Analysis and Improvements of NTRU Encryption Paddings

Phong Q. Nguyen and David Pointcheval

CNRS/Département d'informatique, École normale supérieure
45 rue d'Ulm, 75005 Paris, France
{phong.nguyen,david.pointcheval}@ens.fr
http://www.di.ens.fr/~{pnguyen,pointche}

Abstract. NTRU is an efficient patented public-key cryptosystem proposed in 1996 by Hoffstein, Pipher and Silverman. Although no devastating weakness of NTRU has been found, Jaulmes and Joux presented at Crypto '00 a simple chosen-ciphertext attack against NTRU as originally described. This led Hoffstein and Silverman to propose three encryption padding schemes more or less based on previous work by Fujisaki and Okamoto on strengthening encryption schemes. It was claimed that these three padding schemes made NTRU secure against adaptive chosen-ciphertext attacks (IND-CCA2) in the random oracle model. In this paper, we analyze and compare the three NTRU schemes obtained. It turns out that the first one is not even semantically secure (IND-CPA). The second and third ones can be proven IND-CCA2–secure in the random oracle model, under however rather unusual assumptions. They indeed require a partial-domain one-wayness of the NTRU one-way function which is likely to be a stronger assumption than the one-wayness of the NTRU one-way function. We propose several modifications to achieve IND-CCA2–security in the random oracle model under the original NTRU inversion assumption.

1 Introduction

The NTRU cryptosystem [13], patented by the company NTRU Cryptosystems (see http://www.ntru.com), is one of the fastest public-key encryption schemes known. Although this may not be a decisive advantage compared to hybrid encryption with say RSA, NTRU has attracted considerable interest and is being considered by the *Efficient embedded security standards* [6] and the *IEEE P1363 study group for future public-key cryptography standards* [18]. It is therefore important to know how to use the NTRU cryptosystem properly.

The security of NTRU is based on the hardness of some lattice problems, namely the shortest and closest vector problems (see for instance the survey [24]). More precisely, it was first noticed by Coppersmith and Shamir [3] that ideal lattice basis reduction algorithms could heuristically recover NTRU's private key from the public key. This does not necessarily imply that NTRU is insecure, as currently known lattice basis reduction algorithms (such as LLL [20] or its improvements [28]) do not seem to perform sufficiently well in practice in very high dimension, while NTRU is so far the only lattice-based cryptosystem that can cope with high dimensions without sacrificing performances. Nor does it mean that the security of NTRU is strictly equivalent to the hardness of lattice problems, although the basic NTRU problem is equivalent to the lattice shortest vector problem in a very particular class of lattices called modular convolution lattices in [21].

The NTRU cryptosystem as originally described is easily seen to be semantically insecure. At Crypto '00 [19], Jaulmes and Joux further presented simple chosen-ciphertext attacks that can recover the private key. This shows that the NTRU cryptosystem as originally described should be viewed as a probabilistic trapdoor one-way function rather than a probabilistic cryptosystem (see also recent work by Micciancio on lattice-based cryptosystems [22]). NTRU Cryptosystems therefore proposed three padding schemes (two in [16] and a third one in [15]) to make NTRU secure against adaptive chosen-ciphertext attacks in the random oracle model. No security proof was provided by NTRU Cryptosystems (see [16, 15]). In this paper, we analyze the three NTRU schemes obtained. It turns out that the first scheme is not even semantically secure (IND-CPA). The second and third ones can be proven IND-CCA2–secure, in the random oracle model, but under rather unusual assumptions. Indeed, a partial-domain one-wayness of the NTRU one-way function is required, and that assumption is likely to be stronger than the one-wayness of the NTRU one-way function, as opposed to the situation of RSA (see [9]). Besides, the security proofs we obtain for such paddings are not efficient enough to be meaningful for the parameters recommended by NTRU Cryptosystems.

We therefore propose and compare new paddings to make NTRU IND-CCA2–secure in the random oracle model under the basic NTRU assumption, and not a stronger assumption: The new paddings give rise to better bounds for the security proof, and their computational overhead appears to be negligible. It should be stressed that no security proof in the standard model is known for NTRU, and that the search for an efficient and secure NTRU padding scheme is not a trivial matter. Although there now exist generic padding schemes (such as REACT [25]) that can enhance the security (in the random oracle model) of any cryptosystem, the case of NTRU differs from more usual cryptosystems such as RSA or El Gamal because the cost of hashing is no longer negligible compared to the cost of encryption and decryption, and because of special properties of the NTRU trapdoor function.

The rest of the paper is organized as follows. In Section 2, we recall security notions for public-key encryption schemes. In Section 3, we review the NTRU primitive and related computational assumptions. In Section 4, we present and analyze the various paddings proposed by NTRU. In Section 5, we consider new paddings and compare several constructions which make NTRU IND-CCA2–secure in the random oracle model.

2 Public-Key Encryption

The goal of encryption schemes is to achieve confidentiality of communications. In the public-key scenario, anyone knowing Alice's public key pk can send Alice a message that only she will be able to recover, thanks to her private key sk.

2.1 Definitions

A public-key encryption scheme Π over a message space $\mathcal{S}_{\mathsf{M}}$ is formally defined by three algorithms:

- a *key generation algorithm* $\mathcal{K}(1^k)$ (k being the security parameter), which produces a pair $(\mathsf{pk}, \mathsf{sk})$ of public and private keys.
- an *encryption algorithm* $\mathcal{E}_{\mathsf{pk}}(m; r)$ which outputs a ciphertext c corresponding to the plaintext $m \in \mathcal{S}_{\mathsf{M}}$, using random coins $r \in \mathcal{S}_{\mathsf{R}}$, according to the public key pk.
- a *decryption algorithm* $\mathcal{D}_{\mathsf{sk}}(c)$ which outputs the plaintext m associated to the ciphertext c (or $\perp$, if c is an invalid ciphertext), given the private key sk.

