A Complete Public-Key Cryptosystem

Dima Grigoriev[1], Edward A. Hirsch[2] *, and Konstantin Pervyshev[3]

[1] IRMAR, Université de Rennes,
Campus de Beaulieu, 35042 Rennes, cedex France,
Web: http://name.math.univ-rennes1.fr/dimitri.grigoriev/
[2] Steklov Institute of Mathematics at St.Petersburg,
27 Fontanka, 191023 St.Petersburg, Russia,
Web: http://logic.pdmi.ras.ru/~hirsch/
[3] St.Petersburg State University, Russia,
Mathematics and Mechanics Department,
Web: http://logic.pdmi.ras.ru/~pervyshev/

Abstract. We present a cryptosystem which is complete for the class of probabilistic public-key cryptosystems with bounded error. Besides traditional encryption schemes such as RSA and El Gamal, this class contains probabilistic encryption of Goldwasser-Micali as well as Ajtai-Dwork and NTRU cryptosystems. The latter two are known to make errors with some small positive probability.

To our best knowledge, no complete public-key cryptosystem has been known before, whether encryption/decryption errors are allowed or not. At the same time, other complete primitives such as Levin's universal one-way function [1] have been known for a long time.

1 Introduction

Reductions between computational problems have great meaning for complexity theory. In particular, reduction is a natural tool for establishing computational hardness of various algorithmic problems. With reductions, one can prove that some problem is at least as hard as another one. Furthermore, for some classes of problems, it is possible to find a *complete* problem, which is at least as hard as *any* other problem in the class.

The same idea can be applied to cryptography, where problems of breaking cryptographic primitives are considered. The notion of a reduction and the notion of completeness may be adapted to this cryptographic setting. In particular, they may help to find the hardest to break implementations of cryptographic primitives. If a complete public-key cryptosystem $\mathcal{C}$ exists, one could argue that under the assumption that secure public-key cryptography is possible at all, this is $\mathcal{C}$ what provides a secure cryptosystem.

There is a number of important cryptographic primitives such as, to name a few, one-way functions and public-key cryptosystems. In [1], Levin presented a *universal (complete)* one-way function. Informally speaking, it is a polynomial-time computable function which is the hardest to invert (read "to break") among all polynomial-time computable functions. Therefore, if hard-to-invert polynomial-time functions exist, then Levin's function is one of them. Readers are referred to [7, Section 2.4.1] for a more detailed discussion. Surprisingly, complete public-key encryption schemes have been unknown so far.

* Supported in part by INTAS project 04-77-7173 and Federal Agency for Science and Innovations, contract 02.442.11.7290.

1.1 Probabilistic Encryption

The main insight of this paper is to look for a complete public-key encryption scheme in the class of probabilistic encryption schemes with bounded error rather than in the class of all "perfect" cryptosystems (that do not have encryption/decryption errors). This approach is motivated by a recent result of Fortnow and Santhanam [9, Theorem 2] on a time hierarchy for probabilistic algorithms that are allowed to fail on a small fraction of inputs.

The fact that we consider "imperfect" probabilistic encryption schemes makes our result even more interesting. We present a public-key cryptosystem which is complete for the class that contains all "traditional" public-key encryption schemes as well as probabilistic encryption of Goldwasser and Micali [2], one of the first encryptions schemes, Ajtai-Dwork cryptosystem [3], which is based on lattice problems, and NTRU [4]. The latter two permit small positive probability of error during encryption and decryption of a message.

In recent works, it has been shown that error probability of probabilistic cryptosystems can be easily reduced down to negligible one [5, 6]. Therefore it is of great interest to find the hardest to break probabilistic public-key encryption scheme, even one with small positive probability of error.

1.2 Reductions

In order to formalize the notion of "the hardest-to-break" cryptographic primitive, we use reductions. A cryptographic primitive S_1 is reducible to another primitive S_2, iff there exists a probabilistic oracle procedure R (called a *reduction*) that, given an oracle A that breaks S_2, breaks S_1.

