR = \mathbb{Z}_t[x]/(x^n+1)$, $R = \mathbb{Z}[x]/(x^n+1)$
- d power of 2, $q \in \mathbb{N}$ prime integer, $t \in \mathbb{N}$ s.t. $1 < t < q$
- $\chi_{\text{key}}, \chi_{\text{err}}$ distributions on R
- operations on ciphertexts in $R_q := \mathbb{Z}_q[x]/(x^n+1)$

The [BLLN13] SHE scheme (informal)

KeyGen(λ) :

- $[\cdots]$
- set sk $:= f \in R_q$

Encrypt(pk, m):

- for message $m + tR$, let $[m]_t$ be its representative
- sample $s, e \leftarrow \chi_{\text{err}}$
- output ciphertext $c = [\lfloor q/t \rfloor [m]_t + e + hs]_q \in R_q$

Decrypt(sk, c):

- output $m = \left[\left\lfloor \frac{t}{q} \cdot [fc]_q \right\rceil \right]_t \in R_t$

Key Recovery Attack against [BLLN13]

KeyGen(λ) :

- $[\cdots]$
- set sk $:= f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{n-1}x^{n-1} \in R_q = \frac{\mathbb{Z}_q[x]}{(x^n+1)}$

Encrypt(pk, m):

- $[\cdots]$
- output $c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1}x^{n-1} \in R_q = \frac{\mathbb{Z}_q[x]}{(x^n+1)}$

Decrypt(sk, c):

- output $m = \left[\left\lfloor \frac{t}{q} \cdot [fc]_q \right\rceil\right]_t \in R_t$

Comparison with [DGM15]

- attack already exists in [DGM15], but require $6(t^2+t) < q$ and $B^2 < \dfrac{q}{36t^2}$ (conditions not assumed in [LATV12])
- our attack: works for all parameters. More efficient than [DGM15]:

	Our Attack	Attack from [DGM15]
(t is odd)	$\lceil \log_2(B/t) \rceil$	$n \cdot \lceil \log_2 B \rceil$
(t is even but not 2)	$\lceil \log_2(B/t) \rceil + n$	$n \cdot \lceil \log_2 B \rceil$
($t = 2$)	$\lceil \log_2(B/t) \rceil + n$	$n \cdot \lceil \log_2 B \rceil + n$

- n: power of 2; $B \ll q$ bound on coefficient of χ; $t \geq 2$ integer

Key Recovery Attack against [BLLN13]

KeyGen(λ) :

- $[\cdots]$
- set sk $:= f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{n-1} x^{n-1} \in R_q = \frac{\mathbb{Z}_q[x]}{(x^n+1)}$

Encrypt(pk, m):

- $[\cdots]$
- output $c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1} \in R_q = \frac{\mathbb{Z}_q[x]}{(x^n+1)}$

Decrypt(sk, c):

- output $m = \left[\left\lfloor \frac{t}{q} \cdot [fc]_q \right\rceil \right]_t \in R_t$

Key recovery attack - The main idea - we omit the details

- General idea: as usual, gradually reducing the interval in which the sk lie
- However, more complicated since we have to take into account and create several cases according to t odd, t even but $\neq 2$, and $t = 2$
- After each step k, f_i is determined up to an error $\frac{q}{2^k t}$
- we continue in this fashion until $\frac{q}{2^k t} \leq 1$

Conclusion and Future Directions

- SHE schemes from [BV11b, BV11a, BGV12, Bra12, GSW13, LATV12, BLLN13] suffer from key recovery attacks when the attacker is given access to the decryption oracle
- together with results from [LMSV12]: most existing SHE schemes suffer from key recovery attacks; not IND-CCA1 secure
- next step: to investigate whether it is possible to enhance these SHE schemes to avoid key recovery attacks and make them IND-CCA1 secure

Thank you for your attention!

massimo.chenal@uni.lu; qiang.tang@uni.lu

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan.
(leveled) fully homomorphic encryption without bootstrapping.
In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325. ACM, 2012.

Daniel Bleichenbacher.
Chosen ciphertext attacks against protocols based on the rsa encryption standard pkcs #1.
In *Advances in Cryptology - CRYPTO 1998*, pages 1–12, 1998.

Joppe W. Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig.
Improved security for a ring-based fully homomorphic encryption scheme.
In *Cryptography and Coding*, LNCS. Springer, 2013.

Mihir Bellare and Adriana Palacio.
Towards plaintext-aware public-key encryption without random oracles.
In *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 48–62. 2004.

Zvika Brakerski.
Fully homomorphic encryption without modulus switching from classical gapsvp.
In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *LNCS*, pages 868–886. 2012.

Zvika Brakerski and Vinod Vaikuntanathan.
Fully homomorphic encryption from ring-lwe and security for key dependent messages.
In *Advances in Cryptology - CRYPTO 2011*, pages 505–524, 2011.