2.2 Security Notions

The simplest security notion is *one-wayness*: with public data only, an attacker cannot recover the whole plaintext m of a given ciphertext c. More formally, the success of any adversary $\mathcal{A}$ in inverting $\mathcal{E}_{\mathsf{pk}}$ without knowledge of the private key should be negligible over the probability space $\mathcal{S}_{\mathsf{M}} \times \mathcal{S}_{\mathsf{R}}$, and the internal random coins of the adversary and the algorithms $\mathcal{K}$ and $\mathcal{E}$:

$$\mathsf{Succ}^{\mathsf{ow}}_{\Pi}(\mathcal{A}) = \Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), c = \mathcal{E}_{\mathsf{pk}}(m; r) : \mathcal{A}(\mathsf{pk}, c) = m].$$

However, many applications require a higher security level, such as *semantic security* (*a.k.a. indistinguishability of encryptions* [11], denoted IND): if an attacker has some information about the plaintext, the view of the ciphertext should not leak any additional information. This security notion requires the computational intractability of winning with probability significantly better than 1/2 the following game: the adversary chooses two messages; the challenger selects at random one of these two messages, encrypts it, and sends the ciphertext to the adversary; the adversary guesses which one of the two messages has been encrypted. In other words, an adversary is seen as a 2-stage Turing machine (A_1, A_2), and the advantage $\mathsf{Adv}^{\mathsf{ind}}_{\Pi}(\mathcal{A})$ should be negligible for any adversary, where the advantage is formally defined as:

$$2 \times \Pr_{r \overset{R}{\leftarrow} \mathcal{S}_{\mathsf{R}}} \left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (m_0, m_1, s) \leftarrow A_1(\mathsf{pk}), \\ b \overset{R}{\leftarrow} \{0, 1\}, c = \mathcal{E}_{\mathsf{pk}}(m_b; r) : A_2(m_0, m_1, s, c) = b \end{array} \right] - 1.$$

Another important security notion is *non-malleability* [5]. Here, we ask that an adversary, given a ciphertext, should not be able to create a new ciphertext such that the two plaintexts are meaningfully related. This notion is, in general, stronger than semantic security, but it was shown [1] to be equivalent in the strongest scenario (see below).

On the other hand, an attacker can use many kinds of attacks, depending on the information available to him. First, in the public-key setting, the adversary can encrypt any plaintext of its choice with the public key: this basic scenario is called *chosen-plaintext attack*, and denoted by CPA. Extended scenarios allow the adversary restricted or unrestricted access to various oracles:

- a *validity-checking oracle*, which answers whether or not its input c is a valid ciphertext. This leads to so-called *reaction attacks* [12].
- a *plaintext-checking oracle*, which given as input a pair (m, c) answers whether or not the ciphertext c is a ciphertext of the message m. This gives rise to *plaintext-checking attacks* [25], which we denote by PCA.
- a *decryption oracle*, which returns the decryption of any ciphertext, with the only restriction that it should be different from the challenge ciphertext. When the oracle is available only before knowledge of the challenge ciphertext, the attack is a *non-adaptive chosen-ciphertext attack* (*a.k.a. lunchtime attack* [23]), which we denote by $\mathsf{CCA1}$. When the adversary still has access to the decryption oracle in the second stage, we talk about *adaptive chosen-ciphertext attacks* [27], denoted by $\mathsf{CCA2}$.

The article [1] provides a general study of all these security notions and attacks. Its main result states that semantic security and non-malleability are equivalent in the $\mathsf{CCA2}$–scenario. This security level is now widely accepted as the standard notion of security to be achieved by a public-key encryption scheme, and is sometimes called *chosen-ciphertext security.*

3 The NTRU Primitive

3.1 Description and notation

In this section, we present the NTRU cryptosystem as originally described in [13] by Hoffstein, Pipher and Silverman. As mentioned in the introduction, this should be viewed as a primitive function rather than a cryptosystem. Several modifications of NTRU have recently been proposed in [15]: we will present those later.

Let k be the security parameter. The NTRU primitive works in the ring $\mathcal{P} = \mathbb{Z}[X]/(X^N - 1)$ where N is a safe prime typically around a few hundreds, whose value increases with k: NTRU Cryptosystems recommends specific values of N, such as $N = 251$ or $N = 503$. The ring $\mathcal{P}$ is identified with the set of integer polynomials of degree $< N$, and its multiplication is denoted by $*$. Polynomials will be denoted by letters in the Sans Serif font, such as f.

Note that the function that maps any polynomial $\mathsf{f} \in \mathcal{P}$ to the sum $\mathsf{f}(1) \in \mathbb{Z}$ of its coefficients is a ring homomorphism. NTRU uses two integer parameters: a small power of 2, denoted by q, such as $q = 128$ or $q = 256$, and a small integer

$p < q$ co-prime with q, such as $p = 3$. The restriction of N to safe primes was apparently made to guarantee that the multiplicative order of p modulo N is sufficiently large. NTRU performs operations in $\mathcal{P}$ modulo p or q.

The private key sk consists of two polynomials $\mathsf{f}, \mathsf{g} \in \mathcal{P}$ randomly chosen with very small coefficients such that f is invertible modulo both p and q: there exist f_p and f_q in $\mathcal{P}$ such that: $\mathsf{f} * \mathsf{f}_p \equiv 1 \pmod{p}$ and $\mathsf{f} * \mathsf{f}_q \equiv 1 \pmod{q}$. In [13], f and g only have coefficients in $\{0, \pm 1\}$ with a prescribed (publicly known) number of 1, -1 and 0 such that $\mathsf{f}(1) = 1$ and $\mathsf{g}(1) = 0$. All the integer constraints (N, p, q, the number of 1, -1, 0 in polynomials, *etc.*) are deduced from a security parameter k, and this process is described in [13, 15]. Then, f and g are uniformly distributed polynomials among the polynomials that satisfy the required constraints. The public key pk is $\mathsf{h} = \mathsf{g} * \mathsf{f}_q \pmod{q}$. Therefore, $(\mathsf{h}, (\mathsf{f}, \mathsf{g})) \leftarrow \mathcal{K}(1^k)$.

The message space is $\mathcal{S}_{\mathsf{M}} = \{-(p-1)/2 \ldots +(p-1)/2\}^N$, whose elements are viewed as elements of $\mathcal{P}$. To encrypt a message m, one selects at random a sparse polynomial $\mathsf{r} \in \mathcal{P}$ with very small coefficients: In [13], r only has coefficients in $\{0, \pm 1\}$ with a prescribed (publicly known) number of 1, -1 and 0 such that $\mathsf{r}(1) = 0$ (we denote the set of these specific polynomials by $\mathcal{S}_{\mathsf{R}}$). The ciphertext is:

$$\mathcal{E}_{\mathsf{pk}}(\mathsf{m}; \mathsf{r}) = \mathsf{e} = \mathsf{m} + p\mathsf{r} * \mathsf{h} \pmod{q}.$$

To decrypt, the following congruence is used:

$$\mathsf{e} * \mathsf{f} \equiv \mathsf{m} * \mathsf{f} + p\mathsf{r} * \mathsf{g} \pmod{q}.$$

In the right-hand part of this congruence, we have two convolution products of polynomials with very small coefficients and quite a few zeroes (except possibly m). Therefore, if the above reduction is centered (one takes the smallest residue in absolute value), the above congruence is likely[1] to be an equality over $\mathbb{Z}[X]$. By further reducing $\mathsf{e} * \mathsf{f}$ modulo p, one thus obtains $\mathsf{m} * \mathsf{f} \pmod{p}$, hence m thanks to f_p. Note that there is a potential probability of decryption failure, if the above equality $\pmod{q}$ does not hold in $\mathbb{Z}$.