As an illustration, Levin's result on universal one-way function means that inversion of any polynomial-time computable function F is reducible to inversion of the universal one-way function. One may see that this notion of a reduction is quite similar to the one used in complexity theory: task of breaking a primitive is an analogue of recognizing a language.

1.3 Our Result

We present an encryption scheme[4] which is complete for the class of all probabilistic public-key encryption schemes with bounded error. Our encryption scheme is provably secure[5] under the weakest possible assumption that secure public-key encryption schemes exist at all.

We stress that our encryption scheme is of theoretical interest only. It is impractical as it is secure only for huge values of its security parameter. For such values, key generation, encryption and decryption algorithms, which are polynomial-time in the security parameter, require a huge amount of time and communication. However, from now on, one may think of a concrete public-key encryption scheme when (dis)proving the existence of hard-to-break public-key encryptions.

As complete cryptographic primitives are the natural bridges from cryptography to complexity theory, our successful attempt to construct a complete cryptosystem suggests that probabilistic encryption schemes with errors may be easier to research rather than "perfect" ones.

[4] We use the terms "cryptosystem" and "encryption scheme" interchangeably.

[5] We consider schemes for encrypting a single bit; the adversary is given the public key and the encrypted message.

1.4 Previous Work

To our best knowledge, it is the first successful attempt to construct a complete public-key cryptosystem, although the existence of a complete (universal) one-way function has been known for a long time [1, 7]. More of that, the idea of a universal one-way function may be traced back to Levin's optimal algorithm for **NP**-complete problems. The same idea was used later to construct a universal learning algorithm [8].

We prefer to use the term "complete" instead of "universal" in this paper to stress that there is a reduction (of a certain form) between any public-key encryption scheme and our complete cryptosystem. From the view point of complexity theory, the latter is stronger than a statement that the existence of some hard-to-break cryptosystem implies that our cryptosystem is hard-to-break also.

In complexity theory, the existence of a complete language for some class of languages is closely related to the existence of a time hierarchy in the class: namely, both complete languages and hierarchy theorems are known for "syntactic" (i.e., efficiently enumerable) classes, but are hard to devise for other ("semantic") classes. Recently, much work has been put into proving the existence of time hierarchies for probabilistic algorithms with two-sided error (for which no complete problems are known). An interesting paper of Fortnow and Santhanam [9] shows that there exists a time hierarchy for heuristic probabilistic algorithms with two-sided error. Such algorithms are permitted not to satisfy error bound on some small fraction of inputs. Our research to big extent is motivated by this work.

An important ingridient of our proof is the amplification of cryptosystem correctness, i.e. a procedure that reduces the error probability of a given cryptosystem while moderately worsening its security. Such correctness amplification technique is known from the work of Dwork, Naor and Reingold [6]. For more discussion on this topic, see recent papers by Holenstein and Renner [10, 11].

2 Definitions

Definition 1. *A public key encryption scheme S consists of three probabilistic worst-case polynomial-time algorithms (G, E, D), for key generation, encryption and decryption respectively.*

Key generation algorithm G is given a security parameter 1^n as input, and outputs the public key and secret key pair $(pk, sk) \leftarrow G(1^n)$. Encryption algorithm E takes as input a public key pk, a one-bit plaintext message m and produces a ciphertext $M = E_{pk}(m)$. Finally, decryption algorithm D takes as input a secret key sk and a ciphertext. The output of D is a message $m' = D_{sk}(M)$, which may fail to equal the original message m. For more details, see [12].

Definition 2. *A public key encryption scheme is $\delta(n)$-correct, iff for all sufficiently large security parameters n, for any one-bit message $m \in \{0, 1\}$,*

$$\mathbf{Pr}[D_{sk}(E_{pk}(m)) = m] \geq \delta(n) \quad \textit{for} \quad (pk, sk) \leftarrow G(1^n),$$

where probability is taken over randomness of algorithms G, E and D.

