Immunizing Encryption Schemes from Decryption Errors

Cynthia Dwork[1], Moni Naor[2⋆], and Omer Reingold[2⋆⋆]

[1] Microsoft Research, SVC
1065 L'Avenida
Mountain View, CA 94043
dwork@microsoft.com

[2] Weizmann Institute of Science
Rehovot 76100, Israel
{moni.naor,omer.reingold}@weizmann.ac.il

Abstract. We provide methods for transforming an encryption scheme susceptible to decryption errors into one that is immune to these errors. Immunity to decryption errors is vital when constructing non-malleable and chosen ciphertext secure encryption schemes via current techniques; in addition, it may help defend against certain cryptanalytic techniques, such as the attack of Proos [33] on the NTRU scheme.

When decryption errors are very infrequent, our transformation is extremely simple and efficient, almost free. To deal with significant error probabilities, we apply amplification techniques translated from a related information theoretic setting. These techniques allow us to correct even very weak encryption schemes where in addition to decryption errors, an adversary has substantial probability of breaking the scheme by decrypting random messages (without knowledge of the secret key). In other words, under these weak encryption schemes, the only guaranteed difference between the legitimate recipient and the adversary is in the frequency of decryption errors. All the above transformations work in a standard cryptographic model; specifically, they do not rely on a random oracle.

We also consider the random oracle model, where we give a simple transformation from a one-way encryption scheme which is error-prone into one that is immune to errors.

We conclude that error-prone cryptosystems can be used in order to create more secure cryptosystems.

1 Introduction

In their seminal paper on semantic security Goldwasser and Micali defined a public key encryption scheme as one where the decryption is perfect, i.e., given

⋆ Incumbent of the Judith Kleeman Professorial Chair. Research supported in part by a grant from the Israel Science Foundation. Part of this work was done while visiting Microsoft Research, SVC.

⋆⋆ Most of this research was performed while at AT&T Labs - Research and while visiting the Institute for Advanced Study, Princeton, NJ.

a properly formed ciphertext the answer is always the *unique* corresponding plaintext [20]. More formally, let the encryption algorithm be E and the corresponding decryption algorithm be D. If E maps a message m with random coins r to a ciphertext $c = E(m, r)$, then it is always the case that $D(E(m, r)) = m$. However, some cryptosystems do not satisfy this condition, two notable examples being the Ajtai-Dwork cryptosystem [1] and NTRU [21]. (In fact, sometimes a cryptosystem is deliberately designed to have ambiguous decryption; see more in Section 6.)

One might think that an encryption scheme with small probability of decryption error is merely an aesthetic nuisance, since the event of a decryption error can be compared to the event of an adversary guessing the secret key, which should be rare. However, serious difficulties arise in trying to construct cryptosystems secure under more stringent notions of security, such as non-malleability and chosen-ciphertext immunity, based on systems with ambiguous decryption. In fact, all known "bootstrapping" methods for constructing strong cryptosystems fail when the underlying one is susceptible to errors[3]. Furthermore, Proos was able to exploit decryption errors in his attack on the NTRU scheme [33]. Our goal in this work is to discuss general methods for eliminating errors and constructing secure cryptosystems based on less than perfect underlying schemes.

1.1 Random Oracles and the Real World

The literature contains constructions for cryptographic primitives in two well studied models: the random oracle world as described below, and the real world, where the assumption of a random oracle may not be justified. In general it is more difficult and involved to provide and prove correct constructions in the real world model.

If one makes the simplifying assumption that a specific function behaves as an idealized random function (random oracle), then it is possible to obtain simple and efficient constructions of public-key encryption schemes that are secure against chosen ciphertext attacks in the post-processing mode ("cca-post", also known as CCA2); these include OAEP and its variants [5, 3, 32, 15, 6], Fujisaki-Okamoto [14] and REACT [31][4]. However, it is not known if any one of these methods (or some other method) can be used to convert every public-key cryptosystem – including systems with decryption errors – that is semantically secure (or that satisfies even some weaker property such as one-wayness on the messages) against chosen plaintext attacks into one that is secure against stronger attacks, such as cca-post attacks (see below for more information on attacks). Among the problems in applying these approaches are that in the underlying "input" cryptosystem (1) there can exist ciphertexts which are valid encryptions

[3] One reason for the failure of those methods is that when the adversary chooses the input to the decryption algorithm, this input can have a distribution completely different from that of correctly encrypted messages and so the error probability may be large instead of small

[4] The meaning of such results is the subject of much debate (see e.g., [8, 13, 2]).

of two different plaintext messages; and (2) the decryption mechanism may sometimes fail to return "invalid" on an invalid ciphertext. As mentioned above, these problems were exploited by Proos [33] to attack various paddings of NTRU [30].

In the real world we have no idealized function, and we must do with what nature gives us. An important idea used either explicitly or at least implicitly in the construction of chosen ciphertext secure cryptosystem in the real world is to add some redundancy to the encryption and provide a proof of consistency of the ciphertext. The most general form of the proof of consistency is via a *non-interactive zero-knowledge* proof system (NIZKs) [11, 27, 29, 34], but there are also more specific methods [9, 10]. Here too a cryptosystem with possible decryption errors may cause problems in the construction. Take for instance the method that is based on a pair of keys together with a NIZK of consistency (this is the one suggested by Naor and Yung [29] and also a subsystem of the Dolev, Dwork, and Naor scheme [11]). A central idea in the proof of security is that knowing any of several private keys is sufficient for the decryption, and which one (of the several) is known is indistinguishable to the adversary. However, if there is no unique decryption, then seeing which plaintext is returned may leak which key is known, and the proof of security collapses.

Our Results:

We suggest methods for dealing with errors in both worlds described above:

In the land of random oracles: We provide a generic and efficient method for converting any public-key cryptosystem where decryption errors may occur, but where an adversary cannot retrieve the plaintext of a randomly chosen message (sometimes known as one-way cryptosystem), into one that is secure against chosen ciphertext attack in the post-processing mode. This is done in Section 5.

The real world: We show two transformations from cryptosystem with errors to ones without. When decryption errors are very infrequent, our transformation is extremely simple and efficient, almost free. The case of significant error probabilities is technically more involved. Our transformation for this case corrects even very weak encryption schemes where in addition to decryption errors, an adversary has substantial probability of breaking the scheme by decrypting random messages (without knowledge of the secret key). In other words, under these weak encryption schemes, the only guaranteed difference between the legitimate recipient (holder of the secret key) and the adversary is in the frequency of decryption errors: the legitimate recipient experiences fewer errors than does the adversary.

To demonstrate the subtleties of this task, consider the case where the legitimate recipient decrypts correctly with probability 9/10 (and let us assume for simplicity that otherwise he gets an error message), but the adversary decrypts correctly with probability 1/10. A natural approach is to use error correcting codes, setting the parameters in such a way that the legitimate recipient will have enough information to decode, whereas the adversary will get no information. This approach indeed works in the information theoretic counterpart of a channel where the receiver gets the piece of information with certain probability and the eavesdropper with another. But it is not clear how to carry it through in the computational setting. Therefore, the solutions given in this paper use a different approach: we apply amplification techniques translated from the related information theoretic setting of [35]. We note that here, too, the computational setting introduces additional complications.

