# A key recovery attack to the scale-invariant NTRU-based somewhat homomorphic encryption scheme

Eduardo Morais
Ricardo Dahab

October 2014

**Abstract**

In this paper we present a key recovery attack to the scale-invariant NTRU-based somewhat homomorphic encryption scheme proposed by Bos et al [BLLN13] in 2013. The attack allows us to compute the private key for $t > 2$ and when the private key is chosen with coefficients in $\{-1, 0, 1\}$. The efficiency of the attack is optimal since it requires just one decryption oracle query, showing that if we don't look for this kind of vulnerabilities in homomorphic encryption constructions we are likely to choose insecure parameters. The existence of a key recovery attack means that the scheme is not CCA1-secure. Indeed, almost every somewhat homomorphic construction proposed till now in the literature is vulnerable to this kind of attack, hence our result indicates that building CCA1-secure homomorphic schemes is not trivial. We also provide tables showing how the multiplicative depth is affected when the critical parameter $\mathcal{B}_{\text{key}}$ is chosen in order to mitigatte the attack.

## 1 Introduction

The construction of ***fully homomorphic encryption*** (FHE) was conjectured in 1978 by Rivest, Adleman and Dertouzos [RAD78]. Although it was immediately recognized as a very interesting possibility in Cryptography, no concrete construction was known until 2009, when Gentry used ideal lattices to settle this conjecture [Gen09].

In short, ciphertexts produced by an FHE scheme can be added or multiplied, in such a way that we obtain the corresponding addition or multiplication of the respective plaintexts. The ability to algebraically operate over ciphertexts is of great importance because we can transform any algorithm into a sequence of additions and multiplications. Therefore, such a scheme can evaluate any algorithm solely with access to the encryption of its input, and such that the computation returns the encryption of the output.

Since Gentry's work, many FHE constructions have appeared in the literature. However, all the proposals have a common drawback: they are not practical. Initially, the algorithms involved in the constructions, although having polynomial complexity, have high polynomial degree, which turned out to be an obstacle to be transposed. Later, asymptotic complexity became much better, indeed we have now constructions with polylog overhead per operation, but with terribly high constants.

Although fully homomophic encryption is not practical yet, many constructions have been proposed recently achieving a somewhat homomorphic encryption (SHE) scheme. These constructions are indeed very useful in practice, specially in order to provide security in the scenario of cloud computing. SHE is important also in the implementation of ***private information retrieval*** (PIR) protocols, which can be seen as a building block to the solution for the privacy problem that emerges when we give our data to the cloud.

In this work, we are going to present a ***key recovery*** attack to an efficient scale-invariant NTRU-based SHE scheme. Gentry's original construction is based on ideal lattices, which is a subarea of cryptography whose attention by the cryptology community has increased due to the advent of post-quantum cryptography, since quantum computers can perform no better than classical computers to solve lattices hard problems. Besides that, lattice-based cryptography is supported by worst-case security proofs, what is a strong argument in relation to conventional average-case security reductions. On the other hand NTRU is a practical lattice-based cryptosystem that remained without such a security proof for a long time, but recently this problem was solved [SS13] and NTRU-based cryptosystems appeared back as a fruitful research area. Lastly, scale-invariant homomorphic encryption was proposed by Brakerski [Bra12], presenting a construction that avoids the utilization of modulus switching technique, considerably simplifying the scheme.

### 1.1 Notation

An important technicality about the running time of the algorithms that will be presented is that they are required to be polynomial-time in the size of their input, therefore the security parameter $\lambda$ must be expressed in unary representation, otherwise the algorithm could have exponential complexity. Thus, whenever $\lambda$ appears it will mean the unary representation of the security parameter.

Notation $\lfloor x \rceil$ is used to round $x$ to the nearest integer, while notation $[x]_q$ is used to denote centralized modular reduction, i.e. reduction modulo $q$, but with result given in the interval $(-q/2, q/2]$.