3.2 Efficiency

A multiplication in $\mathcal{P}$ requires $\mathcal{O}(N^2 \log q)$ elementary operations. It follows that the cost of encryption and decryption is $\mathcal{O}(N^2 \log q)$. Since the key generation process is such that $q = \mathcal{O}(N)$, the cost of both encryption and decryption is almost quadratic in the security parameter. Note that in most of the required convolution products, at least one of the polynomial is relatively sparse. And since q is a small power of two, the above complexity is rather pessimistic in practice.

[1] We stress that no provable precise estimate on the probability of such an event is known, and [13] only uses a heuristic estimate which seems to be validated by practice.

3.3 Optimizations

The authors of NTRU recently proposed several modifications in [15] to improve the efficiency of the scheme:

- Choosing p as an appropriate polynomial p instead of a small number co-prime with q (*e.g.* $\mathsf{p} = X + 2$). The polynomial must be such that the ideal spanned by the polynomial must be co-prime with the ideal $\langle q \rangle$ spanned by q in $\mathcal{P}$. The aim of this modification is to reduce the probability of decryption failure. It also enables a simpler encoding of messages.
- Selecting f, g, and r with a special form instead of just a prescribed number of 0, -1 and 1. For instance, $\mathsf{r} = \mathsf{r}_1\mathsf{r}_2$ where r_1 and r_2 are sparse polynomials with a prescribed number of 0, -1 and 1. In all the proposals of [15], the values of $\mathsf{f}(1)$, $\mathsf{g}(1)$ and $\mathsf{r}(1)$ are always publicly known.

It is worth noting that such modifications may have an impact on the security of NTRU, but at the moment, no specific attack is known.

Our results apply to the original NTRU scheme, as well as to most of these optimizations of NTRU. However, to simplify the presentation, we will restrict to the case of the original NTRU primitive $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where $p = 3$, $\mathsf{f}(1) = 1$ and $\mathsf{g}(1) = \mathsf{r}(1) = 0$.

3.4 Computational assumptions

To formally analyze the security of the NTRU cryptosystem, one needs clear and well-defined computational assumptions. First, we consider the one-wayness of the NTRU primitive:

Definition 1 (The NTRU Inversion Problem). *For a given security parameter k, which specifies N, p, q and several other constraints, as well as $(h, (f, g)) \leftarrow \mathcal{K}(1^k)$ and $e = m + pr * h$, where $m \in \mathcal{S}_\mathsf{M}$ and $r \in \mathcal{S}_\mathsf{R}$, find m.*

For any adversary $\mathcal{A}$, we denote by $\mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(\mathcal{A})$ its success for breaking the *NTRU Inversion Problem*, where

$$\mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(\mathcal{A}) = \Pr\left[\begin{matrix}(\mathsf{h}, (\mathsf{f}, \mathsf{g})) \leftarrow \mathcal{K}(1^k), \mathsf{m} \in \mathcal{S}_\mathsf{M}, \mathsf{r} \in \mathcal{S}_\mathsf{R}, \\ \mathsf{e} = \mathsf{m} + p\mathsf{r} * \mathsf{h} : \mathcal{A}(\mathsf{e}, \mathsf{h}) = \mathsf{m}\end{matrix}\right].$$

The *NTRU assumption* says that the NTRU inversion problem is hard to solve for any sufficiently large parameter.

Next, we consider the difficulty of only partially inverting this function, which will be useful to study NTRU paddings:

Definition 2 (The NTRU λ-Partial-Domain Inversion Problem). *For a given security parameter k, which specifies N, p, q and several other constraints, as well as $(h, (f, g)) \leftarrow \mathcal{K}(1^k)$ and $e = m + pr * h$, where $m \in \mathcal{S}_\mathsf{M}$ and $r \in \mathcal{S}_\mathsf{R}$, find $[m]_\lambda$, where $[m]_\lambda$ denotes the λ least significant coefficients of m.*

As above, for any adversary $\mathcal{A}$, we denote by $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_\lambda}(\mathcal{A})$ its success for breaking the *NTRU λ-Partial-Domain Inversion Problem*, where

$$\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_\lambda}(\mathcal{A}) = \Pr\left[\begin{array}{l}(\mathsf{h},(\mathsf{f},\mathsf{g})) \leftarrow \mathcal{K}(1^k), \mathsf{m} \in \mathcal{S}_\mathsf{M}, \mathsf{r} \in \mathcal{S}_\mathsf{R}, \\ \mathsf{e} = \mathsf{m} + p\mathsf{r} * \mathsf{h} : \mathcal{A}(\mathsf{e},\mathsf{h}) = [\mathsf{m}]_\lambda\end{array}\right].$$

Note that the NTRU encryption primitive is malleable with respect to circular shifts: $\mathcal{E}_{\mathsf{pk}}(X * \mathsf{m}; X * \mathsf{r}) = X * \mathcal{E}_{\mathsf{pk}}(\mathsf{m}; \mathsf{r})$. This implies that any λ-consecutive-coefficient search problem is equivalent to the above *NTRU λ-Partial-Domain Inversion Problem*. Because of the specific encodings used by NTRU, it will also be useful to consider the difficulty of obtaining some information bits of the pre-image only:

Definition 3 (The NTRU ℓ-Partial-Information Inversion Problem). *For a given security parameter k, which specifies N, p, q and several other constraints, as well as $(h,(f,g)) \leftarrow \mathcal{K}(1^k)$ and $e = m + pr * h$, where $m \in \mathcal{S}_\mathsf{M}$ and $r \in \mathcal{S}_\mathsf{R}$, find ℓ bits of information about m.*

As above, we denote by $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pi-ow}_\ell}(\mathcal{A})$ its success for breaking the *NTRU ℓ-Partial-Information Inversion Problem*, where

$$\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pi-ow}_\ell}(\mathcal{A}) = \Pr\left[\begin{array}{l}(\mathsf{h},(\mathsf{f},\mathsf{g})) \leftarrow \mathcal{K}(1^k), \mathsf{m} \in \mathcal{S}_\mathsf{M}, \mathsf{r} \in \mathcal{S}_\mathsf{R}, \\ \mathsf{e} = \mathsf{m} + p\mathsf{r} * \mathsf{h}, (f, y) \leftarrow \mathcal{A}(\mathsf{e},\mathsf{h}) : y = f_\ell(\mathsf{m})\end{array}\right].$$

In the above definition, the adversary $\mathcal{A}$ outputs a (computable) bijective function $f : \mathcal{S}_\mathsf{M} \to \{0,1\}^{\mathsf{mLen}}$ (where mLen denotes the bit-length of an optimal encoding for polynomials in $\mathcal{S}_\mathsf{M}$) and $y \in \{0,1\}^\ell$. Furthermore, f_ℓ denotes the truncation of f to its ℓ least significant bits.