The conclusion we reach is that once provided with noninteractive zero knowledge proof systems, one can convert essentially *any* public-key cryptosystem with decryption errors into one that is secure against chosen ciphertext attack in the postprocessing mode.

Related Work: In addition to the work mentioned above we should point out two specific papers that converted an error-prone scheme into an error free one. Goldreich, Goldwasser and Halevi [18] showed how to eliminate decryption errors in the Ajtai-Dwork [1] cryptosystem. Our methods, especially those of Section 3, can be seen as a general way of achieving that goal. In the papers of Howgrave-Graham et al. [23, 24] the problem of constructing an CCA-post-secure NTRU-based method in the random oracles world is considered.

2 Preliminaries

Notation and Conventions

We will abbreviate "probabilistic polynomial time Turing Machine" with PPTM. We use the notation $poly(\cdot)$ to refer to some polynomially bounded function and $neg(\cdot)$ to refer to some function that is smaller than $1/p(\cdot)$ for any polynomial $p(\cdot)$ (for all sufficiently large inputs). For any integer n, we let U_n denote the uniform distribution over $\{0,1\}^n$. We let the operation $\oplus$ on two bit-strings denote their bit-wise XOR.

2.1 Public-Key Encryption - Correctness

A public-key encryption scheme consists of three probabilistic polynomial time algorithms (G, E, D), for key generation, encryption and decryption respectively. For simplicity we fix n to be both the security parameter and input length, and assume that the message space is $\{0,1\}^n$. Algorithm G, for the key generation is given 1^n as input (as well as internal random coins), and outputs the public key and secret key pair (pk, sk). We have that $|pk| = |sk| = poly(n)$. E and D are, respectively, the encryption and decryption algorithms. E takes as input a public key pk, an n-bit plaintext message m, and uses internal random coins. We refer to the output $c \in E_{pk}(m)$ as the ciphertext. When we want to refer to E's additional $poly(n)$-long random input r explicitly, we will use the notation $E_{pk}(m; r)$. Finally, D takes as input a secret key sk and a ciphertext. The output of D is either a message m' (which may fail to equal the original message m) or $\perp$ to indicate invalid (we are deliberately not attaching semantics to a response of "invalid"). The standard definition of public-key encryption schemes requires perfect correctness. Namely, that if the input c to D_{sk} is well constructed using E_{sk}, then the output $D_{sk}(c)$ is supposed to retrieve the original plaintext. We make this explicit in the next definition.

Definition 1. *A public-key encryption scheme* (G, E, D) *is* perfectly correct *if the following holds:*

– *For every message* m *of length* n*, for every pair* (pk, sk) *generated by* G *on input* 1^n*, and all possible coin tosses of* E *and* D*, it should hold that* $D_{sk}(E_{pk}(m)) = m$*.*

Although we allowed D to output $\perp$ we made no assumption on the probability of $\perp$ being the output in case the ciphertext is indeed *invalid* (where invalid means that there do not exist m and r such that $c = E_{pk}(m; r)$).

We now want to relax the notion of public key-encryption so as to allow decryption errors. We define an encryption scheme to be α-correct, if the probability of decryption error is at most $1 - \alpha$.

Definition 2. *For any function* $\alpha : \mathbb{N} \mapsto [0, 1]$*, a public-key* (G, E, D) *encryption scheme is* α-correct *if* $Pr[D_{sk}(E_{pk}(m)) \neq m] \leq 1 - \alpha(n)$*, where the probability is taken over the random coins of* G *used to generate* (pk, sk) *on input* 1^n*, over the choice of* $m \in U_n$*, and over the random coins of* E *and* D*.*

In the above definition the error probability is taken over the random choice of the message (uniformly at random), the randomness of the encryption and decryption *and the choice of the key.* In particular, some keys may be completely useless as they don't allow decryption at all. We now consider the case that the bound on the decryption error holds for all keys or for all but a negligible fraction of the keys. These definitions are relevant here for two reasons: (1) Our transformations will be a bit more efficient if we only try to immunize against this kind of errors. (In the sense that the key of the revised scheme will only include a single key of the original scheme.) (2) Our transformations will produce schemes that are "almost-all-keys perfectly correct" rather than perfectly correct encryptions. This means that decryption errors can only occur with a negligible probability *over the choice of the key.* Note that such errors are usually much less harmful, and in particular such schemes can be made non-malleable using "standard" techniques (unlike the case where errors may occur for a substantial fraction of the keys).

Definition 3. *Let* (G, E, D) *be any public-key encryption scheme and* $\alpha : \mathbb{N} \mapsto [0, 1]$ *an arbitrary function.*

– (G, E, D) *is* all-keys α-correct *if for every pair* (pk, sk) *generated by* G *on input* 1^n*,* $Pr[D_{sk}(E_{pk}(m)) \neq m] \leq 1 - \alpha(n)$*, where the probability is taken over the choice of* $m \in U_n$*, and over the random coins of* E *and* D*.*
– (G, E, D) *is* almost-all-keys α-correct *if with probability* $(1 - neg(n))$ *over the random coins of* G *used to generate* (pk, sk) *on input* 1^n*,* $Pr[D_{sk}(E_{pk}(m)) \neq m] \leq 1 - \alpha(n)$*, where the probability is taken over the choice of* $m \in U_n$*, and over the random coins of* E *and* D*.*

– (G, E, D) *is* almost-all-keys perfectly correct *if with probability* $(1 - neg(n))$ *over the random coins of* G *used to generate* (pk, sk) *on input* 1^n, $Pr[D_{sk}(E_{pk}(m)) \neq m] = 0$, *where the probability is taken over the choice of* $m \in U_n$, *and over the random coins of* E *and* D.

2.2 Public-Key Encryption - Security

Semantic security [20] has established itself as essentially the minimal desired notion of security for encryption schemes. Intuitively, a public-key encryption scheme is semantically secure if anything that a polynomial-time adversary can compute about the plaintext m given the ciphertext $c = E_{pk}(m)$, it can also compute without access to c. Semantic security was shown in [20] to be equivalent to the indistinguishability of ciphertexts, which intuitively means that ciphertexts which correspond to different plaintexts are indistinguishable. Three basic modes of attack for which semantic security was considered are: chosen plaintext attack (which for public-key encryption essentially amounts to giving the adversary the public-key pk and allowing the adversary to decide the challenge distribution), and chosen ciphertext attack in the preprocessing and the postprocessing modes (in both the adversary also gets access to a decryption oracle; in the preprocessing mode this access ends when the ciphertext challenge is published). Semantic security under these attacks is denoted IND-CPA, IND-CCA-Post and IND-CCA-Pre respectively. An even stronger notion of security than semantic security is that of non-malleability [11]. Intuitively, here the adversary should not even gain a (non-negligible) advantage in creating an encryption of a message that relates to m. Non malleability with respect to the above attacks is denoted NM-CPA, NM-CCA-Post and NM-CCA-Pre respectively. For the formal definitions of the above notions we rely on [11].