Zvika Brakerski and Vinod Vaikuntanathan.

efficient fully homomorphic encryption from (standard) lwe.
In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 97–106, 2011.

JungHee Cheon, Jean-SÃ©bastien Coron, Jinsu Kim, MoonSung Lee, TancrÃ¨de Lepoint, Mehdi Tibouchi, and Aaram Yun.
Batch fully homomorphic encryption over the integers.
In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 315–335. 2013.

Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi.
Fully homomorphic encryption over the integers with shorter public keys.
In *Advances in Cryptology - CRYPTO 2011*, pages 487–504, 2011.

Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi.
Public key compression and modulus switching for fully homomorphic encryption over the integers.
In *Advances in Cryptology - EUROCRYPT 2012*, pages 446–464, 2012.

Massimo Chenal and Qiang Tang.
On key recovery attacks against existing somewhat homomorphic encryption schemes.
IACR Cryptology ePrint Archive, Report 2014/535, 2014.

Ricardo Dahab, Steven Galbraith, and Eduardo Morais.
Adaptive key recovery attacks on ntru-based somewhat homomorphic encryption schemes.
In *Information Theoretic Security - 8th International Conference, ICITS*, volume 9063 of *LNCS*, pages 283–296. Springer, 2015.

Jintai Ding and Chengdong Tao.
A new algorithm for solving the approximate common divisor problem and cryptanalysis of the fhe based on gacd.
IACR Cryptology ePrint Archive, Report 2014/042, 2014.

Craig Gentry.
A Fully Homomorphic Encryption Scheme.
PhD thesis, Stanford, CA, USA, 2009.

Craig Gentry.
Fully homomorphic encryption using ideal lattices.
In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178. ACM, 2009.

Craig Gentry.
Computing arbitrary functions of encrypted data.
Commun. ACM, 53(3):97–105, March 2010.

Craig Gentry and Shai Halevi.
Fully homomorphic encryption without squashing using depth-3 arithmetic circuits.
In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 107–109, 2011.

Craig Gentry and Shai Halevi.
Implementing gentry's fully-homomorphic encryption scheme.
In *Advances in Cryptology - EUROCRYPT 2011*, pages 129–148, 2011.

Craig Gentry, Shai Halevi, and Nigel P. Smart.
Better bootstrapping in fully homomorphic encryption.

In *Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography*, PKC'12, pages 1–16, 2012.

Craig Gentry, Shai Halevi, and Nigel P. Smart.
Fully homomorphic encryption with polylog overhead.
In *Advances in Cryptology - EUROCRYPT 2012*, pages 465–482, 2012.

Craig Gentry, Amit Sahai, and Brent Waters.
Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based.
In Ran Canetti and JuanA. Garay, editors, *Advances in Cryptology - CRYPTO 2013*, volume 8042 of *LNCS*, pages 75–92. 2013.

Jeffrey Hoffstein, Jill Pipher, and JosephH. Silverman.
Ntru: A ring-based public key cryptosystem.
In JoeP. Buhler, editor, *Algorithmic Number Theory*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer Berlin Heidelberg, 1998.

Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan.
On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption.
In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1219–1234, New York, NY, USA, 2012. ACM.

Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren.
On cca-secure somewhat homomorphic encryption.
In *Proceedings of the 18th International Conference on Selected Areas in Cryptography*, SAC'11, pages 55–72, 2012.

Daniele Micciancio and Chris Peikert.

Trapdoors for lattices: Simpler, tighter, faster, smaller.
IACR Cryptology ePrint Archive, Report 2011/501, 2011.

Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan.
Can homomorphic encryption be practical?
In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, pages 113–124, 2011.

Koji Nuida.
A simple framework for noise-free construction of fully homomorphic encryption from a special class of non-commutative groups.
IACR Cryptology ePrint Archive, Report 2014/097, 2014.

Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos.
On data banks and privacy homomorphisms.
Foundations of Secure Computation, Academia Press, pages 169–179, 1978.

Oded Regev.
On lattices, learning with errors, random linear codes, and cryptography.
In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, 2005.

Damien Stehle and Ron Steinfeld.
Faster fully homomorphic encryption.
In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 377–394. 2010.

N. P. Smart and F. Vercauteren.
Fully homomorphic encryption with relatively small key and ciphertext sizes.

In *Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography*, PKC'10, pages 420–443, 2010.

Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan.
Fully homomorphic encryption over the integers.
In *Advances in Cryptology - EUROCRYPT 2010*, pages 24–43, 2010.

Zhenfei Zhang, Thomas Plantard, and Willy Susilo.
On the cca-1 security of somewhat homomorphic encryption over the integers.
In *Proceedings of the 8th International Conference on Information Security Practice and Experience*, ISPEC'12, pages 353–368, 2012.