The Hardness of Hensel Lifting: The Case of RSA and Discrete Logarithm

Dario Catalano, Phong Q. Nguyen, and Jacques Stern

École normale supérieure
Département d'informatique
45 rue d'Ulm, 75230 Paris Cedex 05, France.
{Dario.Catalano,Phong.Nguyen,Jacques.Stern}@ens.fr

Abstract. At ACM CCS '01, Catalano *et al.* proposed a mix of the RSA cryptosystem with the Paillier cryptosystem from Eurocrypt '99. The resulting scheme, which we call RSAP, is a probabilistic cryptosystem which is both semantically secure under an appropriate decisional assumption and as efficient as RSA, but without the homomorphic property of the Paillier scheme. Interestingly, Sakurai and Takagi presented at PKC '02 a proof that the one-wayness of RSAP was equivalent to the RSA assumption. However, we notice in this paper that the above proof is not completely correct (it works only in the case when a perfect oracle - i.e. an oracle that always provides correct answers - is given). We fix the proof by presenting a new proof based on low-dimensional lattices. The new proof, inspired by the work of Sakurai and Takagi, is somewhat related to Hensel lifting and the N-adic decomposition of integer exponentiation. Roughly speaking, we consider the problem of computing $f(x) \bmod M^\ell$ given $f(x) \bmod M$ and an exponent $\ell > 1$. By studying the case $f(x) = x^e$ and M is an RSA-modulus, we deduce that the one-wayness of RSAP is indeed equivalent to the RSA assumption, and we are led to conjecture that the one-wayness of the original Paillier scheme may not be equivalent to the RSA assumption with exponent N. By analogy, we also study the discrete logarithm case, namely when $f(x) = g^x$ and M is a prime, and we show that the corresponding problem is curiously equivalent to the discrete logarithm problem in the subgroup spanned by g.

Keywords: Public-key, RSA, Paillier, Discrete logarithm, Hensel, One-wayness, Lattice.

1 Introduction

Many basic computational problems in number theory can be efficiently solved by first looking at the problem modulo a (small) prime number p and then performing a so-called Hensel lifting, which iteratively transforms solutions modulo p into solutions modulo arbitrary powers of p. This is for instance the case with factorization of univariate integer polynomials, and with integer root finding of univariate integer polynomials (see [1, 5]). The lifting process has been dubbed

Hensel lifting because of the pioneering work of the German mathematician Hensel on p-adic numbers at the end of the 19th century. The p-adic numbers are beyond the scope of this paper, and we refer the interested reader to [8] for more information: Let us just briefly mention that, mathematically speaking, the p-adic numbers are an extension (depending on p) of the field $\mathbb{Q}$ of rational numbers, which is built as a completion of $\mathbb{Q}$ with respect to a specific metric (different from the usual absolute valuation $|x-y|$) related to the decomposition in base p of every positive integer. The link between Hensel lifting and p-adic numbers is natural: Hensel lifting produces solutions modulo increasing powers of p which can be viewed as better and better approximations of some "true" solution, where the quality of the approximation is measured thanks to the specific metric of the p-adic numbers.

In this paper we consider Hensel lifting from a cryptographic perspective. We study the hardness of the general problem of computing $f(x) \bmod M^\ell$ (where ℓ is an integer ≥ 2) given $f(x) \bmod M$, where the function f is implemented as either the RSA function or the Discrete Logarithm function. More precisely, we investigate the following problems:

1. Given an RSA modulus N and the value $x^e \bmod N$ where $0 \leq x < N$, how hard is it to compute $x^e \bmod N^\ell$ (for $\ell > 1$)?
2. Given a prime p, an integer g and the value $g^x \bmod p$ where x is defined modulo the order of g, how hard is it to compute $g^x \bmod p^\ell$ (again for $\ell > 1$)?

Motivation and Previous work. At Eurocrypt '99 Paillier [11] proposed a new cryptosystem based on a novel computational problem: the composite residuosity class problem. The details of the scheme are given below, for now let us highlight the main contributions of Paillier's construction. Given an RSA modulus N, the multiplicative group $\mathbb{Z}_{N^2}^*$ can be partitioned into N equivalence classes according to the following equivalence relation: $a, b \in \mathbb{Z}_{N^2}^*$ are equivalent if and only if the product ab^{-1} is an *N-th residue* modulo N^2, where by N-residue we intend an element $x \in \mathbb{Z}_{N^2}^*$ such that there exists $y \in \mathbb{Z}_{N^2}^*$ satisfying the equation $x \equiv y^N \bmod N^2$.

