Eliminating Decryption Errors in the Ajtai-Dwork Cryptosystem

Oded Goldreich[1] Shafi Goldwasser[1,2] Shai Halevi[2]

[1] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, ISRAEL.
[2] Laboratory for Computer Science, MIT, Cambridge, MA 02139.
E-mail: {oded, shafi, shaih}@theory.lcs.mit.edu

Abstract. Following Ajtai's lead, Ajtai and Dwork have recently introduced a public-key encryption scheme which is secure under the assumption that a certain computational problem on lattices is hard on the worst-case. Their encryption method may cause decryption errors, though with small probability (i.e., inversely proportional to the security parameter). In this paper we modify the encryption method of Ajtai and Dwork so that the legitimate receiver always recovers the message sent. That is, we make the Ajtai-Dwork Cryptosystem error-free.

Keywords: Public-key Encryption Schemes, Computational Problems in Lattices.

1 Introduction

A major project of our field is to find concrete hard problems which can be used for "doing Cryptography" (e.g., constructing encryption schemes, message-authentication codes and digital signatures). As current state of the art in Complexity Theory does not allow to prove that such (cryptographically-useful) problems are hard, one has to rely on unproven and yet plausible assumptions. It is thus important to have as many alternative/unrelated assumption as possible, so that Cryptography can be based on any one of them. So far there are very few alternatives; and so Ajtai's work [1], which suggests a new domain out of which adequately-hard problems can be found, marks an important day for Cryptography.

In particular, Ajtai constructed a one-way function based on the assumption that Lattice Reduction is hard in the worst-case. Following his lead, Ajtai and Dwork have recently introduced a public-key encryption scheme which is secure, provided that the following (worst-case complexity) assumption holds [2]:

Assumption ISVP (Infeasibility of Shortest Vector Problem): There exists no polynomial-time algorithm, which given an arbitrary basis for an n-dimensional lattice, having a "unique poly(n)-shortest vector", finds the shortest (non-zero) vector in the lattice. By having a *unique* poly(n)-*shortest vector* we mean that any vector of length at most poly(n) times bigger than the shortest vector is an integer multiple of the shortest vector.

The encryption method of Ajtai and Dwork [2], has a non-zero decryption-error probability. Specifically, when working with security parameter n, the ciphertext of the message bit '1' is decrypted to be a '0' with probability $\frac{1}{n}$. (The ciphertext corresponding to the message bit '0' is always decrypted as '0'.)

In this paper we modify the encryption method of Ajtai and Dwork so that every message is always decrypted correctly. Thus, we obtain a error-free encryption scheme which is secure under the same assumption used by Ajtai and Dwork.

2 The Encryption Scheme

In this section we recall the construction of Ajtai and Dwork [2] and describe our modification of it. We start by introducing a few notations which are used throughout the paper.

2.1 Notations

We denote the set of integers by $\mathcal{Z}$, and the set of real numbers by $\mathcal{R}$. For any number ϵ between 0 and $\frac{1}{2}$, we denote by $\mathcal{Z} \pm \epsilon$ the set of real numbers for which the distance to the nearest integer is at most ϵ.

The n-dimensional Euclidean space is denoted by $\mathcal{R}^n$. For two vectors $x, y \in \mathcal{R}^n$, we denote the inner-product of x and y by $\langle x, y \rangle$. Given a set of n linearly independent vectors $w_1, \ldots, w_n \in \mathcal{R}^n$, the *parallelepiped which is spanned by the w_i's* is the set

$$P(w_1, \ldots, w_n) \stackrel{\text{def}}{=} \left\{ \sum_i \alpha_i w_i \;:\; \alpha_i \in [0,1),\; i = 1, \ldots, n \right\}$$

The *width* of $P(w_1, \ldots, w_n)$ is the minimum over i of the Euclidean distance between w_i and the subspace spanned by the other w_j's.

Given a parallelepiped $P = P(w_1, \ldots, w_n)$ and a vector v, we *reduce v modulo P* by obtaining a vector $v' \in P$ so that $v' = v + \sum_i c_i w_i$, where the c_i are all integers. We denote this process by $v' = v \bmod P$.