Definition 3. *A probabilistic black-box A $\epsilon(n)$-breaks an encryption scheme (G, E, D), if for infinitely many security parameters n,*

$$\mathbf{Pr}[A_{pk}(1^n, E_{pk}(m)) = m] \geq \epsilon(n) \quad \textit{for} \quad (pk, sk) \leftarrow G(1^n),$$

where probability is taken over uniform selection of a one-bit message m and over randomness of algorithms G, E and of black-box A.

Since we consider public-key encryption schemes that encode one-bit messages, it makes sense to consider only those $\epsilon(n)$ and $\delta(n)$ that are greater than $1/2$. Also, note that Definition 2 implies that knowing private key sk one can $\delta(n)$-break the encryption scheme.

We could have defined $\delta(n)$-correct encryption schemes in a different way, so that the probability in the definition is also taken over randomly chosen messages $m \in \{0,1\}$ like in Definition 3. However, Definition 2 that we use makes our construction of a public-key encryption scheme a little bit easier.

Definition 4. *A probabilistic black-box A breaks an encryption scheme (G,E,D), if it $(1/2 + 1/p(n))$-breaks the encryption scheme for some polynomial $p(n)$.*

An encryption scheme (G,E,D) is secure, if no probabilistic polynomial-time Turing machine breaks (G,E,D).

Some encryption schemes may be harder to break than others. So let us define a reduction between a pair of encryption schemes similarly to that between two languages. In some sense, the notion of a break of a cryptosystem is an analogue of recognition of a language in complexity theory.

Definition 5. *An encryption scheme (G_1,E_1,D_1) is reducible to an encryption scheme (G_2,E_2,D_2), if there exists a probabilistic polynomial time oracle machine R, such that for* any *probabilistic black-box A that breaks (G_2,E_2,D_2), R^A breaks (G_1,E_1,D_1).*

Obviously, if encryption scheme (G_1,E_1,D_1) is secure, then (G_2,E_2,D_2) is also secure. Thus, it makes sense to find a complete encryption scheme, a one to which any other encryption scheme, secure or not, is reducible.

Proposition 1. *Reductions are transitive.*

3 Our Approach

In our construction of a complete public key encryption scheme (which is the hardest to break among all encryption schemes), we employ a natural idea of combining all possible public-key encryption schemes $(G_i,E_i,D_i)_{i\in\mathbb{N}}$, in order to achieve the maximal possible level of security. For a given security parameter n, we combine the first n of encryption schemes in a way that their combination is secure if and only if at least one of them is secure.

To encrypt a message $m \in \{0,1\}$, our complete public key encryption scheme $\mathcal{S} = (\mathcal{G},\mathcal{E},\mathcal{D})$ performs as follows. First, key generator $\mathcal{G}$ on input 1^n obtains pairs of public and private keys $(pk_1,sk_1) \leftarrow G_1(1^n),\ldots(pk_n,sk_n) \leftarrow G_n(1^n)$. As we will see in Lemma 3, w.l.o.g. key generators G_i work in time, say, n^4.

Then, encryption algorithm $\mathcal{E}$ selects $x_1,\ldots x_n \in_R \{0,1\}$ [6] and produces a codeword $(E_{1,pk_1}(x_1),$ $\ldots, E_{n,pk_n}(x_n), x_1\oplus\ldots\oplus x_n\oplus m)$. Evidently, if at least one encryption E_i is secure, then no adversary has noticeable advantage over random guessing in learning x_i from $E_{i,pk_i}(x_i)$, thus in learning m from $x_1 \oplus \ldots \oplus x_n \oplus m$, even if all other encryption schemes are insecure.

[6] i.e. independently and uniformly at random from $\{0,1\}$

In order to extract the message m from $x_1 \oplus \ldots \oplus x_n \oplus m$, having our private keys $(sk_1, \ldots sk_n)$, we need to recover all x_j from $E_{j,pk_j}(x_j)$. However, most of encryption schemes are probably not correct in the sense that $D_{j,sk_j}(E_{j,pk_j}(x_j))$ is not necessarily equal to x_j. We overcome this difficulty by using only those encryption schemes that are correct in a certain sense. And this is exactly the place where we have to allow encryption/decryption errors.