The composite residuosity class problem is then the problem to determine, on input a random value $w \in \mathbb{Z}_{N^2}^*$ to which class such an element belongs. The one-wayness of Paillier's scheme is provably equivalent to the class problem which turns out to be related but not known to be equivalent to the problem of inverting RSA, when the public encryption exponent is set to N. The semantic security of Paillier's scheme is provably equivalent to a decisional variant of the class problem. Paillier's paper has sparkled a huge amount of research due to its beautiful and original mathematical structure. Moreover the scheme is very attractive for many practical applications because of its *homomorphic* property: given the ciphertexts $c_1 = \mathbf{ENC}(m_1)$ and $c_2 = \mathbf{ENC}(m_2)$, an encryption of $m_1 + m_2$ can easily be obtained by simply multiplying c_1 and c_2.

The main drawback of Paillier's scheme is its cost: encryption and decryption cost respectively two and one modular exponentiations, but all the operations are performed modulo N^2. Moreover the exponents used have all order $\Omega(N)$. To improve the efficiency of the scheme, Catalano *et al.* [4] proposed a mix of Paillier's scheme with the RSA scheme, whose running time is comparable to that of plain RSA, and which is still semantically secure under an appropriate decisional assumption. The new scheme follows from an alternative decryption process for a particular instance of Paillier's scheme, which allows to drastically reduce the size of the encryption exponent. Interestingly enough, even though the modification proposed in [4] only slightly changes the encryption scheme, it deeply influences its mathematical structure. In the following we will refer to the Catalano *et al.* cryptosystem as the *RSA– Paillier* Cryptosystem (RSAP for brevity).

Later, Sakurai and Takagi [12] further studied the properties of the RSA-Paillier scheme and presented a proof that its one-wayness is equivalent to the problem of inverting RSA. Unfortunately, even though the proposed ideas are very appealing, they turn out not to be completely sound from a technical point of view. Specifically they prove that the one-wayness of RSA-Paillier cryptosystem is equivalent to the problem of computing, given a value of the form $r^e \bmod N$, the "lifted" value $r^e \bmod N^2$. Then the proof proceeds by a standard *reductio ad absurdum* argument: they prove that if one has an oracle to efficiently solve the above lifting problem this oracle could be used to construct an efficient algorithm that computes the least significant bit of RSA (which, in turn, is known to be a hard core predicate [2] for the RSA function [7]). However, as we will show in section 3, the argument is flawed, in the sense that the proposed technique works only for the particular case in which the oracle gives a correct answer with probability 1 (and we will note that another result of [12] related to another variant of RSA suffers from the same flaw). Thus the problem of proving the equivalence between the one-wayness of RSA and the one-wayness of RSA-Paillier remains open for the general case in which the provided oracle answers correctly only for a non-negligible fraction of the inputs.
A variant of the Hensel lifting problem was discussed by Takagi [13], who proposed some efficient variants of RSA using N-adic expansion.

Our Results. Our contributions can be summarized as follows. First of all we prove that the one-wayness of the RSA-Paillier function is actually equivalent to that of the RSA function. We then turn our attention to the original Paillier's trapdoor function and we prove the following, somehow surprising, results:

1. Given a random RSA modulus N, computing $r^N \bmod N^2$ from a value $r^N \bmod N$ where $0 \leq r < N$ is as hard as solving the composite residuosity class problem.
2. Given a random RSA modulus N, computing $r^N \bmod N^3$ from a value $r^N \bmod N$ where $0 \leq r < N$ is as hard as inverting RSA when the public exponent is set to N.

In some sense, the above results seem to provide an intuitive separation between the Class assumption, introduced by Paillier, and the RSA assumption. This leads us to conjecture that the one-wayness of the Paillier scheme is not equivalent to the RSA assumption with exponent N.

Our techniques can be generalized to the discrete logarithm function (modulo a prime p) as well, and we prove that, under certain conditions, the problem of computing $g^x \bmod p^\ell$ when $g, p, h = g^x \bmod p$ are given is equivalent to the problem of computing x. More precisely, the order ω of g modulo p is assumed to be prime and publickly known, and the integer ℓ is defined as the unique positive integer such that $g^\omega \not\equiv 1 \pmod{p^\ell}$ and $g^\omega \equiv 1 \pmod{p^{\ell-1}}$.

Road map. The paper is organized as follows. In Section 2 we provide definitions and notations that are useful for the rest of the paper. Then we quickly describe Paillier's cryptosystem and its variant from [4]. Section 3 presents our results for the RSA case. The discrete logarithm case is discussed in Section 4. We conclude the paper with some remarks and directions for future research in Section 5.