More generally, we denote by $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{ow}}(t)$, $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_\lambda}(t)$ and $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pi-ow}_\ell}(t)$, the maximal success among all the adversaries with a running time bounded by t.

Relations among computational assumptions. The definition of our assumptions implies that for all t and λ

$$\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{ow}}(t) \leq \mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_\lambda}(t) \leq \mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pi-ow}_\ell}(t),$$

where ℓ is the bit-length of an optimal encoding for λ coefficients. And in the specific case $\lambda = N$, all the inequalities become equalities:

$$\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{ow}}(t) = \mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_N}(t) = \mathsf{Adv}_{\mathsf{ntru}}^{\mathsf{pi-ow}_{\mathsf{mLen}}}(t).$$

Remark that if $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_\lambda}(t) = 1$ for some (t, λ), then $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{ow}}(\lceil N/\lambda \rceil t) = 1$, where $\lceil x \rceil$ denotes the smallest integer larger than x. This is because the malleability of the NTRU encryption primitive allows to reduce any instance of the NTRU inversion problem to $\lceil N/\lambda \rceil$ instances of the NTRU λ-partial-domain inversion problem, using appropriate shifts of e. Note however that the multiplication by X does not provide a random self-reduction: the previous reduction implies nothing on $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{ow}}(\lceil N/\lambda \rceil t)$ if $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_\lambda}(t) < 1$. In fact, since no random self-reducibility property is known for NTRU, it is likely that if $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_\lambda}(t) < 1$, then the NTRU λ-partial-domain inversion problem is strictly harder than the NTRU inversion problem. Besides, the lack of random self-reducibility also suggests that the NTRU ℓ-partial-information inversion problem is strictly harder than the NTRU λ-partial-domain inversion problem.

In the following, we assume that all the above problems become intractable for a sufficiently large security parameter k, and for sufficiently large enough ℓ and λ (since random guesses lead to $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pd-ow}_\lambda}(1) = 1/p^\lambda$ and $\mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pi-ow}_\ell}(1) = 1/2^\ell$).

3.5 The Security of the NTRU Primitive

The best attack known against NTRU is based on lattice reduction, but this does not imply that lattice reduction is necessary to break NTRU. See [3, 13, 24] for further information. Based on numerous experiments, the authors of NTRU claimed in [13] that all known lattice-based attacks are exponential in N, and therefore suggested relatively small values of N. The parameter N must be prime, otherwise the lattice attacks can be improved due to non-trivial factors of $X^N - 1$ (see [10]). Because the key-size of NTRU is only $\mathcal{O}(N \log q)$, one can allow reasonably high lattice dimensions, while all other known knapsack-based or lattice-based cryptosystems have a key-size which is at least quadratic in the security parameter.

NTRU, like most public-key cryptosystems, should not be directly used as originally described. For instance, NTRU is easily seen to be semantically insecure, as $\mathsf{e}(1) \equiv \mathsf{m}(1) \pmod{q}$ because $\mathsf{r}(1) = 0$. This yields a significant bias for any adversary to distinguish between two possible plaintexts which one has been encrypted. In fact, although the NTRU cryptosystem is probabilistic, there is a public plaintext-checking oracle: one can easily check whether a given message m corresponds to a ciphertext e, which implies that any security level in the CPA scenario holds in the PCA one as well. This is because the shape of r is publicly verifiable and the public key h is "pseudo-invertible" modulo q with overwhelming probability. More precisely, one can compute from h a polynomial $\mathsf{H} \in \mathcal{P}$ such that for any polynomial $\mathsf{s} \in \mathcal{P}$ such that $\mathsf{s}(1) \equiv 0 \pmod{q}$:

$$\mathsf{h} * \mathsf{H} * \mathsf{s} \equiv \mathsf{s} \pmod{q}.$$

Then, if $\mathsf{e} = \mathsf{m} + p\mathsf{r} * \mathsf{h} \pmod{q}$ is a ciphertext of m, we have since $\mathsf{r}(1) \equiv 0 \pmod{q}$:

$$p\mathsf{r} \equiv (\mathsf{e} - \mathsf{m}) * \mathsf{H} \pmod{q}.$$

This allows to retrieve the random polynomial r modulo q, whose shape is publicly verifiable. By injectivity of encryption, we thus obtain a public plaintext-checking oracle.

We briefly explain this "pseudo-inversion" since we have not found any reference. Because N is an odd prime and q is a power of two, the ring $\mathcal{P}_q = \mathbb{Z}_q[X]/(X^N - 1)$ is isomorphic to $\mathcal{P}_1 \times \mathcal{P}_2$ where $\mathcal{P}_1 = \mathbb{Z}_q[X]/(X - 1)$ and

$\mathcal{P}_2 = \mathbb{Z}_q[X]/(X^{N-1} + X^{N-2} + \cdots + 1)$. Denote by ϕ_1 and ϕ_2 respectively the reduction modulo $X - 1$ and $X^{N-1} + X^{N-2} + \cdots + 1$. Of course, ϕ_1 is simply the evaluation at 1. Since $\mathsf{h}(1) \equiv 0 \pmod q$, $\phi_1(\mathsf{h})$ is not invertible in $\mathcal{P}_1$, and therefore h is not invertible in $\mathcal{P}_q$. However, $\phi_2(\mathsf{h})$ is very likely to be invertible in $\mathcal{P}_2$ (the proportion of invertible elements can be computed as in [14]). And its inverse can easily be computed: for instance, by computing the inverse in $\mathbb{F}_2[X]/(X^{N-1} + X^{N-2} + \cdots + 1)$, and then lifting it modulo q. Eventually, one can derive an appropriate polynomial $\mathsf{H} \in \mathcal{P}$ from this inverse.