Although it is impossible to test a given encryption scheme for 100% correctness, we are able to determine with high confidence whether a given scheme is almost correct. So we select those encryption schemes that pass our test for "almost correctness" and combine them in our construction. Since the errors in several almost correct schemes may accumulate and increase, we employ correctness amplification technique [6], the last ingredient of our proof.

Definition 6. *We denote the class of all 2/3-correct public key encryption schemes (every algorithm of which runs in probabilistic polynomial time) with* **PKCS**.

Sometimes, a key generation algorithm is allowed to run in expected polynomial time [12, Section 7.1]. However, since we allow encryption schemes to fail with positive probability, these two possible definitions are equivalent.

Notice that we do not include any requirement on security into this definition. Therefore some schemes from **PKCS** may be much more easier (sometimes trivial) to break than others. Our goal is exactly to find the scheme that is the hardest-to-break in this class.

The choice of the constant 2/3 in the definition is arbitrary because it is possible to efficiently amplify correctness of an encryption scheme while preserving its security. In fact, we could have defined **PKCS** as a class of encryption schemes that are $(1 - \frac{1}{p(n)})$-correct for any fixed polynomial $p(n)$.

The remaining part of the paper constitutes a proof of the following theorem.

Theorem 1. *There exists a complete* **PKCS**.

4 Correctness Test

In order to distinguish for a given security parameter n between those encryption schemes that are $(\delta + \Delta)$-correct and those that are less than $(\delta - \Delta)$-correct, we will use the straightforward test given below:

Correctness Test $((G, E, D), 1^n, T, \delta, k)$
  for $m \in \{0, 1\}$ **do**
    for $i := 1$ **to** k **do**
      Run $G(1^n)$ for T steps to obtain a pair of keys (pk, sk)
      Run $E_{pk}(m)$ for T steps to obtain a ciphertext M
      Run $D_{sk}(M)$ for T steps to obtain a decrypted message m'
      if $m' = m$ **then** $X_{m,i} := 1$ **else** $X_{m,i} := 0$
  if for both $m \in \{0, 1\}$, $\sum_{i=1}^{k} X_{m,i} > \delta \cdot k$ **then accept else reject**

In this test, if one of the algorithms G, E, and D wants to work for more than T steps, it is stopped after allowed T execution steps, and its output tape by that moment is considered as the result of the execution.

Proposition 2 (Chernoff bound). *Let $X_1, X_2, \ldots, X_n$ be independent Poisson trials with $\mathbf{Pr}[X_i = 1] = p_i$ and $\mathbf{Pr}[X_i = 0] = 1 - p_i$. Then if X is the sum of X_i and if μ is $\mathbb{E}[X]$, for any λ, $0 < \lambda \leq \mu$:*

$$\mathbf{Pr}[X \leq \mu - \lambda] < e^{-\lambda^2/2n}, \qquad \mathbf{Pr}[X \geq \mu + \lambda] < e^{-\lambda^2/2n}.$$

The following proposition can be proved using the Chernoff and union bounds:

Proposition 3. *For a given security parameter n, every $(\delta + \Delta)$-correct encryption scheme, every algorithm of which runs in time T, passes the test almost for sure, failing with probability less than $2e^{-\frac{\Delta^2 \cdot k}{2}}$. At the same time, if the encryption scheme, every algorithm of which is restricted to run no more than time T, is less than $(\delta - \Delta)$-correct, then it passes the test with probability at most $e^{-\frac{\Delta^2 \cdot k}{2}}$.*

Consequently, we can distinguish in probabilistic polynomial time with high confidence those encryption schemes that are, say, $(1 - \frac{1}{12n})$-correct from those that are not $(1 - \frac{1}{4n})$-correct. Thus, if we have n encryption schemes $S_i, 1 \leq i \leq n$, and combine into a new encryption scheme (Section 6.1) only those of them that pass the test, we get at least 2/3-correct encryption scheme (by union bound). It remains to show that every scheme $S_i \in \mathbf{PKCS}$, which is 2/3-correct, is reducible to some $(1 - \frac{1}{12n})$-correct encryption scheme, that is to amplify the correctness of encryption schemes (cf. [6]).