2.2 The Ajtai-Dwork Construction

Let us recall the Ajtai-Dwork construction.[3] To simplify the exposition we present the scheme in terms of real numbers, but we always mean numbers with some fixed finite precision. (Following [2], one should use n-bit binary expansion of real numbers when working with security parameter n).

[3] The scheme which we describe below is slightly different than the original scheme in [2]. The difference between these schemes is insignificant, however (this is mostly a matter of presentation style).

Common Parameters. Given security parameter n, we let $m \stackrel{\text{def}}{=} n^3$, and $\rho_n \stackrel{\text{def}}{=} 2^{n \log n}$. We denote by B_n (for Big or cuBe) the n-dimensional cube of side-length ρ_n. Also, we denote by S_n (for Small or Sphere) the n-dimensional sphere of radius n^{-8}. Namely, we have

$$B_n \stackrel{\text{def}}{=} \{x \in \mathcal{R}^n : 0 \leq x_i < \rho_n,\ i = 1, \ldots, n\} \quad \text{and} \quad S_n \stackrel{\text{def}}{=} \{x \in \mathcal{R}^n : \|x\| \leq n^{-8}\}$$

Private-key. Given security parameter n, the private-key is a uniformly chosen vector in the n-dimensional unit sphere. We denote this vector by u.

Public-key. For a private key u, denote by $\mathcal{H}_u$ the distribution on points in B_n which is induced by the following process.

1. Pick a point a uniformly at random from the set $\{x \in B_n : \langle x, u \rangle \in \mathcal{Z}\}$.
2. For $i = 1, \ldots, n$, select $\delta_1, \ldots, \delta_n$ uniformly at random from S_n.
3. Output the point $v = a + \sum_i \delta_i$.

Using this notation, the public key which correspond to the private key u is obtained by picking the points $w_1, \ldots, w_n, v_1, \ldots, v_m$ independently at random from the distribution $\mathcal{H}_u$, subject to the constraint that the width of the parallelepiped $P(w_1, \ldots, w_n)$ is at least $n^{-2}\rho_n$. In the sequel, we often use the notations $\mathbf{w} \stackrel{\text{def}}{=} (w_1, \ldots, w_n)$, $\mathbf{v} \stackrel{\text{def}}{=} (v_1, \ldots, v_m)$, and $\mathbf{e} \stackrel{\text{def}}{=} (\mathbf{w}, \mathbf{v})$.
(Remark: It is shown in [2] that of we pick $w_1, \ldots, w_n$ uniformly in $\mathcal{H}_u$, then the width of $P(w_1, \ldots, w_n)$ will be large enough, with probability at least $1 - n^{-1/2}$.)

Encryption. The encryption works in a bit-by-bit fashion. Namely, to encrypt a string $s = \sigma_1\sigma_2 \ldots \sigma_\ell$, each bit σ_i is encrypted separately.

To encrypt a '0', we uniformly select $b_1, \ldots, b_m$ in $\{0, 1\}$, and reduce the vector $\sum_{i=1}^m b_i \cdot v_i$ modulo the parallelepiped $P(\mathbf{w})$. The vector $x = (\sum_{i=1}^m b_i \cdot v_i) \bmod P(\mathbf{w})$ is the ciphertext which correspond to the bit '0'.

To encrypt a '1' we uniformly select a vector x in the parallelepiped $P(\mathbf{w})$. This vector is the ciphertext which correspond to the bit '1'.

Decryption. Given a ciphertext, x, and the private-key u, we compute $\tau = \langle x, u \rangle$. We decrypt the ciphertext as a '0' if τ is within $1/n$ of some integer and decrypt it as a '1' otherwise.

Decryption errors. It is easy to see that if x is an encryption of '1', then the fractional part of $\langle x, u \rangle$ is distributed almost uniformly in $[0, 1)$. On the other hand, a simple argument show that if x is an encryption of '0' then the fractional part of $\langle x, u \rangle$ is always less than $1/n$ in absolute value. Thus, an encryption of '0' will always be decrypted as '0', and an encryption of '1' has a probability of $2/n$ to be decrypted as '0'.