As previously mentioned, NTRU is also easily malleable using multiplications by X: $\mathcal{E}_{\mathsf{pk}}(X * \mathsf{m}; X * \mathsf{r}) = X * \mathcal{E}_{\mathsf{pk}}(\mathsf{m};\mathsf{r})$. Jaulmes and Joux [19] further presented simple chosen-ciphertext attacks that can recover the private key. Curiously, one of these attacks could be applied to a specific padding proposed by NTRU Cryptosystems to avoid reaction attacks [17]. This stressed the need of an appropriate padding scheme to obtain high levels of security against a vast class of attacks, assuming the NTRU one-way function $\mathcal{E}_{\mathsf{pk}}(\mathsf{m};\mathsf{r})$ is hard to invert (which, by definition, is equivalent to asking that the NTRU inversion problem is hard).

4 The NTRU Cryptosystems

For clarity, in the following, we consider the encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$, which is the same as $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, up to the two public encodings:

$$\mathcal{M} : \{0,1\}^{\mathsf{mLen}} \longrightarrow \mathcal{S}_{\mathsf{M}} \text{ and } \mathcal{R} : \{0,1\}^{\mathsf{rLen}} \longrightarrow \mathcal{S}_{\mathsf{R}}.$$

$$\Pi' \begin{cases} \mathcal{K}'(1^k) = \mathcal{K}(1^k) & = (\mathsf{pk} = \mathsf{h}, \mathsf{sk} = (\mathsf{f}, \mathsf{g})), \\ \mathcal{E}'_{\mathsf{pk}}(m; r) = \mathcal{E}_{\mathsf{pk}}(\mathcal{M}(m); \mathcal{R}(r)) & = \mathcal{M}(m) + p\mathcal{R}(r) * \mathsf{h} \pmod q, \\ \mathcal{D}'_{\mathsf{sk}}(\mathsf{e}) = \mathcal{M}^{-1}(\mathcal{D}_{\mathsf{sk}}(\mathsf{e})) & = \mathcal{M}^{-1}((\mathsf{e} * \mathsf{f} \pmod p)) * \mathsf{f}_p). \end{cases}$$

Because of the encodings, without any assumption, recovering the bit-string m is as hard as recovering the polynomial $\mathsf{m} = \mathcal{M}(m)$. However, recovering ℓ bits of m only provides ℓ bits of information about the polynomial $\mathsf{m} = \mathcal{M}(m)$, which is why we introduced the NTRU ℓ-partial-information inversion problem. From these remarks:

$$\mathsf{Succ}^{\mathsf{ow-cpa}}_{\Pi'}(t) = \mathsf{Succ}^{\mathsf{ow-pca}}_{\Pi'}(t) = \mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(t).$$

4.1 NTRU Paddings

Following the publication of [19], NTRU proposed several padding schemes in [16, 15] to protect NTRU against adaptive chosen-ciphertext attacks. We note that at the time of the writing of [16], several generic transformations were known to make NTRU IND-CCA2–secure in the random oracle model [2, 7, 8, 26]. However, a few complications arise as the NTRU one-way function cannot be assumed IND-CPA.

All NTRU paddings require a hash function $H : \{0,1\}^{\mathsf{mLen}} \rightarrow \{0,1\}^{\mathsf{rLen}}$, and possibly F and G, whose output size will be made explicit later.

Let M be the original plaintext represented by a k_1-bit string. For each encryption, one generates a random string R, whose bit-length k_2 is between 40 and 80 according to [16, page 2]. However, $k_1 + k_2 \leq \mathsf{mLen}$. Let $\|$ denote bit-string concatenation. The paddings proposed by NTRU are as follows.

Padding I. The first padding is proposed in [16, page 3]. The ciphertext of M with random R is:

$$\mathcal{E}^1_{\mathsf{pk}}(M; R) = \mathcal{E}'_{\mathsf{pk}}(M \,\|\, R; H(M \,\|\, R)).$$

We denote by Π^1 the corresponding encryption scheme.

Padding II. The second padding is proposed in [16, page 3]. The ciphertext is:

$$\mathcal{E}^2_{\mathsf{pk}}(M; R) = \mathcal{E}'_{\mathsf{pk}}((F(R) \oplus M) \,\|\, R; H(M \,\|\, R)),$$

where F is a hash function that maps $\{0,1\}^{k_2} \to \{0,1\}^{k_1}$, and $\oplus$ denotes bitwise exclusive or. We denote by Π^2 the corresponding encryption scheme.

Padding III. The third padding is proposed in [15, page 3], and not in [16]. This is rather curious, as [15] actually suggests the reading of [16] for further details. It is this padding and not the previous ones which is being considered in the CEES standards [6].

This padding first applies an all-or-nothing transformation (OAEP [2]) on the concatenation $M \,\|\, R$. More precisely, it splits each of M and R into equal size pieces $M = \overline{M} \,\|\, \underline{M}$ and $R = \overline{R} \,\|\, \underline{R}$. It then uses two hash functions F and G that map $\{0,1\}^{k_1/2+k_2/2}$ into itself, to compute:

$$m_1 = (\overline{M} \,\|\, \overline{R}) \oplus F(\underline{M} \,\|\, \underline{R}) \text{ and } m_2 = (\underline{M} \,\|\, \underline{R}) \oplus G(m_1).$$

The ciphertext is then:

$$\mathcal{E}^3_{\mathsf{pk}}(M; R) = \mathcal{E}'_{\mathsf{pk}}(m_1 \,\|\, m_2; H(M \,\|\, R)).$$

We denote by Π^3 the corresponding encryption scheme.

4.2 Security Analyses

First of all, one may note that because of the random polynomial r that is generated from $H(M \,\|\, R)$, nobody can generate a valid ciphertext without knowing both the plaintext M and the random R, except with negligible probability. Indeed, for a given ciphertext e, at most one r is acceptable. Without having asked $H(M \,\|\, R)$, the probability for $\mathcal{R}(H(M \,\|\, R))$ to be equal to r is less than $1/2^N$. As a consequence, any security notion satisfied in the CPA scenario is satisfied in the $\mathsf{CCA2}$-scenario. The latter scenario may increase the success probability of an adversary by at most $q_D/2^N$, where q_D is the number of queries asked to the decryption oracle. With a proper bookkeeping, the cost of the simulation increases by $q_H T_{\mathcal{E}}$, at most, where q_H is the number of queries asked to the random oracle H, and $T_{\mathcal{E}}$ the time required for one encryption: for $i = 1, 2, 3$,

$$\mathsf{Adv}_{\Pi^i}^{\mathsf{ind-cca2}}(t) \leq \mathsf{Adv}_{\Pi^i}^{\mathsf{ind-cpa}}(t + q_H T_{\mathcal{E}}) + \frac{2q_D}{2^N}.$$

Therefore, in the following, we focus on the *chosen-plaintext* attacks only.