Both semantic security and non-malleability were originally defined for perfectly correct encryption schemes. Nevertheless they are just as meaningful for schemes with decryption errors. Section 3 gives a very simple way of eliminating decryption errors (as long as they are very rare) while preserving each one of the above six notions of security. Section 4 shows how to immunize much weaker encryption schemes. Here decryption errors will be more likely (may even happen with probability $1 - poly$). In addition, we will make much weaker security assumptions: we will only bound the success probability of the adversary in "inverting E" and completely retrieving the plaintext message m. (Therefore, the only advantage the legitimate recipient has over the adversary is in the probability of decryption.) This notion of weak security is captured by the following definition.

Definition 4. *For any function* $\beta : \mathbb{N} \mapsto [0, 1]$, *a public-key encryption scheme is* β-one-way *(β-OW) if for every PPTM A,* $Pr[A((E_{pk}(m)) = m] \leq \beta(n) + neg(n)$, *where the probability is taken over the random coins of* G *used to generate* (pk, sk) *on input* 1^n, *over the choice of* $m \in U_n$, *and over the random coins of* E *and* A.

We note that unlike semantic security and non-malleability, this notion of security allows the encryption scheme E to be deterministic.

Pseudorandom Generators One of the transformations of this paper uses pseudorandom generators as a main tool. A pseudorandom generator is a function $prg : \{0,1\}^* \mapsto \{0,1\}^*$ such that on n-bit input x, the output $prg(x)$ is $\ell(n) > n$ bits long and such that $prg(U_n)$ is *computationally* indistinguishable from $U_{\ell(n)}$. See [17, 16] for a formal definition.

3 The Case of Infrequent Errors

This section describes a very efficient way for eliminating decryption errors when errors are very rare. If errors are too frequent to apply this technique directly, then one can first apply the amplification methods described in Section 4.

Let E be an encryption scheme where for every message m, the probability over the randomness r of E that $D_{sk}(E_{pk}(m;r)) \neq m$ is tiny. To correct this scheme we use the "reverse randomization" trick from the construction of Zaps [12] and commitment protocols [28] (which can be traced back to Lautemann's proof that BPP is in the polynomial time hierarchy [26]). The idea is very simple: by assumption, only a tiny fraction of "bad" random strings r lead to ciphertexts with decryption errors. Thus, we will arrange that the ciphertexts are constructed using only a rather small fraction of the possible values for r; the particular set of values will depend on the choice of public key. Very minimal independence in the selection of this subset will already assure that we are avoiding the bad strings with very high probability. In addition, the subset will be constructed to be pseudorandom, which will guarantee that the semantic security of the original scheme is preserved. Finally, the construction will ensure that the error probability is *only on the choice of encryption key* – if the encryption key is good, no ciphertext created with this encryption key will suffer a decryption error. The only significant computational cost incurred by this transformation is a single invocation of a pseudorandom generator (and in fact, this may already be performed to save on random bits, in which case the transformation is essentially for free).

For simplicity we state the next construction (and the corresponding theorem) under the assumption that the decryption algorithm D is deterministic. In the case of chosen-plaintext attack (which is probably the most interesting setting of the theorem), this can be obtained simply by fixing the randomness of D as part of the key. The case of chosen-ciphertext attacks is a bit more delicate but still the construction can be easily extended to randomized D.

Construction 31 *Let (G, E, D) be any public-key encryption scheme. Let $\ell(n)$ be the (polynomially bounded) number of bits used by E to encrypt n-bit messages. Without loss of generality assume that $\ell(n) > n$ (as E can always ignore part of its random input). Let prg be a pseudorandom generator that expands n bits to $\ell(n)$ bits.*

Define the public-key encryption scheme (G', E', D') as follows: on input 1^n, the generation algorithm G' outputs $((pk, \bar{r}), sk)$ where (pk, sk) is obtained by invoking G on the same input and $\bar{r} \in U_{\ell(n)}$. On an n-bit input m, the encryption *function E' uses an n-bit random string s and outputs $E_{pk}(m; prg(s) \oplus \bar{r})$. The decryption function D' is identical to D.*

Theorem 1. *Let (G, E, D) be any $(1-2^{-4n})$ correct public-key encryption scheme with D being deterministic. Define (G', E', D') as in Construction 31. Then (G', E', D') is an almost-all-key perfectly correct public-key encryption scheme. Furthermore, if (G, E, D) is NN-AAA secure with NN-AAA $\in$ {IND-CPA, IND-CCA-Post, IND-CCA-Pre, NM-CPA, NM-CCA-Post, NM-CCA-Pre} then so is (G', E', D').*

Proof. For any fixed value of $\bar{r}$, the distribution $prg(U_n) \oplus \bar{r}$ is pseudorandom. Therefore, it easily follows that (G', E', D') is NN-AAA secure (otherwise we could construct a distinguisher that breaks the pseudorandom generator).

It remains to prove the correctness of (G', E', D'), i.e. that with high probability over the choice of keys the scheme is perfectly correct. First, with probability at least $(1-2^{-n})$ over the choice of (pk, sk), the value $\Pr_{m,r}[D_{sk}(E_{pk}(m; r)) \neq m]$ is at most 2^{-3n}. Assume that (pk, sk) satisfies this property. Since $\bar{r}$ is uniformly distributed we also have that $\Pr_{m,s,\bar{r}}[D_{sk}(E_{pk}(m; prg(s) \oplus \bar{r})) \neq m] \leq 2^{-3n}$. As m and s are only n-bit long, we get by a union bound that the probability over $\bar{r}$ that for *some* m and s a decryption error $D_{sk}(E_{pk}(m; prg(s) \oplus \bar{r})) \neq m$ will occur is at most 2^{-n}. We can therefore conclude that for all but at most a 2^{-n+1} fraction of (G', E', D') keys $((pk, \bar{r}), sk)$ the scheme is perfectly correct.

Remark 1. The existence of the pseudorandom generator needed for Construction 31, follows from the security of (G, E, D) (under any one of the notions considered by the theorem). This is because the security of (G, E, D) implies the existence of one-way functions [25] which in turn imply the existence of pseudorandom generators [22].