Moreover, when working over a polynomial ring $R$, if $a(x) \in R$, we use the notation $a[i]$ to denote the i-th coefficient of the polynomial $a(x)$.

### 1.2 Paper organization

This paper is organized as follows. In section 2 we present basic definitions and details about the security model that will be used. In section 3 we described exactly how the SHE scheme BLLN is constructed. In section 4 we provide the main contribution of this paper, which is the key recovery attack and the parameters corrections that are necessary in order to avoid the attack. Finally, in section 5 we give further information about our implementation and give our concluding remarks.

## 2 Fundamentals and security model

In this section we are going to present basic concepts and the security model that we will use throughout the paper.

**Definition 1.** *We say the an adversary has* ***negligible*** *probability of success, if his probability of success is less than any inverse polynomial in the security parameter. In other words, for every constant $c$, the adversary's success probability is smaller than $n^{-c}$ for large enough values of $n$. In general, a function $f$ is called* ***negligible*** *if it holds that for every polynomial $p(n)$, then we have that there exists $N$ such that $f(n) < \frac{1}{p(n)}$, for all integers $n > N$.*

A cryptosystem is secure against ***chosen ciphertext attack*** (CCA2) if there is no polynomial time adversary that can win the following game with non negligible probability.

**Setup.** The challenger obtains $(\mathrm{sk}, \mathrm{pk}) = \textsc{KeyGen}(\lambda)$ and sends $\mathrm{pk}$ to the adversary $\mathcal{A}$.

**Queries.** $\mathcal{A}$ sends ciphertexts to the challenger, before or after the challenge, that returns the corresponding plaintexts.

**Challenge.** The adversary randomly generates two plaintexts $m_0, m_1 \in \mathcal{P}$ and sends to the challenger, that chooses randomly a bit $b \in \{0, 1\}$ and computes the ciphertext $c = \text{ENC}_{\text{pk}}(m_b)$. The challenger sends $c$ to $\mathcal{A}$.

**Answer** $\mathcal{A}$ sends a bit $b'$ to the challenger and wins the game if $b' = b$.

If we allow queries only before the challenge, we say that the cryptosystem is secure against CCA1 adversaries (lunchtime attacks). As previously described, queries can be interpreted as an access to a decryption oracle. If instead we only allow access to an encryption oracle, i. e. the adversary can choose any message that is distinct from $m_0$ and $m_1$ to be encrypted under the same key pair, then we say that the cryptosystem is secure against ***chosen plaintext attacks*** (CPA).

In homomorphic encryption, it is impossible to achieve CCA2 security, because the adversary can add an encryption of zero to the encrypted message, or multiply it by the encryption of one, and send it to the decryption oracle. Many FHE schemes have as public value an encryption of the private key bits, which can be sent to the decryption oracle before the challenge, what makes such schemes insecure against CCA1 adversaries. Indeed, a ***key recovery*** attack is stronger than an CCA1 attack and Loftus et al [LMSV11] showed that Gentry's construction over ideal lattices is vulnerable to it and presented the only SHE proposal that is known to be CCA1 secure.

Recently [CT14], Quenal and Tang showed that many SHE schemes are not CCA1 by presenting a key recovery attack. Although the straightforward application of the ideia is not possible in NTRU-based constructions, we are going to show that a different algebraic trick can be done.

From now on we are going to work over the same algebraic structure, that is given by the cyclotomic ring $R = \mathbb{Z}_q[x]/(x^d + 1)$, where $d$ is a power of 2. Cyclotomic rings are being used in lattice-based cryptography since the breakthrough from Lyubashesky, Peikert and Regev [LPR10]. Such rings lead to the definition of ***ideal lattices***, that corresponds to the lattices whose points are given by the coefficients of the cosets representative elements. Although there is no demonstration that ideal lattices maintain the same security guaranties as conventional lattices, but no significant improvement in ideal lattices hard problems complexities has appeared till now.