Analysis of the First NTRU Padding. The first padding is exactly based on the first Fujisaki-Okamoto conversion [8], which requires the primitive to be $\mathsf{IND\text{-}CPA}$, which is not the case of Π'! However, the one-wayness $\mathsf{OW\text{-}CPA}$ of Π^1 already relies on a stronger assumption than the hardness of the NTRU inversion problem: the hardness of the NTRU k_1-partial-information inversion problem.

More worryingly, contrarily to the claims of [16], the scheme Π^1 is not *semantically secure* ($\mathsf{IND\text{-}CPA}$): let us consider the following adversary which chooses $M_0 = 0^{k_1}$ and $M_1 = 1^{k_1}$. R is unknown, but whatever it is, if the encoding $\mathcal{M}$ is such that R has an impact on at most k_2 coefficients: $\mathcal{M}(M_0 \,\|\, R)(1) \leq k_2$, $\mathcal{M}(M_1 \,\|\, R)(1) \geq k_1$. As already remarked, this value mod q is given by $\mathsf{e}(1)$, which helps to distinguish which message has been encrypted, with advantage 1. Optimizations [15] (such as $\mathsf{r}(1) \neq 0$ but still a public constant) may slightly worsen the advantage, but the advantage is still significant (more than 1/2).

Analysis of the Second NTRU Padding. The second padding is more surprising. Even if it provides *one-wayness* under the sole NTRU assumption, *semantic security* still requires a stronger assumption: the hardness of the NTRU k_2-partial-information inversion problem.

First, to get any information about the bit b such that the message M_b is encrypted, any adversary has to ask either $F(R)$ or $H(M_i \,\|\, R)$. If one denotes by Ask such an event, one obtains: $\mathsf{Adv}_{\Pi^2}^{\mathsf{ind-cpa}}(t) \leq 2\Pr[\mathsf{Ask}]$. In the worst case, by randomly picking one candidate, one can extract R, and thus k_2 bits of information about the polynomial m:

$$\mathsf{Adv}_{\Pi^2}^{\mathsf{ind-cpa}}(t) \leq 2(q_F + q_H) \times \mathsf{Succ}_{\mathsf{ntru}}^{\mathsf{pi-ow}_{k_2}}(t).$$

From a OW-adversary $\mathcal{A}$, which runs within a time bound t, one can get more, whereas the simulations of F and H may be inconsistent. Indeed, the challenge ciphertext e defines R uniquely, but M is a random variable later defined by $F(R)$, $M = F(R) \oplus \mathcal{M}^{-1}(\mathsf{m})$, and then $H(M \,\|\, R) = \mathcal{R}^{-1}(\mathsf{r})$. The latter may not be correctly answered. If $H(M \,\|\, R)$ is not asked, the view of the adversary $\mathcal{A}$ is perfect, and the output M gives $\mathsf{m} = \mathcal{M}(\rho \oplus M)$, where $F(R) = \rho$. But if $F(R)$ has not been asked, the success of the adversary is upper-bounded by $1/2^{k_1}$. If $H(M \,\|\, R)$ has been asked, one guesses one pair $(M \,\|\, R)$ among the queries asked to H, and performs the same as above. There are $q_F + q_H$ possibilities for R, in the first case, or q_H possibilities for $M \,\|\, R$ in the second one. The good candidate can easily be checked. As a result, one gets

$$\mathsf{Succ}^{\mathsf{ow-cpa}}_{\Pi^2}(t) \leq \frac{1}{2^{k_1}} + \mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(t + (q_F + 2q_H)T_{\mathcal{E}}).$$

Analysis of the Third NTRU padding. The third padding makes an incorrect use of the All-Or Nothing Transform, and therefore, the achieved security level is not as high as one might have expected.

Let us first consider an adversary $\mathcal{A}$ against semantic security, which tries to guess between M_0 and M_1 which one has been encrypted, within a time bound t. It is clear that without having asked $H(M_i \,\|\, R)$, $F(\underline{M}_i \,\|\, \underline{R})$, or $G(m_1)$ (which events are denoted AskH, AskF and AskG respectively), the plaintext M and the random R are totally impredictable. However, the probability to ask $F(\underline{M}_i \,\|\, \underline{R})$, without having asked $G(m_1)$, is less than $2q_F/2^{k_2/2}$. Similarly, the probability to ask $H(M_i \,\|\, R)$, without having asked either $G(m_1)$ or $F(\underline{M}_i \,\|\, \underline{R})$, is less than $2q_H/2^{k_2}$. Therefore, m_1 has been asked to G, except with a small probability:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind-cpa}}_{\Pi^3}(\mathcal{A}) &\leq 2\Pr[\mathsf{AskF} \vee \mathsf{AskG} \vee \mathsf{AskH}] \\ &\leq 2\Pr[\mathsf{AskG}] + 2\Pr[\mathsf{AskF} \mid \neg\mathsf{AskG}] + 2\Pr[\mathsf{AskH} \mid \neg\mathsf{AskG} \wedge \neg\mathsf{AskF}] \\ &\leq 2\Pr[\mathsf{AskG}] + \frac{4q_F}{2^{k_2/2}} + \frac{4q_H}{2^{k_2}} = 2\Pr[\mathsf{AskG}] + 4 \times \frac{2^{k_2/2}q_F + q_H}{2^{k_2}}.\end{aligned}$$

If the event AskG occurs, by correctly guessing the query asked to G, one gets m_1:

$$\frac{1}{q_G}\Pr[\mathsf{AskG}] \leq \mathsf{Succ}^{\mathsf{pi-ow}_{\mathsf{mLen}/2}}_{\mathsf{ntru}}(t).$$

As a consequence,

$$\mathsf{Adv}^{\mathsf{ind-cpa}}_{\Pi^3}(t) \leq 2q_G \times \mathsf{Succ}^{\mathsf{pi-ow}_{\mathsf{mLen}/2}}_{\mathsf{ntru}}(t) + \frac{2^{k_2/2}q_F + q_H}{2^{k_2-2}}.$$