5 Correctness Amplification

Assume our encryption scheme (G, E, D) is secure and $(1/2+\delta(n))$-correct. Our aim is to improve its reliability while moderately worsening its security. The recent work by Dwork, Naor and Reingold [6] provides a way to achieve this. However, we present an amplification scheme here because we need to prove that there is a reduction (in our sense) from a weakly correct encryption scheme to an encryption scheme with amplified correctness.

Definition 7. *Let $S^{k(n)} = (G, E, D)^{k(n)}$ be the encryption scheme obtained from encryption scheme $S = (G, E, D)$ as follows. The key generation algorithm $G^{k(n)}$ on input 1^n runs $k(n)$ copies of $G(1^n)$ with independent sources of randomness to obtain pairs of keys $(pk_1, sk_1), \ldots, (pk_{k(n)}, sk_{k(n)})$. The encryption algorithm $E^{k(n)}$, provided a public key $pk_1, \ldots, pk_{k(n)}$ and a message m, runs $k(n)$ copies of $E(m, pk_i)$ (one copy per one value of i) with independent sources of randomness to obtain codewords $M_1, \ldots, M_{k(n)}$. Finally, the decryption algorithm $D^{k(n)}$, provided a private key $sk_1, \ldots, sk_{k(n)}$ and an encoded message $M_1, \ldots, M_{k(n)}$, runs $k(n)$ copies of $D(M_i, sk_i)$ with independent sources of randomness to obtain codewords $m'_1, \ldots, m'_{k(n)}$ and outputs their majority value (we assume that $k(n)$ is odd).*

The next two lemmas are close to [6, Lemma 3]. In the first lemma, which is responsible for the security aspect of the construction, we prove that, for any polynomial $k(n)$, (G, E, D) is reducible to $(G, E, D)^{k(n)}$. Then, in the second lemma, responsible for the correctness, we show that the probability of error of $(G, E, D)^{k(n)}$ is exponentially smaller than the probability of error of the original cryptosystem.

Lemma 1. *Every encryption scheme $S = (G, E, D)$ is reducible via some reduction R to the encryption scheme $S^{k(n)} = (G, E, D)^{k(n)}$. Furthermore, if some black-box A $(1/2 + \epsilon(n))$-breaks $(G, E, D)^{k(n)}$, then R^A $(1/2 + \epsilon(n)/k(n))$-breaks (G, E, D).*

Proof. Assume an adversary A breaks $S^{k(n)}$ with probability greater than $1/2 + \epsilon(n)$ for infinitely many security parameters n. Let us construct an adversary R that breaks S with probability greater than $1/2 + \epsilon(n)/k(n)$ for the same set of security parameters n using adversary A as a black-box.

Our reduction R (with oracle A) on input composed by a public key pk, by a security parameter 1^n and by a codeword $M = E_{pk}(m)$ works as follows:

1. Choose i from $\{1, 2, \ldots, k\}$ uniformly at random. [7]
2. Generate keys $(pk_1, sk_1), \ldots, (pk_k, sk_k) \leftarrow G^k(1^n)$ (actually, we need only the public keys for our reduction).
3. Invoke encryption $(M_1, \ldots, M_{i-1}) \leftarrow E^{i-1}_{(pk_1, \ldots pk_{i-1})}(0)$.
4. Invoke encryption $(M_{i+1}, \ldots M_k) \leftarrow E^{k-i}_{(pk_{i+1}, \ldots pk_k)}(1)$.
5. Output $A(M_1, \ldots M_{i-1}, M, M_{i+1}, \ldots M_k)$.