An important peculiarity of lattice-based cryptography is the utilization of intermediary problems, whose solution can be demonstrated to be as hard as solving worst-case problems like for example $\text{GAP}\mathbf{SVP}_\gamma$ and $\text{GAP}\mathbf{SIVP}_\gamma$, where $\gamma$ is the approximation factor. For instance, the decision version of the LWE problem can be reduced to $\text{GAP}\mathbf{SIVP}_\gamma$, with $\gamma$ a polynomial function in the security parameter. Hence, the LWE problem, detailed in definition 2, only can be efficiently solved if there is a polynomial solution to worst-case lattice problems.

**Definition 2.** *Given the security parameter $\lambda$, let $d$ and $q$ be integers depending on $\lambda$. Let $R_q = R/qR$ and the distribution $\mathcal{D}$ over $R_q$ be a Gaussian discrete distribution that depends on $\lambda$. Then, the LWE problem is to distinguish between the following two distributions: (i) $(a, b) \in R_q^2$, where $a$ and $b$ are chosen according to the uniform distribution and (ii) $(a, b) \in R_q^2$, where $a$ is uniform but $b = as + e$, where $s$ and $e$ are chosen using $\mathcal{D}$.*

## 3 NTRU-based somewhat homomorphic encryption

NTRU cryptosystem [HPS98] is an efficient lattice-based cryptographic scheme, but for many years, the lack of security proof, reducing its security to worst-case lattice hard problems, was the major concern about its utilization. Stehlé and Steinfeld [SS13] presented such a proof, replacing original ring $\mathbb{Z}_q[x]/(x^d - 1)$ by the previously described cyclotomic rings $\mathbb{Z}_q[x]/(x^d + 1)$, where $d$ is restricted to a power of 2.

In 2012, López-Alt, Tromer and Vaikuntanathan [LATV12] proposed the construction of ***multikey fully homomoprhic encryption***, called LTV scheme. The difference here is that users with distinct keys can compute ciphertexts that will be processed by a server in order to obtain the homomorphic evaluation of a determined function. It means that each user will be able to decrypt the function evaluation and this strategy can be followed to construct a multiparty computation scheme.

Doröz, Hu and Sunar [DHS14] customized LTV scheme, proposing a new construction called DHS scheme. They implemented homomorphic evaluation of AES using their proposal, showing that it offers advantage against the BGV scheme [BGV11].

However, LTV scheme is based on non-standard assumptions. In 2013, a scale-invariant NTRU-based scheme was proposed by Bos et al [BLLN13], abbreviated by BLLN scheme. The basic scheme, $\mathcal{E}_{\text{basic}}$, can be described as follows:

**Definition 3.** ***Setup.*** *Given the security parameter* $\lambda$ *and the ring* $R = \mathbb{Z}[x]/\phi_d(x)$*, for a fixed* $d$*, where* $d$ *is a power of two. Let* $\mathcal{D}_{\text{key}}$ *and* $\mathcal{D}_{\text{err}}$ *be distributions on* $R$*.* SETUP *algorithm returns* $(t, d, q, \mathcal{D}_{\text{key}}, \mathcal{D}_{\text{err}})$*.*

***Key generation.*** *Given the output of* SETUP *algorithm, sample polynomials* $f', g \leftarrow \mathcal{D}_{\text{key}}$ *and compute* $f = [tf' + 1]_q$*. Verify if* $f$ *is invertible modulo* $q$*, otherwise choose a new* $f'$*. Compute the inverse* $f^{-1} \in R$ *and* $h = [tgf^{-1}]_q$*. The public key is given by* $pk = h$ *and the private key is given by* $sk = f$*. Algorithm* KEYGEN *returns* $(sk, pk)$*.*

