On Achieving Chosen Ciphertext Security with Decryption Errors

Yang Cui[1], Kazukuni Kobara[2], and Hideki Imai[2]

[1] Dept. of Information & Communication Engineering, University of Tokyo
cuiyang@imailab.iis.u-tokyo.ac.jp
[2] Institute of Industrial Science, University of Tokyo,
Komaba 4-6-1, Meguro-Ku, Tokyo, 153-8505, Japan
{kobara, imai}@iis.u-tokyo.ac.jp
http://imailab-www.iis.u-tokyo.ac.jp/imailab.html

Abstract. Perfect decryption has been always assumed in the research of public key encryption, however, this is not true all the time. For some public key encryption primitives, like NTRU [9] or Ajtai-Dwork [1], the decryption process may not obtain the corresponding message even the encryption and decryption are run correctly. Furthermore, such a kind of decryption errors will lead to some dangerous attacks against the underlying primitive. Another interesting point is that, those primitives are not based on the factoring, nor the discrete log problem which are subject to the Shor's algorithm [18] with quantum computers. This kind of primitives may be promising in the post-quantum cryptography. Therefore, the decryption errors deserve much attention and should be coped with carefully.

In this paper, our main technique is not to use any error-correcting codes to *eliminate* the errors, but to use some padding (transform) to *hide* "bad" errors from attacker's control. We 1) efficiently enhance these error-prone public key encryption primitives to the chosen ciphertext security, even in the presence of the decryption errors, and 2) show that the solution is more generic, rather than some specific padding methods previously presented, to thwart the decryption errors based attacks successfully.

1 Introduction

Public key encryption (PKE) is a crucial building block in cryptography, widely used in many security protocols and schemes. Whilst various PKEs are proposed to fulfill with the requirement in different scenarios, one property of PKE is always assumed, which is the perfect decryption. It means that any validly encrypted ciphertext will lead to the same message corresponding to the ciphertext for certainty. However, there exists a family of PKE that has good performance in the implementation, but fails to have perfect decryption sometime, such as NTRU [9] and Ajtai-Dwork [1] etc.

Even though their decryption errors do not occur often, they do have been affected greatly. Indeed the decryption errors we care about are not only possible

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 173–182, 2006.

to reduce the efficiency of PKE, but might also give additional useful information to potential attackers, thus lead to a fatal attack, such as secret key exposure [16].

Given a PKE (a randomly generated public and secret key pair), for some message and randomness pair, the encryption algorithm may lead to a mapping which is not injective. This would be inevitable if some specific message pairs are chosen, and further it gives the opportunity for the attacker to know the truth - some ciphertexts are corresponding to decryption error message, which is never desired to be known by the attacker with strong power, such as adaptively chosen ciphertext attacker [17].

Although perfect decryption has not been achieved, this kind of PKE is so meaningful after the Shor's factoring algorithm [18]. Its significance lies in that they are not based on the common number-theoretic problems of factoring or discrete log, like RSA or ElGamal, but on the lattice problem which is believed hard to be solved even that the quantum computer is built in the future. Additionally, since the fast implementation of them can be compared with RSA, this family may be a promising replacement of the commonly used PKE, if it could be made immune to the decryption errors.

1.1 Related Work

There are some related work in this context, Goldreich et al [7] proposed a solution to the decryption problem of Ajtai-Dwork [1], but failed to make the scheme secure [12]. Later, Dwork et al [5] generalized the theoretical solution to solve any infrequent decryption errors, using several totally impractical techniques as parallel repetition, hard core bit and direct product. By these only theoretically meaningful techniques, the error probability could be found with only a tiny probability, i.e. the attack using decryption errors is made impossible to run efficiently.

Although some efficient work by Howgrave-Graham et al. [10] provided an exclusive use padding scheme for NTRU, called NAEP, to enhance the security of NTRU even in the presence of decryption errors, it was especially designed and thus not useful for any other PKE. As NAEP did, in the random oracle model [2] [1], an efficient solution in [5] also used a padding to enhance the security practically. However, this transform appears a little complex and not good at bandwidth overhead, having several padding schemes together, like Fujisaki-Okamoto [6] combined with PSS-E [4], where a symmetric encryption is also required.