Consider the construction of [11] for NM-CCA-post secure public key cryptosystems. This requires (i) a perfectly correct public-key cryptosystem which is semantically secure against chosen plaintext attacks (ii) A non-interactive zero-knowledge (NIZK) proof system for NP (that is for some specific language in NP) (iii) other primitives that can be based on one-way functions. Furthermore, if we replace in that construction the perfectly correct cryptosystem with one that is almost-all-keys-perfectly-correct, then all that happens is that the resulting construction is also of a similar nature. Therefore we can conclude

Corollary 2 *If $(1 - 2^{-4n})$-correct public-key encryption schemes semantically secure against chosen plaintext attacks exist and NIZK proof system for NP exist, then almost-all-key perfectly correct public-key encryption schemes which are NM-CCA-post secure public key cryptosystems exist.*

4 Immunizing Very Weak Encryption Schemes

We now consider much weaker encryption schemes than in Section 3. Here the encryption may only be α-correct and β-OW where α and β may be as small as

$1/poly$. Naturally, α has to be larger than β as otherwise the legitimate recipient of a message will have no advantage over the adversary (and such a scheme is useless and trivial to construct). The transformation given here works under the assumption that $\beta < \alpha^4/c$ for some fixed constant c. An interesting open problem is to give a transformation that works for even smaller gaps. Nevertheless, as we discuss below, having the transformation work for a gap $\beta - \alpha$ that is larger than an arbitrary constant, may involve improving the corresponding transformation in the related information-theoretic setting of [35].

4.1 Polarization in the statistical setting

Sahai and Vadhan [35], give an efficient transformation of a pair of distributions (X_0, X_1) (encoded by the circuits that sample them) into a new pair of distributions (Y_0, Y_1). The transformation "polarizes" the statistical distance between X_0 and X_1. If this distance is below some threshold β' then the statistical distance between Y_0 and Y_1 is exponentially small. If on the other hand the distance between X_0 and X_1 is larger than another threshold α' then the statistical distance between Y_0 and Y_1 is exponentially close to 1. The condition for which this transformation works is that $\beta' < \alpha'^2$.

What is the relation between this problem and ours? Consider an α-correct and β-OW encryption scheme, for one-bit messages. Let X_0 be the distribution of encryptions of 0 and X_1 the distribution of encryptions of 1. Intuitively we have that the legitimate recipient can distinguish these distributions with advantage $\alpha - (1 - \alpha) = 2\alpha - 1$ (recall that $\alpha > 1/2$), while the adversary cannot distinguish the distributions with advantage better than $2\beta - 1 < 2\alpha - 1$. Our transformation produces a new encryption scheme; let Y_0 and Y_1 be the corresponding distributions. We now have that the ability of the adversary to distinguish between Y_0 and Y_1 shrinks (to negligible), whereas the legitimate recipient distinguishes with probability that is exponentially close to 1. In fact, this intuitive similarity can be formalized to show that any transformation in the computational setting that is "sufficiently black box" implies a transformation in the statistical setting. This in particular implies that for our transformations to work for any constant gap $\alpha - \beta$, we may need to improve the transformation of [35] (or to use non black-box techniques).

What about the other direction? It seems much harder in general to translate transformations from the statistical setting to the computational one. Nevertheless, the transformations given in this section are heavily influenced by [35]. However, the computational versions of the amplification tools used in [35] are significantly weaker, which imposes additional complications and implies somewhat weaker bounds than those of [35].

4.2 Tools and basic transformations

To improve an α-correct and β-OW encryption scheme (G, E, D), we will use three basic transformations:

Parallel Repetition The encryption E^k of a k-tuple of messages $m_1, \ldots, m_k$ will be defined as $E^k(m_1, \ldots, m_k) = E(m_1), \ldots, E(m_k)$. A negative effect of this transformation is that the probability of correct decryption of the entire k-tuple is reduced to α^k. The gain of the transformation is that the probability of the adversary to break the one-wayness of E^k will also decrease below β (usually in an exponential rate as well). To bound this probability we apply a result of Bellare et al. [4] on the amplification of games in parallel execution. To conclude, this transformation makes decryption harder both for the legitimate recipient and for the adversary. As the adversary has a weaker starting point (success probability $\beta \ll \alpha$), it will be hurt more by the transformation.

Hard Core Bit Here we will transform an encryption scheme for strings to one that encrypts single bits. This will employ a hard core predicate in a rather standard fashion. The gain from this transformation is in turning the one-wayness of an encryption scheme into indistinguishability (which is easier to work with and is also our final goal).

Direct Product The encryption $E^{\otimes k}$ of a message m will be the concatenation of k independent encryptions of m under E. This transformation has the reverse affect to E^k: Decryption becomes easier both for the legitimate recipient and for the adversary. As the legitimate recipient has a better starting point (success probability $\alpha \gg \beta$), it will gain more by the transformation.

In the formal definition of E^k and $E^{\otimes k}$, we use *independently generated* keys for each one of the invocations of E by these schemes. This is necessary as a large fraction of the keys of E may be completely useless (i.e., do not allow decryption at all or completely reveal the message). So in order to amplify the security and correctness, we should use more than a single key. This can be avoided if we assume that (G, E, D) is α-correct and β-OW even after we fix the key of E (for all but negligible fraction of the keys). In such a case, the transformations of this paper will become much more efficient (in terms of key size). We now turn to the formal definition of the basic transformations.

Parallel Repetition

Definition 5. *Let (G, E, D) be any public-key encryption scheme, and let $k : \mathbf{N} \mapsto \mathbf{N}$ be any polynomially bounded function. Define (G^k, E^k, D^k) as follows: On input 1^n, the key-generating algorithm G^k invokes G, with input 1^n, $k = k(n)$ times using independent random coins for each invocation. The output of G^k is $(\bar{pk}, \bar{sk})$ where $\bar{pk} = pk_1, \ldots pk_k$, $\bar{sk} = sk_1, \ldots sk_k$, and (pk_i, sk_i) is the output of G in its i^{th} invocation. On input $\bar{m} = m_1, \ldots m_k$ the output $E^k_{\bar{pk}}(\bar{m})$ is defined by $E^k_{\bar{pk}}(\bar{m}) = E_{pk_1}(m_1), \ldots E_{pk_k}(m_k)$, where the k encryptions are performed with independent random coins. Finally, on input $\bar{c} = c_1, \ldots c_k$, the decryption algorithm $D^k_{\bar{sk}}$ tries to decrypt each c_i by applying $D_{sk_i}(c_i)$. It outputs $\perp$ if one of these invocations of D returned $\perp$ and otherwise $D^k_{\bar{sk}}$ outputs the sequence $D_{sk_1}(c_1) \ldots D_{sk_k}(c_k)$.*

Lemma 1. *Let (G, E, D) be any public-key encryption scheme, and let $k : \mathbf{N} \mapsto \mathbf{N}$ be any polynomially bounded function. If (G, E, D) is α-correct and β-OW with $\beta < 1 - 1/poly$, then (G^k, E^k, D^k) is α^k-correct and β'-OW for any $\beta' > 1/poly$ that satisfies $\beta' > 32/(1-\beta) \cdot e^{-k(1-\beta)^2/256}$.*

Proof. The correctness of (G^k, E^k, D^k) follows immediately from the definition. The security is much more delicate. Fortunately, it can be obtained as a simple corollary of a theorem of Bellare, Impagliazzo, and Naor regarding error probability in parallel execution of protocols of up to three rounds ([4] Theorem 4.1). Thus, we need to translate the breaking of (G^k, E^k, D^k) into winning the parallel execution of a game that is composed of at most three messages. Specifically, consider the following game between P and (an honest) V, where V invokes G to select (pk, sk), it selects a uniform message m and sends pk and $E_{pk}(m)$ to P. In return, P sends a message m' and wins if $m = m'$. From the one-wayness of (G, E, D) we get that the best *efficient* strategy of P can win with probability at most $\beta + neg$. Note that the probability of winning the k-times parallel repetition of this game is the same as breaking the one-wayness of (G^k, E^k, D^k). The lemma now follows from Theorem 4.1 of [4].