From a OW-adversary $\mathcal{A}$, which outputs the whole plaintext M, one can get more: indeed, $\mathcal{A}$ has to ask $F(\underline{M} \,\|\, \underline{R})$ and $G(m_1)$, or $H(M \,\|\, R)$ to know M, otherwise $\overline{M}$ is totally impredictable: only m_1 and m_2 are determined by e. But $\underline{M} \,\|\, \underline{R}$ is a random variable defined later by $\underline{M} \,\|\, \underline{R} = m_2 \oplus G(m_1)$, and $\overline{M} \,\|\, \overline{R}$ is a random variable defined by $\overline{M} \,\|\, \overline{R} = m_1 \oplus F(\underline{M} \,\|\, \underline{R})$. Furthermore, the probability to ask $H(M \,\|\, R)$, without having asked $F(\underline{M} \,\|\, \underline{R})$ and $G(m_1)$ is very low, since half of the bits are still impredictable:

$$\begin{aligned}\mathsf{Succ}^{\mathsf{ow-cpa}}_{\Pi^3}(\mathcal{A}) &= \Pr[M \leftarrow \mathcal{A}(\mathsf{e}) \wedge ((\mathsf{AskF} \wedge \mathsf{AskG}) \vee \mathsf{AskH})] \\ &\quad + \Pr[M \leftarrow \mathcal{A}(\mathsf{e}) \wedge \neg((\mathsf{AskF} \wedge \mathsf{AskG}) \vee \mathsf{AskH})] \\ &\leq \Pr[M \leftarrow \mathcal{A}(\mathsf{e}) \wedge \mathsf{AskF} \wedge \mathsf{AskG}] \\ &\quad + \Pr[\mathsf{AskH} \wedge \neg(\mathsf{AskF} \wedge \mathsf{AskG})] + \frac{1}{2^{k_1}} \\ &\leq \Pr[M \leftarrow \mathcal{A}(\mathsf{e}) \wedge \mathsf{AskF} \wedge \mathsf{AskG}] + \frac{q_H}{2^{k_1/2+k_2/2}} + \frac{1}{2^{k_1}}\end{aligned}$$

With the solution $M = \overline{M} \,\|\, \underline{M}$, the query $\underline{M} \,\|\, \underline{R}$ and its answer by F, as well as the query m_1 and its answer by G, one gets m_2, which is possible to check. However, the adversary may detect the simulation, since some inconsistency in the simulation of H may occur, if $M \,\|\, R$ is asked to H. But then one can fully invert the problem, by trying all the candidates in the list of queries to H. Finally, one first checks the q_H possibilities for $M \,\|\, R$, and then tries all the possible combinations between m_1 and $\underline{M} \,\|\, \underline{R}$, to get m_2:

$$\mathsf{Succ}^{\mathsf{ow-cpa}}_{\Pi^3}(t) \leq \mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(t + (q_H + q_F q_G)T_{\mathcal{E}}) + \frac{q_H 2^{(k_1-k_2)/2} + 1}{2^{k_1}}.$$

Discussion. One should remark that k_2 is a crucial security parameter. With too small of a value, some security results become meaningless, namely the semantic security of the second and the third paddings. However, one can see that splitting the k_2-bit random value R in two parts $\overline{R}$ and $\underline{R}$, which are used independently, is a very bad idea: the provable security level of the third construction is less than $1/2^{k_2/2}$. The latter is thus at most 2^{-20}, or 2^{-40}, according to [16, page 2]. Thus, the security proofs we obtain are rather inefficient for the parameters suggested by [16]. Our proofs may however not be tight since we are unaware of any attack achieving the previous security bounds. Nevertheless, the lack of security proofs meaningful for the recommended parameters suggests to look at different paddings.

5 Suggestions and Comparisons

We showed that none of the three suggested paddings provides the maximal security level, that is, IND-CCA2 under the sole NTRU inversion problem. However, some constructions do exist: a better OAEP-based construction or REACT [25], which we will compare later.

5.1 Suggestions

An OAEP-based Scheme. The first suggestion is a variant of the third padding, using two more hash functions

$$F : \{0,1\}^{k_1} \to \{0,1\}^{k_2} \text{ and } G : \{0,1\}^{k_2} \to \{0,1\}^{k_1}.$$

One first computes $s = M \oplus G(R)$ and $t = R \oplus F(s)$. The ciphertext consists of $\mathcal{E}'_{\mathsf{pk}}(s \,\|\, t; H(M \,\|\, R))$. The OAEP construction provides semantic security, while the H function strengthens it to chosen-ciphertext security. An usual argument (see the full version of the paper) shows that:

$$\mathsf{Adv}^{\mathsf{ind-cca2}}_{\mathsf{oaep'}}(t) \leq 2\mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(t + (q_H + q_F q_G)T_{\mathcal{E}}) + \frac{2q_D}{2^N} + \frac{4q_H}{2^{k_1}} + \frac{2q_G}{2^{k_2}}.$$

However, one can thus see that, as in the original OAEP construction with partial-domain one-way permutations [2, 9], the reduction is quadratic in the number of queries to the hash functions F and G.

NTRU-REACT. Thanks to the OW-PCA–security level of the NTRU primitive, one can use the REACT construction. The straightforward application uses two hash functions:

$$G : \{0,1\}^N \to \{0,1\}^\ell \text{ and } H : \{0,1\}^\star \to \{0,1\}^{k_2}.$$

On input a message $M \in \{0,1\}^\ell$, a random $R \in \{0,1\}^{\mathsf{mLen}}$ and another random $R' \in \{0,1\}^N$, one computes $a = \mathcal{E}'_{\mathsf{pk}}(R; R')$, $b = M \oplus G(R)$ and $c = H(a, b, M, R)$. The ciphertext is the triplet (a, b, c).