1.2 Main Contributions

From a practical viewpoint, we expect the padding methods be generic so that it could deal with many other PKEs, no matter whether there exist decryption errors or not. In addition, the efficiency is also important, otherwise it will be too expensive for this kind of error-prone encryptions.

[1] A useful tool to design and analyze the cryptosystem, is widely used in both theory and practice.

Our main method is not to correct the errors, but to hide the errors from the attacker's control. We point out that actually, some existed generic padding may help deterministic primitive immunize the attack from the decryption errors, such as 3-round OAEP [14]. And we provide a new variant of it to cope with probabilistic primitive as well. Note that both of them are generic to adapt to many other PKEs, and very efficient, especially in bandwidth.

Next, we will first explain the security notions and the attack by Proos, then show that some error-prone PKEs could be enhanced to chosen ciphertext security provably when decryption failures occur.

2 Notions and Notations

In the following paper, we define $\mathcal{M}$, $\mathcal{R}$ as the message and randomness space respectively, and $\mathcal{C}$ is the ciphertext space, where $\mathcal{C} = \mathcal{M} \times \mathcal{R}$. $\Pr[\texttt{operation}|\cdot]$ represents the probability of event "$\cdot$" under the corresponding operation. And we say that $\texttt{negl}(k)$ is negligible, if for any constant c, there exists $k_0 \in \mathbf{N}$, s.t. $\texttt{negl}(k) < (1/k)^c$ for any $k > k_0$.

2.1 Public Key Encryption

Definition 1. *Public key encryption Π is defined by a triple of algorithms, ($\mathcal{K}$, $\mathcal{E}$, $\mathcal{D}$):*

- *the key generation algorithm $\mathcal{K}$: on a secret input 1^k ($k \in \mathbf{N}$), in polynomial time in k, it produces a pair of keys (pk, sk), public and secret known respectively.*
- *the encryption algorithm $\mathcal{E}$: on input of message $m \in \mathcal{M}$ and public key pk, the algorithm $\mathcal{E}(m, r)$ produces the ciphertext c of m, $c \in \mathcal{C}$. (random coins $r \in \mathcal{R}$).*
- *the decryption algorithm $\mathcal{D}$: By using a ciphertext c and the secret key sk, $\mathcal{D}$ returns the plaintext m, s.t.*

$$\Pr[\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m, r)) = m] = 1$$

or when it is an invalid ciphertext, outputs $\perp$. This algorithm is deterministic.

Definition 2. *Error-prone Public key encryption $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$*

- *$\mathcal{K}'$ is equivalent to $\mathcal{K}$, except that there may exist such a pair (pk, sk), corresponding to the $\mathcal{D}'$ but not to $\mathcal{D}$.*
- *$\mathcal{E}'$ is equivalent to $\mathcal{E}$, except that there may exist pairs (m, r) which do not fit the algorithm $\mathcal{D}$.*
- *$\mathcal{D}'$ decrypt the ciphertext $c \in \mathcal{C}$ with overwhelming probability, though,*

$$\Pr[\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m, r)) \neq m] \leq \texttt{negl}(k)$$

Definition 3. *A public key encryption scheme is said to be* OW-PCA *secure, if any polynomial-time adversary $\mathcal{A}$, with the public data and the help of the plaintext-checking oracle $\mathcal{O}_{\mathsf{pca}}$, can get the whole preimage of the ciphertext with* *at most q queries to $\mathcal{O}_{\mathsf{pca}}$, in a time bound t and a winning probability no more than negligible:*

$$\Pr\left[\begin{array}{l|l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k) & \\ m \leftarrow \mathcal{M}, r \overset{R}{\leftarrow} \Omega & m' = m \\ c \leftarrow \mathcal{E}_{\mathsf{pk}}(m;r), m' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{pca}}}(c) & \end{array}\right] \leq \mathtt{negl}(k)$$