2 Preliminaries

Notation (Basically quoted from [4]). In the following we denote by $\mathbb{N}$ the set of natural numbers, by $\mathbb{R}^+$ the set of positive real numbers, by $\mathbb{Z}_N$ the ring of integers mod N, which we identify to the set $\{0, 1 \ldots, N-1\}$, and by $\mathbb{Z}_N^*$ its subset of invertible elements. In particular, we view elements of $\mathbb{Z}_N$ as integers of $\{0, \ldots, N-1\}$: for instance, if $r \in \mathbb{Z}_N$, $r^e \bmod N^2$ denotes the integer r raised to the power e (as an integer and not as an integer mod N), eventually taken modulo N^2. We say that a function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}^+$ is *negligible* iff for every polynomial $P(n)$ there exists a $n_0 \in \mathbb{N}$ s.t. for all $n > n_0$, $\mathsf{negl}(n) \leq 1/P(n)$. We denote by $\mathcal{PRIMES}(k)$ the set of primes of length k. For $a, b \in \mathbb{N}$ we write $a \propto b$ if a is a non zero multiple of b.

If A is a set, then $a \leftarrow A$ indicates the process of selecting a at random and uniformly over A (which in particular assumes that A can be sampled efficiently).

If N is an RSA modulus (i.e. $N = pq$ with p, q primes), then we denote by $RSA[N, e]$ the RSA function with exponent e. In the following we will assume that $RSA[N, e]$ is a one-way function, i.e. that given N of unknown factorization, a public exponent e and $RSA[N, e](x) = x^e \bmod N$, for random x it is infeasible to efficiently compute x. We will refer to this conjecture as the $RSA[N, e]$ *assumption.*

Paillier's Scheme. Let $N = pq$ be an RSA modulus and consider the multiplicative group $\mathbb{Z}_{N^2}^*$. Let g be an element whose order is a multiple of N in $\mathbb{Z}_{N^2}^*$. Paillier [11] defines the following function

$$\mathcal{F}_g : \mathbb{Z}_N^* \times \mathbb{Z}_N \to \mathbb{Z}_{N^2}^*$$

$$\mathcal{F}_g(r, m) = r^N g^m \bmod N^2$$

and proves the following statements:

- The function $\mathcal{F}_g$ is a trapdoor permutation. The trapdoor information is the factorization of N.
- Inverting $\mathcal{F}_g$ is equivalent to inverting $RSA[N, N]$.

By the first property above, once g is fixed, for a given $w \in \mathbb{Z}^*_{N^2}$, there exists a unique pair (m, r) such that $w = r^N g^m \bmod N^2$. We say that m is the *class* of w relative to g, and we indicate this value with $Class_g(w)$. We define the *Computational Composite Residuosity Class Problem* as the problem of computing m when N, g and w are provided. We will assume this to be an intractable problem. More formally, we use the following definition from [3]:

Definition 1. We say that computing the function $Class_g(\cdot)$ is hard if, for every probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function $\mathsf{negl}()$ such that

$$\Pr\left[\begin{array}{l} p, q \leftarrow \mathcal{PRIMES}(n/2);\ \ N = pq; \\ g \leftarrow \mathbb{Z}^*_{N^2} \text{ s.t. } ord(g) \propto N; \\ c \leftarrow \mathbb{Z}_N;\ \ z \leftarrow \mathbb{Z}^*_N;\ \ w = g^c z^N \bmod N^2; \\ \mathcal{A}(N, g, w) = c \end{array}\right] = \mathsf{negl}(n)$$

In his paper Paillier proves that the function $Class$ is random self reducible [2] over $g \in \mathbb{Z}^*_{N^2}$, i.e. that its complexity is independent of the specific base g used.

The RSA-Paillier scheme. Let $N = pq$ be an RSA modulus and consider the multiplicative group $\mathbb{Z}^*_{N^2}$. For a random $e \in \mathbb{Z}_N$ such that $\gcd(e, \lambda(N^2)) = 1$, Catalano *et al.* [4] defined the following function

$$\mathcal{E}_e : \mathbb{Z}^*_N \times \mathbb{Z}_N \longrightarrow \mathbb{Z}^*_{N^2}$$

$$\mathcal{E}_e(r, m) = r^e(1 + mN) \bmod N^2$$

and they proved it is a trapdoor permutation equivalent to $RSA[N, e]$.
To encrypt a message m, one simply chooses a random $r \in \mathbb{Z}^*_N$ and sets $c = (1+mN)r^e \bmod N^2$. From the ciphertext c, anyone knowing the factorization of N can retrieve the message, by first computing $r = \sqrt[e]{c} \bmod N$ and then getting m as $\frac{(c(r^{-1})^e \bmod N^2)-1}{N}$ over the integers.
Notice that, in order for the above decryption procedure to work it is not necessary to assume $\gcd(e, \lambda(N^2)) = 1$. As a matter of fact, one can consider exponents e such that $\gcd(e, \lambda(N)) = 1$.
In this sense by letting $e = N$ we go back to an instance of Paillier's scheme where g is set to $(1 + N)$. For the purposes of this paper, however, we will assume $\gcd(e, \lambda(N^2)) = 1$. The reason for this choice will become clearer in the next section.