Hard Core Bit

For concreteness we will use the Goldreich-Levin (inner product) bit [19]. This could be replaced with hard-core bits implied by other error-correcting codes that have strong list-decoding properties.

Definition 6. *Let (G, E, D) be any public-key encryption scheme, where the encryption function operates on plaintexts of length $\ell \geq 1$, and let $k : \mathbf{N} \mapsto \mathbf{N}$ be any polynomially bounded function. Define $(G^\odot, E^\odot, D^\odot)$ as follows: $G^\odot$ is simply identical to G. On a one-bit message σ, the encryption function $E^\odot_{pk}$ samples two ℓ-bit strings m and r uniformly at random and outputs $E_{pk}(m), r, \langle m, r\rangle \oplus \sigma$, where $\langle m, r\rangle$ is the inner product of m and r (mod 2). On input c, r, σ' the decryption function $D^\odot_{pk}$ evaluates $m' = D_{pk}(c)$. If $m' \neq \perp$, then $D^\odot_{pk}$ outputs $\langle m', r\rangle \oplus \sigma'$, otherwise $D^\odot_{pk}$ outputs a random bit.*

Lemma 2. *Let (G, E, D) be any public-key encryption scheme. If (G, E, D) is α-correct and β-OW, then $(G^\odot, E^\odot, D^\odot)$ is $(1/2+\alpha/2)$-correct and $1/2+O(\sqrt{\beta})$-OW. In particular, if β is negligible then $(G^\odot, E^\odot, D^\odot)$ is IND-CPA secure.*

Proof. For correctness, note that if $m' = D_{pk}(c) = m$ (as in Definition 6), then $D^\odot_{pk}$ decrypts correctly with probability one. Otherwise $D^\odot_{pk}$ decrypts correctly with probability half (since the probability over r that for any $m' \neq m$ we have that $\langle m', r\rangle = \langle m, r\rangle$ is half). We can therefore conclude that the probability of correct decryption is at least $\alpha \cdot 1 + (1-\alpha) \cdot 1/2 = 1/2 + \alpha/2$.

For security, let us first assume that β is negligible. In this case $(G^\odot, E^\odot, D^\odot)$ is (1/2)-OW and equivalently is IND-CPA secure. Assume for the sake of contradiction that there exists an efficient adversary that decrypts $D^\odot_{pk}$ with probability $1/2 + 1/poly$ without access to sk. In this case, there is an efficient adversary that given $E_{pk}(m)$ and r guesses $\langle m, r\rangle$ with probability $1/2 + 1/poly$. Now we obtain from [19] that there exists an efficient adversary that given $E_{pk}(m)$ outputs m with probability $1/poly$. This contradicts the assumption that (G, E, D) is neg-OW.

Finally, let us consider the case where β is non-negligible. Assume for the sake of contradiction that there exists an efficient adversary that decrypts $D^{\odot}_{pk}$ with probability $1/2 + \epsilon$, where $\epsilon = c \cdot \sqrt{\beta}$ for some large constant c (note that $\epsilon > 1/poly$). This again implies the existence of an efficient adversary that given $E_{pk}(m)$ and r guesses $\langle m, r\rangle$ with the same probability. Using a tight enough version of the reconstruction algorithm for the Goldreich-Levin hard-core bit, we can conclude that there exists an efficient adversary that given $E_{pk}(m)$ computes a list of $O(1/\epsilon^2)$ candidates that include m with probability $1/2$. This means that this adversary can also guess m with probability $\Omega(\epsilon^2)$ which can be made say 2β by setting the constant c to be large enough. This contradicts the β-one-wayness of (G, E, D) and completes the proof of the lemma.

Direct Product

Definition 7. *Let (G, E, D) be any public-key encryption scheme, and let $k : \mathbb{N} \mapsto \mathbb{N}$ be any polynomially bounded function. Define $(G^{\otimes k}, E^{\otimes k}, D^{\otimes k})$ as follows: On input 1^n, the key-generating algorithm $G^{\otimes k}$ invokes G, with input 1^n, $k = k(n)$ times using independent random coins for each invocation. The output of $G^{\otimes k}$ is $(\bar{pk}, \bar{sk})$ where $\bar{pk} = pk_1, \ldots pk_k$, $\bar{sk} = sk_1, \ldots sk_k$, and (pk_i, sk_i) is the output of G in its i^{th} invocation. On input m the output $E^{\otimes k}_{\bar{pk}}(m)$ is defined by $E^{\otimes k}_{\bar{pk}}(m) = E_{pk_1}(m), \ldots E_{pk_k}(m)$, where the k encryptions are performed with independent random coins. Finally, on input $\bar{c} = c_1, \ldots c_k$, the decryption algorithm $D^{\otimes k}_{\bar{sk}}$ tries to decrypt each c_i by applying $D_{sk_i}(c_i)$. It outputs the value that is obtained the largest number of times (ties are resolved arbitrarily).*

We will use the direct product transformation only for encryptions of single bits. In this case, it is convenient to express correctness and security in terms of the advantage over half.

Lemma 3. *Let (G, E, D) be any public-key encryption scheme over the message space $\{0, 1\}$, and let $k : \mathbb{N} \mapsto \mathbb{N}$ be any polynomially bounded function. If (G, E, D) is $(1/2 + \alpha)$-correct and $(1/2 + \beta)$-OW, then $(G^{\otimes k}, E^{\otimes k}, D^{\otimes k})$ is $(1/2 + k\beta)$-OW and for every $\epsilon > 0$, it is $(1 - \epsilon)$-correct as long as $k > c \cdot 1/\alpha^2 \cdot \log 1/\epsilon$ for some fixed constant c.*

Proof. The one-wayness of $(G^{\otimes k}, E^{\otimes k}, D^{\otimes k})$ is obtained by a standard hybrid argument. Correctness is also simple to show using Chernoff bound. We note that we assume here that decryption errors occur with roughly the same probability for encryptions of zero and encryptions of one. For example, it is sufficient to assume that both $\Pr[D_{sk}(E_{pk}(0)) = 0] > 1/2 + \alpha/2$ and $\Pr[D_{sk}(E_{pk}(1)) = 1] > 1/2 + \alpha/2$. This is with no loss of generality as biases of D (towards outputting zero or towards one) can always be corrected.