Remark. Naturally, the security of OW-PCA primitive is dependent on the inverting the cipher even with the help of plaintext-checking oracle, which is a polynomial-time turing machine able to decide whether a cipher and a message is the corresponding encryption pair, or not, which is firstly introduced by Okamoto and Pointcheval [13]. The reason why we introduce the notion hereby is, some famous padding like REACT [13] has been used to enhance the security of error-prone primitive, e.g. NTRU, without concerning the decryption errors. The result is rigorously proved though, it loses the security as soon as decryption error based attack is employed. Furthermore, if we could show that some transforms are possible to rescue the provable security based on the OW-PCA even in the presence of decryption errors, we may successfully enhance lots of the public key encryption primitives, since almost all commonly used PKEs are in OW-PCA security.

Beyond the one-wayness, the *polynomial indistinguishability* [8] of the encryption can make the leakage of any partial information as hard as that of the whole plaintext. In order to make sense in the strongest attack scenario, the IND should be considered in the CCA model, called IND-CCA [17], which has become the *de facto* requirement of the public key cryptosystem, as follows.

Definition 4. *A public key encryption scheme is* IND-CCA *secure, if there exists no polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ who, under the help of the decryption oracle, can distinguish the encryption of two equal-length, distinct plaintexts, with the probability significantly greater than 1/2 (the only restriction is that the target ciphertext cannot be sent to the decryption oracle directly). More formally, the scheme is* IND-CCA *secure, if with the time bound t, decryption oracle querying bound q, the following is satisfied:*

$$\Pr_{\substack{b \overset{R}{\leftarrow} \{0,1\} \\ r \overset{R}{\leftarrow} \Omega}}\left[\begin{array}{l|l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k) & \\ (m_0, m_1, s) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathsf{pk}) & \\ c \leftarrow \mathcal{E}_{\mathsf{pk}}(m_b; r) & \hat{b} = b \\ \hat{b} \leftarrow \mathcal{A}_2^{\mathcal{O}}(c, m_0, m_1, s, \mathsf{pk}) & \end{array}\right] \leq \frac{1}{2} + \mathtt{negl}(k)$$

Remark. IND-CCA security is such a strong security notion that it is considered to leak no single bit of the useful information against even very dangerous attack. On the other hand, however, it is subject to the decryption errors as well. For example, the famous Naor-Yung paradigm [11], which uses two independent public keys to encrypt one same message, together with some proof that the message two ciphertext encrypted is the same, is denied as long as decryption errors occur. Thus, we can find that the failure of decryption leads to not only efficiency lost, but also security flaw. Similarly for Ajtai-Dwork scheme [1], the decryption errors were claimed to be eliminated by Goldreich et al [7], however, later was pointed put to be insecure or totally impractical by Nguyen [12]. Very recently, Proos gave a successful attack based on decryption failure of NTRU, which denied the provable security of many previous transforms, such as REACT-NTRU, and OAEP-NTRU.

3 Proos's Attack

In 2003, Proos [16] provided an attack which for an error-prone public key encryption $(\mathcal{K}', \mathcal{E}', \mathcal{D}')$, can break the scheme totally, i.e. to find the secret key, whereas the scheme remains IND-CCA secure, if with perfect decryption$(\mathcal{K}, \mathcal{E}, \mathcal{D})$.

If an encryption scheme has the perfect decryption, the act of decrypting a valid ciphertext will provide no useful information to attackers. However, if the error-prone decryption is employed, the error occurred may give useful information for attackers to determine the information of the secret key, such as whether a valid ciphertext is correctly encrypted or not. Note that even a valid ciphertext is encrypted correctly, the secret information is still possible to leak due to the imperfect decryption. Next we will explain the attack by Proos [2].