2.3 An Error-free Construction

We proceed now to describe our modification which eliminates the decryption errors from the construction above. In this modified scheme, just like in the original Ajtai-Dwork scheme, encrypting a '0' results in a ciphertext x such that $\langle x, u\rangle$ is close to an integer. However, in our scheme we also make sure that encrypting a '1' results in a ciphertext x such that $\langle x, u\rangle$ is far from any integer. The modified scheme is as follows:

Common Parameters and private-key. The common parameters n, m, ρ_n, B_n and S_n, and the private key u, are set in exactly the same manner as in the original scheme.

Public-key (modified). The vectors $w_1, \ldots, w_n, v_1, \ldots, v_m$ are chosen in exactly the same manner as in the original scheme.

In addition, we pick i_1 uniformly at random from all the indices i for which $\langle a_i, u\rangle \in 2\mathcal{Z} + 1$, where a_i is the large vector used to generate v_i (i.e., $v_i = a_i + \sum_j \delta_j$). That is, i_1 is selected so that $\langle a_{i_1}, u\rangle$ is an odd integer. We note that with probability $1 - 2^{-\Omega(m)}$ such an index exists.[4] The public-key consists of the sequence of points $(w_1, \ldots, w_n, v_1, \ldots, v_m)$ and the integer i_1.

Encryption (modified). We encrypt a '0' just like in the original scheme, by uniformly selecting $b_1, \ldots, b_m \in \{0, 1\}$, and reducing the vector $\sum_{i=1}^m b_i \cdot v_i$ modulo the parallelepiped $P(\mathbf{w})$. The vector $x = (\sum_{i=1}^m b_i \cdot v_i) \bmod P(\mathbf{w})$ is the ciphertext which correspond to the bit '0'.

The difference is in the encryption of a '1'. We do that by uniformly selecting $b_1, \ldots, b_m \in \{0, 1\}$, and reducing the vector $\frac{1}{2}v_{i_1} + \sum_{i=1}^m b_i \cdot v_i$ modulo the parallelepiped $P(\mathbf{w})$. The vector $x = \left(\frac{1}{2}v_{i_1} + \sum_{i=1}^m b_i \cdot v_i\right) \bmod P(\mathbf{w})$ is the ciphertext which correspond to the bit '1'.

Decryption (modified)*:* Given a ciphertext, x, and the private-key u, we compute $\tau = \langle v, u\rangle$. We decrypt the ciphertext as a '0' if τ is within $1/4$ of some integer and decrypt it as a '1' otherwise.

In contrast to the encryption scheme in [2], we can show that in our scheme there is no decryption error. Specifically, we have:

Proposition 1 (error-free decryption): *For every $\sigma \in \{0, 1\}$, every choice of the private and public keys, and every choice of b_i's by the encryption algorithm, the ciphertext, x, satisfies $\langle x, u\rangle \in \mathcal{Z} + \frac{\sigma}{2} \pm \frac{1}{n}$.*

Proof (sketch): The case of $\sigma = 0$ is the same as for the original Ajtai-Dwork scheme. The case of $\sigma = 1$ follows from the same arguments, using the fact that $\langle \frac{1}{2}v_{i_1}, u\rangle \in \mathcal{Z} + \frac{1}{2} \pm n^{-7}$. □

[4] Otherwise, we may simply use the identity function for encryption/decryption.