4.3 Combining the Basic Transformations

The three basic transformations defined above can be combined in various ways to improve α-correct and β-OW encryption schemes. The most efficient combination depends on the particular values of α and β. We will not attempt to optimize the efficiency of our transformations but rather to demonstrate their effectiveness. For that we consider two settings of the parameters: (1) β is an arbitrary constant smaller than one and α is also a constant smaller than one (that depends on β). (2) α is as small as $1/poly$ and β is non-negligible ($\beta = \Omega(\alpha^4)$).

Constant Decryption Errors

Theorem 3. *For any constant $\beta < 1$ there exists a constant $\alpha < 1$ such that if there exists an α-correct and β-OW public-key encryption scheme then there exists an almost-all-keys perfectly-correct IND-CPA secure public-key encryption scheme.*

Proof. Set α to be a constant such that $e^{-(1-\beta)^2/256} < \alpha^8$ and let (G_0, E_0, D_0) be an α-correct and β-OW public-key encryption scheme. Define the following systems:

- $(G_1, E_1, D_1) = (G_0^{k_1}, E_0^{k_1}, D_0^{k_1})$ where $k_1 = log_\alpha(1/n)$. Lemma 1 implies that (G_1, E_1, D_1) is $(1/n)$-correct and $O(1/n^8)$-OW.
- $(G_2, E_2, D_2) = (G_1^{\odot}, E_1^{\odot}, D_1^{\odot})$. Lemma 2 implies that (G_2, E_2, D_2) is $(1/2 + n/2)$-correct and $(1/2 + O(1/n^4))$-OW.
- $(G_3, E_3, D_3) = (G_2^{\otimes k_2}, E_2^{\otimes k_2}, D_2^{\otimes k_2})$ where $k_2 = O(n^3)$, for which Lemma 3 implies that (G_3, E_3, D_3) is $(1 - 2^{-5n})$-correct and $(1/2 + O(1/n))$-OW.
- $(G_4, E_4, D_4) = (G_3^n, E_3^n, D_0^n)$. Lemma 1 implies that (G_1, E_1, D_1) is $(1 - 2^{-5n})^n$-correct, which means that it is also $(1 - n \cdot 2^{-5n})$-correct. In addition it is $(1/p)$-OW *for any polynomial* p. Thus it is also *neg*-OW.
- $(G_5, E_5, D_5) = (G_4^{\odot}, E_4^{\odot}, D_4^{\odot})$. Lemma 2 implies that (G_5, E_5, D_5) is $(1 - (n/2) \cdot 2^{-5n})$-correct and IND-CPA secure.

Theorem 3 now follows as a corollary of Theorem 1.

Very Frequent Decryption Errors

Theorem 4. *There exists some positive constant c such that for any functions $\alpha > 1/poly$ and $\beta < \alpha^4/c$ the following holds: If there exists an α-correct and β-OW public-key encryption scheme then there exists an almost-all-keys perfectly-correct IND-CPA secure public-key encryption scheme.*

Proof. Let (G_0, E_0, D_0) be an α-correct and β-OW public-key encryption scheme. The conditions of the theorem imply that it is also (α^4/c)-OW.

Define $(G_1, E_1, D_1) = (G_0^{\odot}, E_0^{\odot}, D_0^{\odot})$. Lemma 2 implies that (G_1, E_1, D_1) is $(1/2 + \alpha/2)$-correct and $(1/2 + O(\alpha^2/\sqrt{c}))$-OW.

Define $(G_2, E_2, D_2) = (G_1^{\otimes k}, E_1^{\otimes k}, D_1^{\otimes k})$. For any constant $\epsilon > 0$ we can let $k = O(1/\alpha^2)$ (with the constant hidden in the big O notation depending on ϵ), such that Lemma 3 will imply that (G_2, E_2, D_2) is $(1-\epsilon)$-correct and $(1/2 + O(1/\sqrt{c}))$-OW. Setting c to be a large enough constant implies that (G_2, E_2, D_2) is $(3/4)$-OW. In other words, for any constant $\epsilon > 0$, if c is a large enough constant, there exists a $(1-\epsilon)$-correct and $(3/4)$-OW encryption scheme. Theorem 4 now follows as a corollary of Theorem 3.

4.4 Conclusion - Obtaining Non-Malleability

As discussed in the introduction, one of the main motivations in dealing with decryption errors is obtaining non-malleability and chosen ciphertext security. As with Corollary 2 we now get from Theorem 4 the following corollary.

Corollary 5 *There exists some positive constant c such that for any functions $\alpha > 1/poly$ and $\beta < \alpha^4/c$ the following holds: If there exists an α-correct and β-OW public-key encryption scheme and NIZK proof system for NP exist, then there exists an almost-all-keys perfectly-correct NM-CCA-post secure public-key encryption scheme.*

5 Dealing with Errors Using Random Oracles

In this section we provide an integrated construction for transforming error-prone public-key encryption schemes with some negligible probability of error that are not necessarily secure against chosen ciphertext attacks into schemes that enjoy non-malleability against a chosen ciphertext attack of the post-processing kind. The advantage over the construction of Section 3 is that it works for *any* negligible probability of error (no need to first decrease the error probability to $2^{-\Omega(n)}$ where n is the message length).

Let (G, E, D) be a public-key encryption scheme that for public key pk maps a message $m \in \{0,1\}^n$ and random coins string $r \in \{0,1\}^\ell$ into a ciphertext $c = E_{pk}(m, r)$ (since we may start with a scheme that is not necessarily semantically secure, we consider also the case of deterministic encryption, so ℓ may be 0). We assume without loss of generality that the decryption algorithm D is deterministic[5]. The properties that we assume E satisfies are:

- α **correctness and few bad pairs** For a random message m and random r we have $\Pr[D_{sk}(E_{pk}(m,r)) \neq m] \leq 1 - \alpha(n)$, where $1-\alpha(n)$ is negligible. The probability is over the choice of m, r. We call a pair (m, r) where $D_{sk}(E_{pk}(m,r)) \neq m$ a *bad* pair. The set of bad pairs is sparse in $\{0,1\}^{n+\ell}$
- **One-wayness** For any polynomial time adversary $\mathcal{A}$ and for $c = E_{pk}(m, r)$ for random m and r we have $\Pr_{m,r}[\mathcal{A}(c, pk) = m]$ is negligible. In other words, E is 0-OW.

[5] This may justified, for instance by applying a pseudo-random function to the message in order to obtain the random bits and adding the seed of the function to the secret key.