3.1 Decipherable Ciphertext Attacks

Let $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ be an error-prone public key encryption primitive. Given a randomly generated key pair $(\mathsf{pk},\mathsf{sk})$, and a decipherable oracle, $DC_{(\mathsf{pk},\mathsf{sk})}$ is an oracle which on input $(x,\ r,\ y)$ s.t. $\mathcal{E}_{\mathsf{pk}}(x,r) = y$ returns whether or not $\mathcal{D}_{\mathsf{sk}}(y) = x$. That is, a DC oracle can be used to determine if a valid ciphertext encrypted using pk can be correctly decrypted using sk. An attack using the public information and a DC oracle will be named as a decipherable ciphertext attack (DCA). Since naturally, a DC oracle gives additional information on the decryption, DCA is stronger than plain chosen plaintext attack. As it is also able to be simulated by decryption oracle, DCA is no stronger than CCA, obviously. And it is also adapted to the perfect decryption case, though seems a little redundant.

3.2 Attack on IND-CCA Transform

The encryption primitive may not be IND-CCA secure originally, however, there are many ways to enhance its security to the "appropriate" level, such as by the Optimal Asymmetric Encryption Padding (OAEP) [3], or Rapid Enhanced-security Asymmetric Cryptosystem Transform (REACT) [13] in the random oracle model.

Unfortunately, by the Proos's attack [16], guaranteed security for perfect decryption transform is not available any more for imperfect decryption ones. There exists such a scheme which can be proven secure in the perfect decryption scenario, but fails to hold the security in the imperfect decryption scenario, due to

[2] Due to the page limit, we omit the introduction of NTRU. Please refer to [9, 10] for the details why decryption errors occur.

leakage of useful information when answering the query of decipherable ciphertext attack. We will explain it in the following.

Take the OW-PCA secure PKE $(\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as an example, the enhanced security $\Pi^R = (\mathcal{K}^R, \mathcal{E}^R, \mathcal{D}^R)$ as the following:

- $\mathcal{K}^R = \mathcal{K}'$.
- $\mathcal{E}^R_{\mathsf{pk}}(m, s, r)$, for a message m, choose randomness s and r, let $c_1 = \mathcal{E}_{\mathsf{pk}}(s, r)$; and use cryptographic hash functions G and H to compute $c_2 = G(s) \oplus m$, with $c_3 = H(s, m, c_1, c_2)$. At last, define the ciphertext $c = (c_1, c_2, c_3)$.
- $\mathcal{D}^R_{\mathsf{sk}}(c_1, c_2, c_3)$, for a ciphertext $c = (c_1, c_2, c_3)$, $s' = \mathcal{D}_{\mathsf{sk}}(c_1)$, $m' = G(s') \oplus c_2$ and $c_3' = H(s, m', c_1, c_2)$. If $s' \in \mathcal{M}$ and $c_3' = c_3$ then output m', otherwise output $\perp$.

The above encryption is able to be proven the IND-CCA security without the presence of the decryption error, due to [13], while we hereby would like to consider another situation.

It turns out that when Π^R with the decryption errors, the IND-CCA security loses, due to the fact that OW-PCA PKE will not return $\perp$ for all invalid ciphertexts, which would provide a convenience for the attacker to break the PKE totally.

Consider a PKE with decryption errors $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$, and assume that the attacker has found k invalid ciphertext $y_1, y_2 ..., y_k$ of Π', then the attacker could build $\Pi^{R'}$ by using REACT transform as follows:

- $\mathcal{K}^{R'} = \mathcal{K}'$
- $\mathcal{E}^{R'}_{\mathsf{pk}}(m, r) = \mathcal{E}'_{\mathsf{pk}}(m, r)$
-

$$\mathcal{D}^{R'}_{\mathsf{sk}}(y) = \begin{cases} \mathcal{D}'_{\mathsf{sk}}(y) & \text{if } y \notin \{y_1, y_2, ..., y_k\} \\ x & \text{if } y = y_i, \text{ and the i-th bit of } \mathsf{sk} \text{ is } 1 \\ \perp & \text{otherwise} \end{cases}$$

Clearly, the new $\Pi^{R'}$ and original Π' are indistinguishable to a PCA attacker, thus $\Pi^{R'}$ is also OW-PCA secure.