Note that all invocations of key generator G and encryption E use independent sources of randomness. Then success probability of R^A is:

$$\begin{aligned}
\mathbf{Pr}[R^A(E_{pk}(m)) = m] &= \frac{1}{2}\mathbf{Pr}[R^A(E_{pk}(1)) = 1] + \frac{1}{2}\mathbf{Pr}[R^A(E_{pk}(0)) = 0] = \\
&= \frac{1}{2}\left(\frac{1}{k}\mathbf{Pr}[A(E^k(1)) = 1] + \sum_{i=2}^{k}\frac{1}{k}\mathbf{Pr}[A(E^{(i-1)}(0), E^{k-(i-1)}(1)) = 1]\right) + \\
&+ \frac{1}{2}\left(\sum_{i=1}^{k-1}\frac{1}{k}\mathbf{Pr}[A(E^i(0), E^{k-i}(1)) = 0] + \frac{1}{k}\mathbf{Pr}[A(E^k(0)) = 0]\right) = \\
&= \frac{1}{2k}\left(\mathbf{Pr}[A(E^k(1)) = 1] + \mathbf{Pr}[A(E^k(0)) = 0]\right) + \frac{k-1}{k}\cdot\frac{1}{2} > \\
&> \frac{1}{2k}\cdot 2\left(\frac{1}{2} + \epsilon\right) + \frac{k-1}{k}\cdot\frac{1}{2} = \left(\frac{1}{2k} + \frac{\epsilon}{k}\right) + \left(\frac{1}{2} - \frac{1}{2k}\right) = \frac{1}{2} + \epsilon/k,
\end{aligned}$$

where probability is taken over uniform choice of one-bit message m, randomness of key generations $pk \leftarrow G(1^n)$ and $pk_i \leftarrow G(1^n)$, randomness of encryptions $E_{pk}(m)$, $E_{pk_i}(0)$ and $E_{pk_i}(1)$ and over randomness of adversaries A and R.

Lemma 2. *If (G, E, D) is $(\frac{1}{2} + \delta(n))$-correct, then $(G, E, D)^{k(n)}$ is $(1 - e^{-\delta^2(n)k(n)/2})$-correct.*

Proof. Recall that $D^{k(n)}$ decodes the encrypted message $M_1, \ldots M_{k(n)}$, where $M_i = E_{pk_i}(m)$, by taking majority of outcomes $D_{sk_i}(M_i)$. The key pairs $(pk_i, sk_i) \leftarrow G(1^n)$ are generated independently, therefore, applying the Chernoff bound, we estimate the probability of error of the encryption scheme $(G, E, D)^{k(n)}$ on a message m as

$$\mathbf{Pr}[\sum X_i \leq k/2] = \mathbf{Pr}[\sum X_i \leq (1/2 + \delta)k - \delta k] < e^{-\frac{\delta^2 k^2}{2k}} = e^{-\frac{\delta^2 k}{2}},$$

where X_i indicates whether $D_{sk_i}(E_{pk_i}(m)) = m$.

Now we have a method of provably secure correctness amplification that we will employ in the completeness proof of a public-key encryption scheme that we present in the next section.

[7] For brevity, we omit parameter n and write k instead of $k(n)$ and ϵ instead of $\epsilon(n)$.

6 A Complete Public Key Encryption Scheme

Before giving a proof of Theorem 1, we prove a lemma that allows us to simplify the proof of our encryption scheme's completeness:

Lemma 3. *Every scheme $S \in$ **PKCS** is reducible to some scheme $S' \in$ **PKCS**, every algorithm of which runs in time n^2.*

*Every encryption scheme $S' \in$ **PKCS**, every algorithm of which runs in time n^2, is reducible to some $(1 - e^{-n})$-correct encryption scheme $S'' \in$ **PKCS**, every algorithm of which runs in time n^4.*

Proof. The first part of our lemma is proved by the standard padding argument. Note that, since the correctness 2/3 is a constant, it does not decrease when we apply this argument.

In order to prove the second part, it is sufficient to take $S'' = (S')^{72n+1}$ by Lemma 1 and Lemma 2.

6.1 The Construction and Proof of Correctness

Our complete public-key encryption scheme $\mathcal{S} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is given below. For the correctness test, we choose $\delta = \frac{1}{2} \cdot ((1 - \frac{1}{4n}) + (1 - \frac{1}{12n})) = 1 - \frac{1}{6n}$, therefore $\Delta = \frac{1}{12n}$. Also we choose $k = n^4$ (see Proposition 3). Time limit T is set to n^4 (see Lemma 3; it coincides with k only by chance).