The semantic security is clear, since the adversary has no advantage without having asked $G(R)$ or $H(a, b, M_i, R)$. Therefore R, and thus m, can be recovered from the queries asked to G or H:

$$\mathsf{Adv}^{\mathsf{ind-cpa}}_{\mathsf{react}}(t) \leq 2\mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(t + (q_G + q_H)T_{\mathcal{E}}).$$

With chosen-ciphertext attacks, the adversary cannot produce a valid ciphertext without having asked for $H(a, b, M, R)$, except with probability $1/2^{k_2}$. With proper bookkeeping, this does not increase the cost:

$$\mathsf{Adv}^{\mathsf{ind-cca2}}_{\mathsf{react}}(t) \leq 2\mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(t + (q_G + q_H)T_{\mathcal{E}}) + \frac{2q_D}{2^{k_2}}.$$

Improved NTRU-REACT. Interestingly, the specific properties of NTRU can be used to improve the above construction. Namely, one can reduce the size and the number of random bits. It requires two hash functions:

$$G : \{0,1\}^{\mathsf{mLen}} \to \{0,1\}^\ell \text{ and } H : \{0,1\}^\star \to \{0,1\}^N.$$

Like the original construction of REACT, it can use any symmetric encryption scheme (E, D), with an ℓ-bit key. On input a message M and a random element $R \in \{0,1\}^{\mathsf{mLen}}$, one computes $K = G(R)$, $b = \mathsf{E}_K(M)$ and $R' = H(R, b)$. Then $a = \mathcal{E}'_{\mathsf{pk}}(R; R')$, and the ciphertext consists of the pair $a \,\|\, b$. One can prove that with this improved scheme and the one-time pad:

$$\mathsf{Adv}^{\mathsf{ind-cca2}}_{\mathsf{react'}}(t) \leq 2\mathsf{Succ}^{\mathsf{ow}}_{\mathsf{ntru}}(t + (q_G + q_H)T_{\mathcal{E}}) + \frac{2q_D}{2^N}.$$

A high rate can be achieved thanks to the hybrid construction. However, one would need to compare the efficiency of the block cipher and the NTRU primitive.

5.2 Comparison of NTRU Cryptosystems

We now compare the efficiency and the security level of all the above constructions. Contrarily to previous complexity analyses, we need to consider the cost of the generation of random bits as well as the cost of hashings. Indeed, this is the only difference between each scheme, since all of them just need to apply once the encryption and decryption primitives in the encryption and decryption algorithms respectively. In the figure given below, mLen and rLen are the bit-length of the message and random inputs for the NTRU primitive, and cLen is the bit-length of the output; k is a security parameter. The columns #rand, Hin and Hout indicate the number of required random bits, the number of bits as input of hash functions and the number of output bits respectively.

With classical parameters, where N is the most crucial data, chosen among 167, 251, 347 and 503, $\mathsf{mLen} = N \log p = \lceil 1.585N \rceil$, $\mathsf{rLen} = N$, $\mathsf{cLen} = 7N$ and k is between 40 and 80.

Schemes	$\|M\|$	$\|C\|$	IND-CCA2	#rand	Hin	Hout
Π^1	mLen $- k$	cLen	NO	k	mLen	rLen
Π^2	mLen $- k$	cLen	PI-OW	k	mLen $+ k$	rLen + mLen $- k$
Π^3	mLen	cLen	PI-OW	k	2mLen	rLen + mLen
OAEP′	mLen $- k$	cLen	OW	k	2mLen	rLen + mLen
REACT′	mLen	mLen + cLen	OW	mLen	3mLen	rLen + mLen

Acknowledgements. We would like to thank Mike Szydlo for helpful discussions.

References

1. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Proc. of Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, 1998.
2. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption. In *Proc. of Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, 1995.
3. D. Coppersmith and A. Shamir. Lattice Attacks on NTRU. In *Proc. of Eurocrypt '97*, LNCS 1233. Springer-Verlag, 1997.
4. NTRU Cryptosystems. Technical reports available at `http://www.ntru.com`. 2002.
5. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
6. Consortium for Efficient Embedded Security. Efficient embedded security standards #1: Implementation aspects of NTRU and NSS.
Draft Version 3.0 available at `http://www.ceesstandards.org`, July 2001.
7. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Proc. of Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, 1999.
8. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. *IEICE Trans. Fundamentals of Electronics, Comunications and Computer Sciences*, E83-A(1), 2000. Special issue on cryptography and information security.
9. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA–OAEP is Secure under the RSA Assumption. In *Proc. of Crypto '01*, LNCS 2139, pages 260–274. Springer-Verlag, 2001.
10. C. Gentry. Key Recovery and Message Attacks on NTRU-Composite. In *Proc. of Eurocrypt '01*, LNCS 2045, pages 182–194. Springer-Verlag, 2001.

11. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
12. C. Hall, I. Goldberg, and B. Schneier. Reaction Attacks against Several Public-Key Cryptosystems. In *Proc. of ICICS '99*, LNCS, pages 2–12. Springer-Verlag, 1999.
13. J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: a Ring based Public Key Cryptosystem. In *Proc. of ANTS III*, LNCS 1423, pages 267–288. Springer-Verlag, 1998. First presented at the rump session of Crypto '96.
14. J. Hoffstein and J. H. Silverman. Invertibility in truncated polynomial rings. Technical report, NTRU Cryptosystems, October 1998. Report #009, version 1, available at [4].
15. J. Hoffstein and J. H. Silverman. Optimizations for NTRU. In *Public-key Cryptography and Computational Number Theory.* DeGruyter, 2000. To appear, available at [4].
16. J. Hoffstein and J. H. Silverman. Protecting NTRU against chosen ciphertext and reaction attacks. Technical report, NTRU Cryptosystems, June 2000. Report #016, version 1, available at [4].
17. J. Hoffstein and J. H. Silverman. Reaction attacks against the NTRU public key cryptosystem. Technical report, NTRU Cryptosystems, June 2000. Report #015, version 2, available at [4].
18. IEEE Standard 1363. Standard specifications for public key cryptography. IEEE. Available from `http://grouper.ieee.org/groups/1363`, August 2000.
19. E. Jaulmes and A. Joux. A Chosen Ciphertext Attack on NTRU. In *Proc. of Crypto '00*, LNCS 1880. Springer-Verlag, 2000.
20. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Ann.*, 261:513–534, 1982.
21. A. May and J. H. Silverman. Dimension Reduction Methods for Convolution Modular Lattices. In *Proc. of CALC '01*, LNCS 2146. Springer-Verlag, 2001.
22. D. Micciancio. Improving Lattice-based Cryptosystems using the Hermite Normal Form. In *Proc. of CALC '01*, LNCS 2146. Springer-Verlag, 2001.
23. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, 1990.
24. P. Q. Nguyen and J. Stern. The Two Faces of Lattices in Cryptology. In *Proc. of CALC '01*, LNCS 2146. Springer-Verlag, 2001.
25. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *Proc. of CT-RSA '01*, LNCS 2020, pages 159–175. Springer-Verlag, 2001.
26. D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem. In *Proc. of PKC '00*, LNCS 1751, pages 129–146. Springer-Verlag, 2000.
27. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Proc. of Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, 1992.
28. C. P. Schnorr. A Hierarchy of Polynomial Lattice Basis Reduction Algorithms. *Theoretical Computer Science*, 53:201–224, 1987.