***Encryption.*** *The plaintext space is* $R/tR$*, then a message is given by a coset* $m + tR$*. Compute* $[m]_t$ *as the representative element of the coset. Sample* $s, e \leftarrow \mathcal{D}_{\text{err}}$ *and compute the ciphertext*

$$c = \text{ENC}_{pk}(m) = \left[\lfloor q/t \rfloor [m]_t + e + hs\right]_q .$$

***Decryption.*** *Return the message*

$$m = \text{DEC}_{sk}(c) = \left[\lfloor (t/q).\, [fc]_q \rceil\right]_t .$$

The security of this scheme is based on an analysis from Gentry et al [GHS12], which in turn used parameters presented in the work from Lindner and Peikert [LP11], showing that the scheme is secure as long as the LWE problem parameters $d, q, \sigma$ respect the following inequality

$$d > \log\left(\frac{q}{\sigma}\right)\frac{\lambda + 110}{7.2}.$$

When applied with homomorphic schemes, this relation acquires a challenging aspect, because as the standard deviation increases, less homomorphic operations can be evaluated, since a larger initial noise would be rapidly propagated, what would require a larger modulus $q$, conforming to the circuit level. Thus, as the ratio $q/\sigma$ determines the LWE-based cryptography security, in order to avoid managing the growth of such weakly related functions, for instance the inter-dependent values of $d$, $q$ and $\sigma$, we can fix a sufficiently large minimum value for $\sigma$, such that attacks that explore small standard deviations are mitigated [AG11].

Distribution $\mathcal{D}_{\text{key}}$ must be chosen according to Stehlé and Steinfeld description [SS13], such that the public key is close enough to the uniform distribution and then reveals almost nothing about the private key. Rigorously, it reveals only a negligible fraction of the secret. Thus, $\mathcal{D}_{\text{key}}$ is a discrete Gaussian on $R_q$ with standard deviation at least $(d\sqrt{\log 8dq})q^k$, for $k$ in the interval $(1/2, 1)$. Furthermore, $\mathcal{D}_{\text{err}}$ is a $\omega(\sqrt{d \log(d)})$-bounded Gaussian distribution, what makes it secure based on the LWE problem.

# 4 Key recovery attack and adjustments

In a key recovery attack, we submit appropriately chosen ciphertexts to a decryption oracle in order to retrieve information about the private key. If we can do that, then we can also solve the challenge in the CCA1 game. Consequently, we have that the key recovery attack is stronger than a CCA1 attack, since if the adversary can calculate the private key having access to a decryption oracle before the challenge, then he can decrypt the challenge and trivially solve it.

In the original paper, BLLN's authors stated that we can choose $f'$ and $g$ with coefficients in $\{-1, 0, 1\}$. But as we are going to show next, when the private key is obtained like that, we can easily compute it using just one query to the decryption oracle. As an starting remark, because $f = tf' + 1$. we have that $f$ has coefficients in $\{-t, 0, t\}$, except for the independent coefficient, which belongs to $\{-t + 1, 1, t + 1\}$.

**Theorem 1.** *Let $m_f = \text{DEC}(\lfloor q/t^2 \rfloor)$ be a polynomial that belongs to $R/tR$, where $\lfloor q/t^2 \rfloor$ is a constant integer polynomial that can easily be computed using the public parameters $q$ and $t$. Then we have that $f = m_f + 1$ for $t > 2$ and such that $t \ll q$, as necessary to ensure correctness of the SHE scheme.*

*Proof.* We have that $\text{DEC}(\lfloor q/t^2 \rfloor) \equiv [\lfloor (t/q).[f.(\lfloor q/t^2 \rfloor)]_q \rceil]_t$. Because we are multiplying $f$ by a constant polynomial, each coefficient of $f$ is multiplied by $\lfloor q/t^2 \rfloor$. Moreover, if the coefficient of $f$ is $-t$, $0$ or $t$, then $[f(q/t^2)]_q$ gives us a number that is in the interval $(-q/2, q/2]$, what allows us to conclude that $[f(\lfloor q/t^2 \rfloor)]_q = f(\lfloor q/t^2 \rfloor)$. Therefore, after multiplying by $t/q$ we obtain an element from $\{-1, 0, 1\}$ in $R/tR$ that corresponds exactly to the coefficients of $f' \in R/qR$ (but remember that $-1 \pmod q$ is different from $-1 \pmod t$, in our case the implementation is done using NTL library, and then we have that $-1 \equiv q - 1 \pmod q$ and $-1 \equiv t - 1 \pmod t$). The restriction $t > 2$ comes from the fact that $-1 \equiv 1 \pmod 2$, then we can't distinguish between $-1$ and $1$ in order to compute $f$. □

Next we present the details of the algorithm. We emphasize that the attack is very fast, since it needs to perform just one query to the decryption oracle. Also, the ciphertext that we must submit to the decryption oracle is very simple to construct.