Key generator $\mathcal{G}\,(1^n)$
  let $I = \emptyset$
  for all $i \in \{1, \dots n\}$
    if Correctness Test accepts
    input $((G_i, E_i, D_i), 1^n, T = n^4, \delta = 1 - \frac{1}{6n}, k = n^4)$ **then**
      (*) Run $G_i(1^n)$ for $T = n^4$ steps to obtain (pk_i, sk_i)
      Include i into set I
  return $\mathcal{P}K = (1^n, I, \{pk_i\}_{i \in I})$ and $\mathcal{S}K = (1^n, I, \{sk_i\}_{i \in I})$

Encryption $\mathcal{E}\,(\mu, \mathcal{P}K)$
  for all $i \in I$ **do**
    Select $m_i \in_R \{0, 1\}$
    (**) Run $E_{i,pk_i}(m_i)$ for $T = n^4$ steps to obtain M_i
  return $\mathcal{M} = (\{M_i\}_{i \in I}, \mu \oplus \bigoplus_{i \in I} m_i)$

Decryption $\mathcal{D}\,(\mathcal{M}, \mathcal{S}K)$
  for all $i \in I$ **do**
    Run $D_{i,sk_i}(M_i)$ for $T = n^4$ steps to obtain m'_i
  return $(\mu \oplus \bigoplus_{i \in I} m_i) \oplus \bigoplus_{i \in I} m'_i$

Obviously, it is an encryption scheme (every algorithm of which runs in polynomial time). Further, it is 2/3-correct, because any encryption scheme that is not $(1 - \frac{1}{4n})$-correct passes the correctness test with probability at most $e^{-\frac{\Delta^2 \cdot k}{2}} = e^{-\frac{n^2}{288}}$ by Proposition 3. Further, for all sufficiently large security parameters n, the set I is guaranted to be nonempty, because the identity encryption (the one that leaves message as it is) always passes the test.

6.2 Proof of Completeness

In order to prove the completeness of $\mathcal{S}$, assume that we have an encryption scheme $S_{j'} \in \mathbf{PKCS}$. By transitivity of reductions and Lemma 3, it is reducible to some $(1 - e^{-n})$-correct public-key encryption scheme $S_j = (G_j, E_j, D_j)$, every algorithm of which runs in time n^4. It remains to construct a reduction from S_j to $\mathcal{S}$.

We define a reduction R on input composed by a security parameter 1^n, by a public key pk and by a codeword $M = E_{j,pk}(m)$, where $m \in_R \{0,1\}$. Note that for any security parameters $n \geq j$, the encryption scheme S_j is among the first n encryption schemes that are combined in $\mathcal{S}$.

1. Simulate key generator $\mathcal{G}$ on input 1^n to obtain a public key $\mathcal{P}K$. The only one exception is that, instead of generating pk_j in step (*), R selects pk_j to be equal to pk (if occasionally S_j fails the correctness test and thus $j \notin I$, then give up). All the other $\{pk_i\}_{i \neq j}$, are generated by $\mathcal{G}$.
2. Simulate encryption $\mathcal{E}$ on input composed by a new uniformly selected message $\mu \in_R \{0,1\}$ and by the public key $\mathcal{P}K$ in order to obtain a codeword $\mathcal{M}$ with the following modification: instead of computing $M_j = E_{j,pk}(m_j)$ in step (**), R selects M_j to be equal to M. Nonetheless, m_j is generated (think of it as of R's guess of m) as well as all the other $\{m_i\}_{i \neq j}$. Then return ciphertext $\mathcal{M} = (\{M_i\}_{i \in I \setminus \{j\}} \cup M\}, \mu \oplus \bigoplus_{i \in I} m_i)$.
3. Provide oracle A with security parameter 1^n, public key $\mathcal{P}K$ and ask it to break the codeword $\mathcal{M}$. Output $A_{\mathcal{P}K}(1^n, \mathcal{M}) \oplus \mu \oplus m_j$.