The following attack could be applied to $\Pi^{R'}$ to recover the secret key. For $1 \le i \le k$ form a $\Pi^{R'}$ ciphertext $y_i^{R'}$ with $s = x$ and c_1 replaced by y_i. The $y_i^{R'}$ then could be sent to the DCA oracle with the knowledge that $y_i^{R'}$ will decrypt to $\perp$ if and only if the i-th bit of sk is zero. Thus $\Pi^{R'}$ is no more IND-CCA.

4 Generic Transforms

Since the underlying attack seems not able to be prevented by the previous scheme, [10] presented a new exclusive use padding for NTRU. However, we find that it is able to employ currently existed generic transform, 3-round OAEP [14] and its variant for this mission. The merit of the schemes is that they are not only useful to error-prone PKE, but also applicable to other common PKE;

furthermore, the 3-round OAEP has a good performance in efficiency, i.e. saves the bandwidth and is less redundant.

4.1 Concrete Construction

The scheme is described in the following:

Setup. On the security parameter, key generation algorithm randomly output a pair of $(\mathsf{pk},\mathsf{sk})$. Assume the random oracle family $\mathcal{H}$, and $F, G, H \xleftarrow{R} \mathcal{H}$,

$$\begin{aligned} F &: \{0,1\}^k \mapsto \{0,1\}^n, \\ G &: \{0,1\}^n \mapsto \{0,1\}^k, \\ H &: \{0,1\}^k \mapsto \{0,1\}^n. \end{aligned}$$

"||" represents bit concatenation. Let a sequence of bit zero be k_0-bit long, then the message length will be $n - k_0$.

Construction (3-round OAEP). The transform for deterministic encryption is defined as the following:

```
Encryption Enc(m)
    $w := [m||0^{k_0}] \oplus F(r)$
    $t := r \oplus G(w)$
    $s := H(t) \oplus w$
    $c := \mathcal{E}_{\mathsf{pk}}(s||t)$
```

```
Decryption Dec(c)
    $\mathcal{D}_{\mathsf{sk}}(c) := (s||t)$
    $w := H(t) \oplus s$
    $r := G(s) \oplus t$
    $m'||\mathsf{o} := w \oplus F(r)$
    If $\mathsf{o} = 0^{k_0}$, then $m = m'$
    otherwise, return $\perp$
```

A New Proposal. When the primitive is a probabilistic encryption scheme, the transform shall be changed, correspondingly. Use one more random oracle $H' \xleftarrow{R} \mathcal{H}$, $H' : \{0,1\}^{n+k} \mapsto \{0,1\}^{k'}$, and $r' = H'(m||r)$, be used as the required randomness of the probabilistic encryption. And the sequence of bit zero becomes not necessary. Others hold the same.

```
Encryption Enc(m)
    $w := m \oplus F(r)$
    $t := r \oplus G(w)$
    $s := H(t) \oplus w$
    $r' := H'(m||r)$
    $c := \mathcal{E}_{\mathsf{pk}}(s||t; r')$
```

```
Decryption Dec(c)
    $\mathcal{D}_{\mathsf{sk}}(c) := (s||t; r')$
    $w := H(t) \oplus s$
    $r := G(s) \oplus t$
    $m' := w \oplus F(r)$
    If $r' = H'(m'||r)$, then $m = m'$
    otherwise, return $\perp$
```

Remark. Actually, 3-round OAEP [14] has been proposed for its nice property of size efficiency. However, another property of this transform that it is inherently immune to the attack based on decryption failures, was not carefully studied and analyzed. Besides, since 2-round OAEP is widely used now, this is a good candidate for promoting uses. And we still provide a slight modification of 3-round OAEP, which adapts to the probabilistic encryption with decryption errors.

4.2 Security Analysis with Decryption Errors

We first give the explanation that error-prone encryptions combined with transforms will be immune to the decryption errors attack, and then check the chosen ciphertext security of both transforms.

Thwarting Decryption Errors. The above transforms could be combined with error-prone PKE Π' with sparse distributions of errors, and then decrease the probability of finding errors by the attacker. For the sake of analysis, we let Π' has an error probability α, where the probability is over the choice of (M, R) message and randomness pair. Thus, we could define the error probability of M and R as α where α is negligible when the message and randomness is chosen randomly [3].