3 Security of the Modified Scheme

To prove the security of the modified scheme, we start by invoking the main result of Ajtai and Dwork [2]:

Theorem 2 [2, Thm 7.1]: *Under Assumption ISVP, it is infeasible to distinguish the encryption of $\sigma = 0$ from a uniformly distributed point in $P(\mathbf{w})$, when given $\mathbf{w}, \mathbf{v}$. (We stress that $\mathbf{w}, \mathbf{v}$ and the encryption of '0' are distributed as described above.)*

Note that this theorem establishes the security (as defined in [3]) of the encryption scheme of Ajtai and Dwork [2], since in that scheme $\sigma = 1$ is encrypted as a uniformly chosen point in $P(\mathbf{w})$. To establish the security of our (modified) encryption scheme (under the same assumption), we need to prove

Theorem 3 (security): *Under Assumption ISVP, it is infeasible to distinguish the encryption of $\sigma = 0$ from the encryption of $\sigma = 1$, when given $\mathbf{w}, \mathbf{v}$ and i_1. (We stress that $\mathbf{w}, \mathbf{v}, i_1$ and the encryptions are distributed as described in the modified scheme.)*

Proof: Recall our notations $\mathbf{w} \stackrel{\text{def}}{=} (w_1, \ldots, w_n)$, $\mathbf{v} \stackrel{\text{def}}{=} (v_1, \ldots, v_m)$ and $\mathbf{e} \stackrel{\text{def}}{=} (\mathbf{w}, \mathbf{v})$. For a bit $\sigma \in \{0, 1\}$, and an encryption key $(\mathbf{e}, i)$, let us denote by $E_{\mathbf{e},i}(\sigma)$ the probabilistic encryption of σ using $(\mathbf{e}, i)$. Also, let us denote by $\Pi_{\mathbf{w}}$ the uniform distribution over $P(\mathbf{w})$. Assuming ISVP, we will show that for both $\sigma = 0$ and $\sigma = 1$, it is infeasible to distinguish $(\mathbf{e}, i, E_{\mathbf{e},i}(\sigma))$ from $(\mathbf{e}, i, \Pi_{\mathbf{w}})$.

First we show that this holds for $\sigma = 0$. Note that this claim is not identical to Theorem 2, as here the distinguisher is given i (for which $\langle v_i, u\rangle \in 2\mathcal{Z}+1 \pm n^{-7}$ holds) as extra information. Still, Theorem 2 does imply the following

Lemma 4 *Under Assumption ISVP, it is infeasible to distinguish $(\mathbf{e}, i, E_{\mathbf{e},i}(0))$ from $(\mathbf{e}, i, \Pi_{\mathbf{w}})$, where $(\mathbf{e}, i)$ are selected as above and $\Pi_{\mathbf{w}}$ is uniformly distributed in $P(\mathbf{w})$.*

Proof. Suppose towards the contradiction that there exists a distinguisher, D, of running-time $t(n)$ and distinguishing gap $\epsilon(n)$ (between $(\mathbf{e}, i, E_{\mathbf{e},i}(0))$ and $(\mathbf{e}, i, \Pi_{\mathbf{w}})$ as in the claim). We construct a new distinguisher, D', which violates Theorem 2. D' works as follows:

input: $\mathbf{e} = (w_1, \ldots, w_n, v_1, \ldots, v_m)$ and x.

preprocessing: Using D, we find an index j which approximately maximizes the distinguishing gap of D on inputs of the form $(\mathbf{e}, j, \cdot)$. This is done by estimating, for every $j = 1, \ldots, m$, the value

$$\text{Prob}[D(\mathbf{e}, j, E_{\mathbf{e},j}(0)) = 1] - \text{Prob}[D(\mathbf{e}, j, \Pi_{\mathbf{w}}) = 1]$$

where the probability is taken over the internal coin tosses of both the encryption algorithm (i.e., choice of b_i's) and D. Invoking D for $\text{poly}(n)/\epsilon(n)^2$ times we may obtain, with overwhelmingly high probability, an approximation of the above upto $\epsilon(n)/4$. Let $\tau \in \{\pm 1\}$ denote the sign of the approximated difference for the best j.

decision: Using j and τ, found in the preprocessing, we invoke D on input $(\mathbf{e}, j, x)$. Let $\sigma \in \{\pm 1\}$ denote the output of D. Then D' outputs $\tau \cdot \sigma$.