By Definition 3, the probability that R^A succeeds is exactly

$$\mathbf{Pr}\left[R^A_{pk}(1^n, E_{j,pk}(m)) = m\right] \quad \textit{for} \quad pk \leftarrow G_j(1^n)$$

where probability is taken over a uniformly selected one-bit message m, over randomness of key generation $G_j(1^n)$, encryption $E_{j,pk}(m)$, reduction R_{pk} and over randomness of the invocation of probabilistic black-box A.

Conditioned on the event that $j \in I$, this probability, by the definition of reduction R and by the definition of cryptosystem $\mathcal{S} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$, is equal to

$$\mathbf{Pr}\left[A_{\mathcal{P}K}(1^n, \mathcal{E}_{\mathcal{P}K}(\mu \oplus m_j \oplus m)) = m \oplus \mu \oplus m_j\right] \quad \textit{for} \quad \mathcal{P}K \leftarrow \mathcal{G}(1^n),$$

where probability is taken over uniformly selected one-bit messages m, μ and m_j, over randomness of key generation $\mathcal{G}(1^n)$, encryption $\mathcal{E}_{\mathcal{P}K}$ and probabilistic black-box A. Notice that key generation $pk \leftarrow G_j(1^n)$ and encryption $E_{j,pk}(m)$ become parts of key generation $\mathcal{P}K \leftarrow \mathcal{G}(1^n)$ and encryption $\mathcal{E}_{\mathcal{P}K}$ correspondingly.

Let $\tilde{\mu} = \mu \oplus m_j \oplus m$. Then the probability above is equal to

$$\mathbf{Pr}\left[A_{\mathcal{P}K}(1^n, \mathcal{E}_{\mathcal{P}K}(\tilde{\mu})) = \tilde{\mu}\right] \quad \textit{for} \quad \mathcal{P}K \leftarrow \mathcal{G}(1^n),$$

where probability is taken over a uniformly selected one-bit message $\tilde{\mu}$, over randomness of key generation $\mathcal{G}(1^n)$, encryption $\mathcal{E}_{\mathcal{P}K}$ and probabilistic black-box A. This is exactly the success probability of adversary A.

To finish the proof, it remains to notice that the encryption scheme $S_j = (G_j, E_j, D_j)$ passes the correctness test with probability greater than $1 - e^{-n}$, thus $j \notin I$ with a negligible probability.

Acknowledgment

We thank Sergey Nikolenko for fruitful discussions on early stages of this work.

References

1. Levin: One-way functions and pseudorandom generators. Combinatorica **7** (1987)
2. Goldwasser, Micali: Probabilistic encryption. Journal of Computer and System Sciences **28** (1984)
3. Ajtai, Dwork: A public-key cryptosystem with worst-case/average-case equivalence. In: ACM Symposium on Theory of Computing (STOC). (1997)
4. Hoffstein, Pipher, Silverman: NTRU: A ring-based public key cryptosystem. In: 3rd International Algorithmic Number Theory Symposium (ANTS). (1998)
5. Goldreich, Goldwasser, Halevi: Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In: Proceedings of Crypto. (1997)
6. Dwork, Naor, Reingold: Immunizing encryption schemes from decryption errors. In: Advances in Cryptology: Proceedings of EUROCRYPT. (2004)
7. Goldreich: Foundations of Cryptography: Volume 1, Basic Tools. Cambridge University Press (2001)
8. Goldreich, Ron: On universal learning algorithms. Information Processing Letters **63** (1997)
9. Fortnow, Santhanam: Hierarchy theorems for probabilistic polynomial time. In: IEEE Symposium on Foundations of Computer Science (FOCS). (2004)
10. Holenstein, Renner: One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In: CRYPTO. (2005)
11. Holenstein: Key agreement from weak bit agreement. In: ACM Symposium on Theory of Computing (STOC). (2005)
12. Goldwasser, Bellare: Lecture notes on cryptography. Summer course on cryptography, MIT (2001)