In addition to the public-key cryptosystem E satisfying the above conditions, we require (i) a shared-key encryption scheme F_S which is NM-CCA-post secure. The keys S are of length k bits. Note that such schemes are easy to construct from pseudo-random functions (see [11]); and (ii) Four functions $H_1 : \{0,1\}^{n/2} \mapsto \{0,1\}^{n/2}$, $H_2 : \{0,1\}^{n/2} \mapsto \{0,1\}^{n/2}$, $H_3 : \{0,1\}^{n/2} \mapsto \{0,1\}^{\ell}$ and $H_4 : \{0,1\}^{n/2} \mapsto \{0,1\}^k$ which will be modelled as ideal random functions. We assume that n is sufficiently large so that $2^{n/2}$ is infeasible.

Construction 51 *Let (G, E, D) be a public-key encryption scheme, H_1, H_2, H_3, H_4 be idealized random functions as above and F_S be shared-key encryption scheme as above.*

Generation *G' operates the same as G and generates a public key pk and secret key sk.*

Encryption E': *Choose $t \in_R \{0,1\}^{n/2}$. Compute $z = H_1(t)$ and $w = H_2(z) \oplus t$ and $r = H_3(z \circ w)$. The encrypted message is composed of two parts (c_1, c_2):*
- *The generated $c_1 = E_{pk}(z \circ w, r)$*
- *The plaintext m itself is encrypted with the shared-key encryption scheme F_s with key $s = H_4(t)$, i.e. $c_2 = F_s(m)$.*

Decryption D': *Given ciphertext (c_1, c_2):*
1. *Apply D to c_1 and obtain candidates for z and w. Set $t = H_2(z) \oplus w$ and $r = H_3(z \circ w)$.*
2. *Check that $H_1(t) = z$ and that for $r = H_3(z \circ w)$ we have that $c_1 = E(z \circ w, r)$.*
3. *Check, using $s = H_4(t)$, that c_2 is a valid ciphertext under F_s.*
4. *If any of the tests fails, output invalid ($\perp$). Otherwise, output the decryption of c_2 using s.*

Note that once $t \in \{0,1\}^{n/2}$ has been chosen, there is unique ciphertext (c_1, c_2) generated from t and encrypting m, which we denote $E'_{pk}(m, t)$. Furthermore, for any ciphertext, once the corresponding $t \in \{0,1\}^{n/2}$ is known, it is easy to decrypt the ciphertext *without access to sk*. This is the key for obtaining security against chosen ciphertext attacks (since it is possible to follow the adversary calls to H_1).

Why does this process immunize against decryption errors? The point is *not* that the decryption errors have disappeared, but that it is hard to find them. We can partition all strings (of length equal to $|E_{pk}(z \circ w, r)|$) into those that are in the range of E (i.e., such that there exist m and r such that the string is equal to $E_{pk}(m, r)$) and those that are not. Consider a candidate ciphertext (c_1, c_2) that is given to the decryption procedure D'. If the prefix of the ciphertext (i.e. c_1) is not in the range of E, then it is going to be rejected by D' (at Step 2). So the security rests on the hardness of finding among the *bad* pairs $(z \circ w, r)$ one where $r = H_3(z \circ w)$ *and* $H_1(H_2(z) \oplus w) = z$. This is difficult for any *fixed* (but sparse) set of bad pairs and a random set of functions H_1, H_2, and H_3 even for an all powerful adversary who is simply restricted in the number of calls to H_1, H_2, and H_3. In particular, as we will explain, if there are q_1 calls to

H_1 and q_2 calls to H_2 then the probability that the adversary finds a bad pair that passes the test is bounded by $q_1(1-\alpha) + q_1q_2/2^{n/2}$. The first term comes from the "natural" method for constructing a pair that satisfies the constraints: Choose an arbitrary y. Apply H_1 to y and call the result z, so that $z = H_1(y)$. Define $w = H_2(z) \oplus y$. Then $r = H_3(z \circ w)$, and we have the pair $(z \circ w, r)$ satisfying the necessary constraints. Note that the pair is completely determined by y, once the random oracles are fixed, and the pair is random, because the oracles are random. So for any method of choosing y the probability of hitting a bad pair is $(1-\alpha)$. This gives us the first term. For the second term, suppose during its history the adversary invokes H_2 a total of q_2 times, say, on inputs $x_1, x_2, \ldots, x_{q_2}$. Let y be arbitrary. Define $w_i = y \oplus H_2(x_i)$, for $i = 1, \ldots, q_2$. We now check to see if $H_1(y) \in \{x_1, \ldots, x_{q_2}\}$. Suppose indeed that $H_1(y) = x_i$ (an event that occurs with probability at most $q_2/2^{n/2}$). Let $z = x_i$. Then we have that $z = H_1(y) = H_1(w_i \oplus H_2(x_i)) = H_1(w_i \oplus H_2(z))$. We let $r = H_3(z \oplus w_i)$ and again we have a pair satisfying the constraints. The total number of pairs we can hope to generate this way is $q_1q_2/2^{n/2}$.

Why does this process protect against chosen ciphertext attacks? This is very much for the same reason that the Fujisaki-Okamoto [14] scheme is secure. Note that hardness of finding a bad pair is true also for someone knowing the private key sk of E, that is *even the creator of the cryptosystem cannot find a bad pair.* Therefore, even under a chosen ciphertext attack w.h.p. a bad pair will not be found. So w.h.p. on all queries given during the attack there is only one response. Furthermore, this response can be given by someone who is aware of the attacker's calls to H_1 (by going over all candidates for t). The addition of the function H_4 and the shared key scheme F_S transforms the system from a one-way scheme into one that is non-malleably secure against chosen ciphertext attacks. From these sketched arguments we get:

Theorem 6. *If (G, E, D) is $(1-neg)$-correct and neg-one-way then (G', E', D') is $(1-neg)$-correct and NM-CCA-post secure.*

6 Conclusions and Open Problems

We have shown how to eliminate decryption errors in encryption schemes (and even handle non-negligible success probability of the adversary). It is interesting to note that sometimes such ambiguity is actually desirable. This is the case with *deniable encryption* [7], where the goal is, in order to protect the privacy of the conversation, to allow a sender to claim that the plaintext corresponding to a given ciphertext is different than the one actually sent.

As discussed in Section 4, an interesting open problem is to give a transformation that deals with α-correct and β-OW encryption schemes when the gap between α and β is very small. For example, we may hope to have $\beta - \alpha$ be an arbitrary constant or even $1/poly$. Nevertheless, as discussed there, having such a strong transformation may involve improving the corresponding transformation in the related information-theoretic setting of [35].

Acknowledgments

We thank Eran Tromer for initially pointing us to Proos's work, Shafi Goldwasser for raising our interest in the problem and Russell Impagliazzo, Adam Smith and Salil Vadhan for conversations concerning amplification. We thank the anonymous referees for helpful comments.