Since we are going to reduce the successful probability of attacker to find such a "bad" pair that leads to the DCA attack, we just analyze that probability before and after the transform is applied. We start with the 3-round OAEP transform. Given some message randomness pair, we at first modify the message gradually, and change the randomness r due to the relation of paddings. The goal of the attacker is to control the input of Π', i.e. $(s||t)$, but only has access to m. But this is obviously difficult, because F, G, H are random oracles, the value passing through them becomes randomly. Therefore, the best strategy of the attacker, rather than randomly guessing, is to query the random oracle and check all the answers to find appropriate $(s||t)$ and their corresponding (m, r). We assume the queries to three random oracles are q_F, q_G, q_H respectively.

By analyzing 3-round OAEP, we have the following fact. Let us first see the t part of the input of Π', we have $t = r \oplus G(w)$, where $w = (m||0^{k_0}) \oplus F(r)$. For searching the appropriate t, the attacker should use three lists to record the query and answer to F, G and H, such as $(r_1, ..., r_{q_F})$, $(f_1, ..., f_{q_F})$ of oracle F, $(w_1, ..., w_{q_G})$, $(g_1, ..., g_{q_G})$ of oracle G and $(t_1, ..., t_{q_H})$, $(h_1, ..., h_{q_H})$ of oracle H. Then, we try to find some "bad" t, where there are corresponding g and h in the lists, s.t. $t = r_i \oplus g_j$, and further choose $s = w_j \oplus h_k$, thus get a candidate pair of (m, r) which leads to an fault decryption. Since the error probability is assumed as α, we can compute the possible probability is bounded by $\alpha \cdot q_G q_H$. From another view, error probability is fixed at first, then all $s||t$ candidates should fit the requirement of $m||0^{k_0}$. Since $m||0^{k_0} = f_i \oplus w_j$, after querying both oracles, $1 - (1 - 1/2^{k_0})^{q_F q_G} \approx q_F q_G / 2^{k_0}$.

According to our analysis, the 3-round OAEP could decrease the error probability occurred $\Pr[Error]_1$ at most

$$\Pr[Error]_1 \leq \alpha \cdot q_G q_H + \frac{q_F q_G}{2^{k_0}}$$

Note that we are able to adjust the parameter to let the above probability tiny enough.

On the probabilistic 3-round OAEP, the situation is likely, except that one more hash oracle is introduced. Hence we have to count this probability as well. Besides similar analysis as above, the attacker has to make the chosen message

[3] Attacker will use a more smart strategy to choose its target.

passing the check by H' hash function. Since it is querying all the possible value to the H' oracle, this probability will be bounded by $q_{H'}/2^{k'}$. The total bound is as the following.

$$\Pr[Error]_2 \leq \alpha \cdot q_G q_H + \frac{q_{H'}}{2^{k'}}$$

On the IND-CCA Security. The proof that 3-round OAEP is fulfilling with IND-CCA seems quite natural to understand after the work by [14]. We will refer to their paper. On the second transform, it seems that this modified version has not been proved yet, although another probabilistic version has been studied in [15], without achieving the exact IND-CCA security.

We just describe the proof strategy of the second transform, and refer to the full version of this paper for detailed proof. The original 3-round OAEP is provable with deterministic one-way permutation, however, it is not possibly to be proved with the probabilistic encryption. The reason is that for probabilistic encryption, even the input message is the same, the ciphertext could be different due to distinct randomness used. Thus for the oracle simulation process, the exact IND-CCA security (definition 4) will be lost easily. We apply one random oracle to check the validity of message and randomness pair, then all the possible pairs from attacker must be contained in the queries of this H' oracle (otherwise, we just simply reject the request). The above problem of original 3-round OAEP can be overcome. The security of this transform bases on OW-PCA (definition 3) security.