Clearly, D' has running time $\mathrm{poly}(n, t(n), \epsilon(n)^{-1})$, which is polynomial in n as long as $t(n)/\epsilon(n)$ is polynomial in n. It is easy to see that

$$|\mathrm{Prob}[D'(\mathbf{e}, E_{\mathbf{e}}(0)) = 1] - \mathrm{Prob}[D'(\mathbf{e}, \Pi_{\mathbf{w}}) = 1]| > \frac{\epsilon(n)}{2} - 2^{-n}$$

(The second term is due to the case where we made some wrong approximation in the preprocessing stage.) Thus, we have a distinguisher violating the conclusion of Theorem 2, and so contradiction follows. □

Using Lemma 4, we easily derive

Lemma 5 *Under Assumption ISVP, it is infeasible to distinguish* $(\mathbf{e}, i, E_{\mathbf{e},i}(1))$ *from* $(\mathbf{e}, i, \Pi_{\mathbf{w}})$*, where* $(\mathbf{e}, i)$ *and* $\Pi_{\mathbf{w}}$ *are as in Lemma 4.*

Proof. Suppose towards the contradiction that there exists a distinguisher, D, of running-time $t(n)$ and distinguishing gap $\epsilon(n)$ (between $(\mathbf{e}, i, E_{\mathbf{e},i}(1))$ and $(\mathbf{e}, i, \Pi_{\mathbf{w}})$ as in the claim). We construct a new distinguisher, D', as follows

input: $\mathbf{e} = (w_1, \ldots, w_n, v_1, \ldots, v_m)$, i and x.
decision: Algorithm D' computes $x' = \left(x - \frac{1}{2}v_i\right) \bmod P(\mathbf{w})$, and outputs $D(x')$.

Observe that $E_{\mathbf{e},i}(0)$ and $E_{\mathbf{e},i}(1) - \frac{1}{2}v_i$ (reduced mod $P(\mathbf{w})$) are identically distributed. Similarly, $\Pi_{\mathbf{w}}$ and $\Pi_{\mathbf{w}} - \frac{1}{2}v_i$ (reduced mod $P(\mathbf{w})$) are identically distributed. Thus, D' distinguishes $(\mathbf{e}, i, E_{\mathbf{e},i}(0))$ from $(\mathbf{e}, i, \Pi_{\mathbf{w}})$, in contradiction to the claim of Lemma 4. The current lemma follows. □

Combining Lemmas 4 and 5, we have established Theorem 3. ■

Comment – An alternative proof of Theorem 3. The security of the encryption scheme in [2] is established via a sequence of reductions. The first reduction assumes an algorithm D which distinguishes between encryptions of 0's and 1's. It then constructs another algorithm D' which distinguishes between sequences of vectors $(\mathbf{w}, \mathbf{v})$ which constitute a public-key, and sequences uniformly distributed points in the big cube B_n (See [2, Lemma 8.1]). On a high level, this is done as follows: Algorithm D' uses the input vectors, $(\mathbf{w}, \mathbf{v})$, to encrypt 0's and 1's as if they constitute a public-key. If D is able to distinguish between encryptions of 0's and 1's, then D' concludes that these vectors indeed constitute a public-key. Otherwise, D concludes that they are just uniformly distributed points.

One can easily verify the argument in [2] holds also for distinguishers of encryptions under our modified scheme. Specifically, one needs to verify that when applying our encryption scheme using m uniformly distributed vectors, the result is distributed almost uniformly in the parallelepiped $P(\mathbf{w})$, regardless of whether a '0' or a '1' was encrypted. □

Acknowledgments

This research was supported by DARPA grant DABT63-96-C-0018.

References

1. Miklos Ajtai. Generating Hard Instances of Lattice Problems. In *28th ACM Symposium on Theory of Computing*, pages 99–108, Philadelphia, 1996.
2. Miklos Ajtai and Cynthia Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence, In *29th ACM Symposium on Theory of Computing*, pages 284-293, 1997.
3. Shafi Goldwasser and Silvio Micali. Probabilistic Encryption, *JCSS*, Vol. 28, No. 2, pages 270–299, 1984.