References

1. M. Ajtai and C. Dwork, *A public-key cryptosystem with worst-case/average-case equivalence*, Proceedings 29th Annual ACM Symposium on the Theory of Computing, El Paso, TX, 1997, pp. 284–293.
2. M. Bellare, A. Boldyreva and A. Palacio, *A Separation between the Random-Oracle Model and the Standard Model for a Hybrid Encryption Problem*, Cryptology ePrint Archive.
3. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. *Relations among notions of security for public-key encryption schemes,* Advances in Cryptology – CRYPTO'98, LNCS 1462, Springer, pp. 26–45.
4. M. Bellare, R. Impagliazzo and M. Naor, *Does parallel repetition lower the error in computationally sound protocols?*, in Proceedings 38th Annual IEEE Symposium on Foundations of Computer Science, Miami Beach, FL, 1997, pp. 374–383.
5. M. Bellare and P. Rogaway, P. Optimal Asymmetric Encryption.In Advances in Cryptology - EUROCRYPT '94 (1995), vol. 950 of LNCS, Springer-Verlag, pp. 92111.
6. D. Boneh, *Simplified OAEP for the RSA and Rabin Functions*, Advances in Cryptology - CRYPTO 2001, LNCS2139, Springer 2001, pp. 275–291.
7. R. Canetti, C. Dwork, M. Naor and R. Ostrovsky, *Deniable Encryption,* Advances in Cryptology - CRYPTO'97, LNCS 1294, Springer, 1997, pp. 90–104.
8. R. Canetti, O. Goldreich, and S. Halevi, *The random oracle methodology*, in Proceedings 30th Annual ACM Symposium on the Theory of Computing, Dallas, TX, 1998, pp. 209–218.
9. R. Cramer and V. Shoup, *A practical public key cryptosystem provable secure against adaptive chosen ciphertext attack*, in Advances in Cryptology—Crypto '98, Lecture Notes in Comput. Sci. 1462, Springer-Verlag, New York, 1998, pp. 13–25.
10. R. Cramer and V. Shoup, Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption, Advances in Cryptology – EUROCRYPT 2002, LNCS 2332, pp. 45–64, Springer Verlag, 2002.
11. D. Dolev, C. Dwork and M. Naor, *Non-malleable Cryptography*, Siam J. on Computing, vol 30, 2000, pp. 391–437.
12. C. Dwork, M. Naor, *Zaps and Their Applications*, Proc. 41st IEEE Symposium on Foundations of Computer Science, pp. 283–293. Full version: ECCC, Report TR02-001, `www.eccc.uni-trier.de/eccc/`.
13. C. Dwork, M. Naor, O. Reingold, L. J. Stockmeyer, Magic Functions, Proc. IEEE FOCS 1999, pp. 523–534.
14. E. Fujisaki and T. Okamoto, How to Enhance the Security of Public-Key Encryption at Minimum Cost. In PKC '99 (1999), vol. 1560 of LNCS, Springer-Verlag, pp. 5368.
15. E. Fujisaki, T. Okamoto, D. Pointcheval, J. Stern, *RSA-OAEP Is Secure under the RSA Assumption* Advances in Cryptology – CRYPTO 2001, Springer, 2001, pp. 260–274.
16. O. Goldreich, **Foundations of Cryptography**, Cambridge, 2001.

17. O. Goldreich, **Modern cryptography, probabilistic proofs and pseudo-randomness**. *Algorithms and Combinatorics*, vol. 17, Springer-Verlag, 1998.
18. O. Goldreich, S. Goldwasser, and S. Halevi, *Eliminating decryption errors in the Ajtai-Dwork cryptosystem*, Advances in Cryptology – CRYPTO'97, Springer, Lecture Notes in Computer Science, 1294, 1997, pp. 105–111.
19. O. Goldreich and L. Levin, A hard-core predicate for all one-way functions, *Proc. 21st Ann. ACM Symp. on Theory of Computing*, 1989, pp. 25-32.
20. S. Goldwasser and S. Micali, Probabilistic Encryption, *Journal of Computer and System Sciences*, vol. 28, 1984, pp. 270-299.
21. J. Hoffstein, J. Pipher, J. H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem, Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, J.P. Buhler (ed.), Lecture Notes in Computer Science 1423, Springer-Verlag, Berlin, 1998, pp. 267–288.
22. J. Hastad, R. Impagliazzo, L. A. Levin and M. Luby, Construction of a pseudo-random generator from any one-way function, *SIAM Journal on Computing*, vol 28(4), 1999, pp. 1364-1396.
23. N. Howgrave-Graham, P. Nguyen, D. Pointcheval, J. Proos, J. H. Silverman, A. Singer, W. Whyte *The Impact of Decryption Failures on the Security of NTRU Encryption*, Proc. Crypto 2003
24. N. Howgrave-Graham, J. H. Silverman, A. Singer and W. Whyte *NAEP: Provable Security in the Presence of Decryption Failures*, Available: `http://www.ntru.com/cryptolab/pdf/NAEP.pdf`
25. R. Impagliazzo and M. Luby, *One-way functions are essential to computational based cryptography*, in Proceedings 30th IEEE Symposium on the Foundation of Computer Science, Research Triangle Park, NC, 1989, pp. 230–235.
26. C. Lautemann, *BPP and the Polynomial-time Hierarchy*, Information Processing Letters vol. 17(4), 1983, pp. 215–217.
27. Y. Lindell, *A Simpler Construction of CCA2-Secure Public-Key Encryption Under General Assumptions*, Advances in Cryptology—Proceedings Eurocrypt 2003, LNCS 2656, 2003, pp. 241–254.
28. M. Naor, *Bit Commitment Using Pseudorandomness*, J. of Cryptology vol. 4(2), 1991, pp. 151–158.
29. M. Naor and M. Yung, *Public-key cryptosystems provably secure against chosen ciphertext attacks* in Proceedings 22nd Annual ACM Symposium on the Theory of Computing, Baltimore, MD, 1990, pp. 427–437.
30. P. Q. Nguyen, D. Pointcheval: *Analysis and Improvements of NTRU Encryption Paddings*, Advances in Cryptology—Proceedings Crypto'2002, Lecture Notes in Computer Science 2442 Springer 2002, pp. 210–225.
31. T. Okamoto and D. Pointcheval, *REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform*, In Proc. of CT-RSA'01 (2001), vol. 2020 of LNCS, Springer-Verlag, pp. 159175.
32. V. Shoup, *OAEP Reconsidered,* Journal of Cryptology 15(4): 223-249 (2002).
33. J. Proos, *Imperfect Decryption and an Attack on the NTRU Ecnryption Scheme*, IACR Cryptlogy Archive, Report 02/2003.
34. A. Sahai, *Non-Malleable Non-Interactive Zero Knowledge and Achieving Chosen-Ciphertext Security*, Proc. 40th IEEE Symposium on Foundations of Computer Science, 1999, pp. 543–553.
35. A. Sahai and S. Vadhan, *A Complete Promise Problem for Statistical Zero-Knowledge*, *Proceedings of the 38th Annual Symposium on the Foundations of Computer Science*, 1997, pp.$\tilde{4}$48–457. Full version: Electronic Colloquium on Computational Complexity TR00-084.