Remark. From above analysis, it may be raised a question that why not just use more hash functions and build more rounds. It is obvious that they are redundant and expensive. More importantly, the 2-round OAEP has been proved insecure against decryption errors attack [16], thus we naturally conclude that 3-round is the best efficient in the presence of decryption errors, from the viewpoint of bandwidth.

5 Conclusion

In this paper (extended abstract), we explain that existing generic transform is suitable for PKEs without or with imperfect decryption, and propose a new variant as well. We present the error probability bound, which decreases much capability of attackers to control the message and ciphertext pair in the CCA attack, and finally contribute to immunize the decryption failure for error-prone PKEs.

References

1. M. Ajtai and C. Dwork, “A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence”, in *STOC 1997*, pp. 284-293, 1997.
2. M. Bellare and P. Rogaway, “Random Oracles Are Practical: A paradigm for designing efficient protocols”, in *Proc. First Annual Conference on Computer and Communications Security*, ACM, 1993.

3. M. Bellare and P. Rogaway, “Optimal Asymmetric Encryption - How to Encrypt with RSA”, in *Eurocrypt’94*, LNCS 950, Springer-Verlag, pp. 92-111, 1995.
4. J.S. Coron, M. Joye, D. Naccache and P. Paillier, “Universal padding schemes for RSA”, in *Crypto’02*, LNCS 2442, Springer-Verlag, pp. 226-241, 2002.
5. C. Dwork, M. Naor and O. Reingold, “Immunizing Encryption Schemes from Decryption Errors”, in *Eurocrypt’04*, LNCS 3027, Springer-Verlag, pp. 342-360, 2004.
6. E. Fujisaki and T. Okamoto, “How to Enhance the Security of Public-Key Encryption at Minimum Cost”, in *PKC’99*, LNCS 1560, Springer-Verlag, pp. 53-68, 1999.
7. O. Goldreich, S. Goldwasser and S. Halevi, “Eliminating Decryption Errors in the Ajtai-Dwork Cryptosystem”, in *Crypto’97*, LNCS 1294, Springer-Verlag, pp. 105-111, 1997.
8. S. Goldwasser and S. Micali, “Probabilistic encryption”, *Journal of Computer Security*, 28:270–299, 1984.
9. J. Hoffstein, J. Pipher and J.H. Silverman, “NTRU: A Ring-Based Public Key Cryptosystem”, in *ANTS’98*, LNCS 1423, Springer-Verlag, pp. 267-288, 1998.
10. N. Howgrave-Graham, P. Nguyen, D. Pointcheval, J. Proos, J.H. Silverman, A. Singer and W. Whyte, “The Impact of Decryption Failures on the Security of NTRU Encryption”, in *Crypto’03*, LNCS 2729, Springer-Verlag, pp. 226-246, 2003.
11. M. Naor and M. Yung, “Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks”, *STOC 1990*, pp. 427-437, 1990.
12. P. Nguyen, “Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto ’97”, in *Crypto’99*, LNCS 1666, Springer-Verlag, pp. 288-304, 1999.
13. T. Okamoto and D. Pointcheval, “REACT: Rapid Enhanced-Security Asymmetric Cryptosystem Transform”, in *CT-RSA’01*, LNCS 2020, Springer-Verlag, pp 159-175, 2001.
14. D.H. Phan and D. Pointcheval, “Chosen-Ciphertext Security without Redundancy”, in *Asiacrypt’03*, LNCS 2894, Springer-Verlag, pp. 1-18, 2003.
15. D.H. Phan and D. Pointcheval, “OAEP 3-Round: A Generic and Secure Asymmetric Encryption Padding”, in *Asiacrypt’04*, LNCS 3329, pages 63-77, Springer-Verlag, 2004.
16. J. Proos, “Imperfect Decryption and an Attack on the NTRU Encryption Scheme”, Cryptology ePrint Archive: Report 2003/002.
17. C. Rackoff and D. Simon, “Non-interactive Zero-knowledge Proof of Knowledge and Chosen Ciphertext Attack”, in *Crypto’91*, LNCS 576, Springer-Verlag, pp. 433-444, 1992.
18. P.W. Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”, *FOCS 1994*, pp. 124-134, 1994.