**Algorithm 4.1** Attack

```
INPUT The public parameters $(q, t, n)$.
OUTPUT The private key $f$.
  $m_f = \text{DEC}(\lfloor q/t^2 \rfloor)$.
  for $i = 0$ till $n$ do
    if $(m_f[i] \equiv 1 \pmod q)$ then
      $f[i] = t$.
    else
      if $(m_f[i] \equiv -1 \pmod q)$ then
        $f[i] = -t$.
      else
        $f[i] = 0$.
  return $f + 1$.
```

We have two remaining challenges: (i) the case $t = 2$ and (ii) the case that $f'$ and $g$ are chosen from $\mathcal{D}_{\text{key}}$ with larger standard deviation. The other two previously mentioned NTRU-based constructions, for instance LTV and DHS, are exactly in this situation, because we have that $t$ is in fact restricted to 2 and the private key is not restricted to $\{-1, 0, 1\}$. Considering that the encoding proposed by Bos at al [BLLN13] depends on the fact that $t \gg 2$ ($t = 1024$ for

| $n$ | $\log(q_{\max})$ | $t$ | $L_{\max}$ |
|---|---|---|---|
| 2048 | 79 | 2 | 3 |
| | | 256 | 2 |
| | | 1024 | 1 |
| 4096 | 157 | 2 | 5 |
| | | 256 | 3 |
| | | 1024 | 3 |
| 8192 | 312 | 2 | 10 |
| | | 256 | 7 |
| | | 1024 | 6 |
| 16384 | 622 | 2 | 20 |
| | | 256 | 13 |
| | | 1024 | 12 |
| 32768 | 1243 | 2 | 37 |
| | | 256 | 26 |
| | | 1024 | 24 |
| 65536 | 2485 | 2 | 71 |
| | | 256 | 50 |
| | | 1024 | 47 |

Table 1: $B_{\text{key}} = 2$

| $n$ | $\log(q_{\max})$ | $t$ | $L_{\max}$ |
|---|---|---|---|
| 2048 | 79 | 2 | 2 |
| | | 256 | 1 |
| | | 1024 | 1 |
| 4096 | 157 | 2 | 5 |
| | | 256 | 3 |
| | | 1024 | 3 |
| 8192 | 312 | 2 | 10 |
| | | 256 | 6 |
| | | 1024 | 6 |
| 16384 | 622 | 2 | 18 |
| | | 256 | 13 |
| | | 1024 | 12 |
| 32768 | 1243 | 2 | 35 |
| | | 256 | 25 |
| | | 1024 | 23 |
| 65536 | 2485 | 2 | 66 |
| | | 256 | 48 |
| | | 1024 | 45 |

Table 2: $B_{\text{key}} = 10$

example) in order to do operations homomorphically, then it would be important to use private keys with larger standard deviation. The case $t = 2$ allows us to homomorphically operate over bits, but it doesn't permit us to enjoy the advantages of the integer encoding. Thus it would be interesting to investigate the practical impact of making the standard deviation wider in the choice of the private key coefficients. If we keep the same parameters that were chosen in the original paper, but change the value of $\mathcal{B}_{\text{key}}$ in order to mitigate the attack just presented, then we obtain the parameters described in table 1 and table 2, for $\mathcal{B}_{\text{key}} = 2$ and $\mathcal{B}_{\text{key}} = 10$ respectively. Furthermore, we have that $\mathcal{B}_{\text{err}} = 6\sigma_{\text{err}} = 48$. For instance, the parameters must satisfy the following condition:

$$(1+\epsilon_1)^{L-1} n^{2L} t^{2L-1} \mathcal{B}_{\text{key}}^{L}((1+\epsilon_1)tV + L(t(\mathcal{B}_{\text{key}} + t) + \ell_{w,q} w \mathcal{B}_{\text{err}}))$$

must be less than $\Delta - (q \pmod{t})/2$, where $\epsilon_1 = 4(nt\mathcal{B}_{\text{key}})^{-1}$ and $V = nt(\mathcal{B}_{\text{key}}(2\mathcal{B}_{\text{err}} + (q \pmod{t}))/2)$, for $\mathcal{B}_{\text{key}}$ and $\mathcal{B}_{\text{err}}$ bounds to the absolute value of elements from distribuitions $\mathcal{D}_{\text{key}}$ and $\mathcal{D}_{\text{err}}$. Lastly, $w$ is the window-size used in the generalized version of the functions $\text{PowerOf2}$ and $\text{BitDecomp}$ and $\ell_{w,q} = \lfloor \log_w(q) \rfloor + 2$ (details follow in the original paper [BLLN13]).

On the other hand, if $t = 2$ our problem is to find out the sign of each coefficient. Concretely, if we use $q/t$ instead of $q/t^2$ to query the decryption oracle, then we can detect when the coefficients of $f'$ are zero, but when they are 1 or $-1$, because we are operating modulo 2, then we always obtain that non-zero coefficients are equal to 1. This computation corresponds to a partial attack, because it is still necessary to find out the sign of non-zero coefficients. However, if the distribution $\mathcal{D}_{\text{key}}$ is fact Gaussian, then zero elements have higher probability to appear. In a recent work [MP13], Micciancio and Peikert studied the utilization of uniform distributions in the intermediary problems LWE and SIS, showing that solving instances of these problems, even when errors are chosen according to uniform distribuitions in the set $\{0, 1\}$, remains hard. Although the reduction from LWE to lattice problems still works, obtaining a

CCA1-secure cryptosystem requires a decryption algorithm that "shuffles" the private key coefficients. Cyclotomic polynomials are multiplied using the convolution operation, which do a relatively good job in order to mix the coefficients, but the attack is possible because we can submit a constant polynomial to the decryption oracle, such that the multiplication returns just a scaling of the coefficients of $f$. Modifying the decryption to return an invalid tag for this kind of ciphertexts is not a good solution, as other queries may still exist. To definitely solve the problem we must present a scheme that even with a polynomial number of queries allowed to the adversary, he can not compute the private key.

The moral is that a decryption oracle is very powerful in homomorphic encryption, because in general, the decryption must be accomplished by a simple operation, in order to have low multiplicative depth, and also to preserve the homomorphism. Thus decryption usually is done by applying such simple operations between the ciphertext and the private key, like for example scalar product and polynomial multiplication. Hence it seems easy to choose an appropriate value for the ciphertext to obtain information about the private key.

## 5 Implementation and concluding remarks

We have implemented BLLN scheme using NTL library [NTL] for polynomial ring arithmetic and using GMP library [GMP] for efficient big number arithmetic. If $t = 1024$, for example, it is possible to encode integers as in the original paper [BLLN13] (or as recently presented by Geihs and Cabarcas [GC14]). This kind of encoding is interesting because it allows computations over big integers instead of over bits or short integers. It is possible also to encode real numbers in a straightfoward manner, by considering a scaling factor and discarding bits of precision after multiplications, in order to maintain the scaling factor correct, as pointed out by Lauter, Alt-López and Naehrig [LLAN14]. In another work from the homomorphic encryption group at Microsoft Research, they presented an application of BLLN scheme to protect medical data [BLN14]. Unfortunately, we must choose the private key with a larger standard deviation, what turns out to decrease the multiplicative depth that the scheme can homomorphically evaluate.

We also have measured the performance of our implementation. Considering 80 bits of security, with the configuration given by $n = 4096$, $q$ a moduli with bit-length equal to 157, $w = 2^{32}$ and $\ell_{w,q} = 6$, the key generation algorithm took $5.37$ seconds to run, the encryption algorithm took $12$ ms, while decryption took $10$ ms. For homomorphic operations, we have that addition took $0.08$ ms, while multiplication took $511$ ms. Interestingly, using $w = 2$, we have that $\ell_{w,q} = 9$, leading to a larger $\mathrm{SwitchKey}$, turning multiplications considerably slower. For instance, we have that multiplication took $4.08$ seconds to run when $w = 2$, one order of magnitude larger than the case where $w = 2^{32}$. These timings were taken in a regular desktop with $4$ Gb of RAM memory and a $2.7$ GHz processor.

We have described a key recovery attack on the scale-invariant NTRU-based SHE scheme to the case where $t > 2$ and private key coefficients are chosen from $\{-1, 0, 1\}$. This attack shows that CCA1 security is hard to be achieved in homomorphic encryption. However we have also provided tables with new parameters that show how they can be changed to mitigate the attack. As we can see, the multiplicative depth that can be homomorphically evaluated is not dramatically changed, what constitutes an incentive to keep looking for optimizations to NTRU-based SHE schemes. Nevertheless, we remark that the assessment of the impact of key recovery attack for homomorphic encryption is unprecedent, since obtaining private key information from decryption oracles has become a standard way to find attacks in this area of research.

## References

[AG11] S. Arora and R. Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

[BGV11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:111, 2011.

[BLLN13] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In Martijn Stam, editor, *IMA Int. Conf.*, volume 8308 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2013.

[BLN14] J. W. Bos, K. Lauter, and M. Naehrig. Private predictive analysis on encrypted medical data. Cryptology ePrint Archive, Report 2014/336, 2014. `http://eprint.iacr.org/`.

[Bra12] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Advances in Cryptology - Crypto 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012.

[CT14] M. Chenal and Q. Tang. On key recovery attacks against existing somewhat homomorphic encryption schemes. In *Latincrypt*, Florianópolis-SC, Brazil, 2014.

[DHS14] Y. Doröz, Y. Hu, and B. Sunar. Homomorphic aes evaluation using NTRU. Cryptology ePrint Archive, Report 2014/039, 2014. `http://eprint.iacr.org/`.

[GC14] M. Geihs and D. Cabarcas. On key recovery attacks against existing somewhat homomorphic encryption schemes. In *Latincrypt*, Florianópolis-SC, Brazil, 2014.

[Gen09] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[GHS12] C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the aes circuit. *IACR Cryptology ePrint Archive*, 2012:99, 2012.

[GMP] GMP website. `https://gmplib.org/`. Accessed: 2014-10-13.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Lecture Notes in Computer Science*, pages 267–288. Springer-Verlag, 1998.

[LATV12] A. López-Alt, E. Tromer, and V. 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.

[LLAN14] K. Lauter, A. Lopez-Alt, and M. Naehrig. Private computation on encrypted genomic data. Technical Report MSR-TR-2014-93, June 2014.

[LMSV11] J. Loftus, A. May, N. P. Smart, and F. Vercauteren. On CCA-secure somewhat homomorphic encryption. In *In Selected Areas in Cryptography*, pages 55–72, 2011.

[LP11] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Proceedings of the 11th International Conference on Topics in Cryptology: CT-RSA 2011*, CT-RSA'11, pages 319–339, Berlin, Heidelberg, 2011. Springer-Verlag.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *Advances in Cryptology EUROCRYPT 2010*, 6110/2010(015848):1?23, 2010.

[MP13] D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. *CRYPTO*, 8042:21–39, 2013.

[NTL] NTL website. `http://www.shoup.net/ntl/`. Accessed: 2014-10-13.

[RAD78] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978.

[SS13] D. Stehlé and R. Steinfeld. Making NTRUEncrypt and NTRUSign as secure as standard worst-case problems over ideal lattices. *IACR Cryptology ePrint Archive*, 2013:4, 2013.