3 The RSA Case

We start this section by introducing a new computational problem, which is actually very similar to a problem presented in [12].
Informally, the problem we have in mind can be stated as follows. Assume an RSA modulus N is provided, given $c = r^e \bmod N$ (where $r \leftarrow \{0, \ldots, N-1\}$),

we want to compute the "lifted" value $r^e \bmod N^\ell$ for $\ell > 1$. More formally we define the following function over $\{0, \ldots, N-1\}$:

$$\mathbf{Hensel} - \mathbf{RSA}[N, e, \ell](r^e \bmod N) = r^e \bmod N^l$$

Note that this function is well-defined over $\{0, \ldots, N-1\}$ because the RSA-function is a permutation over $\mathbb{Z}_N$.

It is immediate to see that if the factorization of the modulus is known then one can efficiently compute **Hensel-RSA**. On the other hand, if the factorization of N is not available, we conjecture it is infeasible to compute such a function in probabilistic polynomial time.

Definition 2. We say that computing the function **Hensel-RSA**$[N, e, \ell](r^e \bmod N)$ is hard if, for every probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function $\mathsf{negl}()$ such that

$$\Pr\begin{bmatrix} p, q \leftarrow \mathcal{PRIMES}(n/2); \quad N = pq; \\ r \leftarrow \{0, \ldots, N-1\}; \quad w = r^e \bmod N; \\ \mathcal{A}(N, e, w, \ell) = r^e \bmod N^\ell \end{bmatrix} = \mathsf{negl}(n)$$

In the next lemma (originally presented, in a slightly different form, in [12]) we make explicit the relation existing between the problem of computing the function Hensel-RSA and the one-wayness of the RSA-Paillier scheme.
The proof is straighforward and is left to the reader.

Lemma 1. *Given an RSA modulus N and a public exponent e, the RSA-Paillier function is one-way if and only if Hensel-RSA[N,e,2] is hard.*

Now, on top of Lemma 1, we prove that the one-wayness of RSA-Paillier is equivalent to the one-wayness of RSA, by showing that the problem of computing Hensel-RSA, with parameters N, e and 2, on input $r^e \bmod N$ and the one of computing r from $r^e \bmod N$ are computationally equivalent. Observe that assuming that Hensel-RSA$[N, e, 2]$ is hard implicitly implies that the RSA$[N, e]$ assumption must hold. Consequently, we will focus on proving that the inverse direction also holds, i.e. that under the RSA$[N, e]$ assumption Hensel-RSA$[N, e, 2]$ is hard.

3.1 A flawed solution

In this paragraph we discuss the approach followed by Sakurai and Takagi [12, Theorem 2], and we show why it is incorrect.
As already sketched in the introduction, they propose the following strategy: assume, for the sake of contradiction, that one has an oracle $\mathcal{O}$ that, on input $r^e \bmod N$, computes $r^e \bmod N^2$ with some non negligible probability of success ε. Then, on input a random RSA ciphertext $r^e \bmod N$, the basic idea of their proof is to use such an oracle to compute the least significant bit of r with some non-negligible advantage, and then apply the bit-security result of [7]. They implement this idea as follows:

1. Run $\mathcal{O}(r^e \bmod N)$ and obtain $b_0 + b_1 N = r^e \bmod N^2$.
2. Run $\mathcal{O}((2^{-1}r)^e \bmod N)$ and obtain $a_0 + a_1 N = (2^{-1}r)^e \bmod N^2$.
3. Return 1 as the *lsb* of r if $a_0 + a_1 N = 2^{-e}(b_0 + b_1 N) \bmod N^2$ holds and 0 otherwise.

Finally they claim that the success probability of the above algorithm is ε^2. However this is not true. As a matter of fact in order for such an estimate to be correct it is crucial to query the oracle on *random* and *independently generated* inputs. Here, on the contrary, the two inputs are clearly not independently sampled. Thus it is not possible to bound by ε^2 the probability of success of the algorithm[1]. By the way, exactly the same mistake appears in another part of the paper [12], more precisely in the proof of [12, Theorem 6], related to the one-wayness of another class of probabilistic variants of RSA.

Furthermore, we note that even if the proof was correct, the reduction would be rather inefficient in terms of oracle calls. Indeed, the reduction makes two oracle calls to obtain only one bit of information on r, which implies that to completely recover r, one has to make at least $2 \log N$ oracle calls. And one also has to use the reduction of the bit-security result of [7].

3.2 Our Solution

With the next theorem we propose a general result connecting the difficulty of computing the Hensel-RSA function with the hardness of inverting RSA. Specifically we prove that, given a public exponent of the form $e = fN^\ell$ (for constants $\ell \geq 0$ and $f > 0$ such that $\gcd(f, \lambda(N^2)) = 1$), Hensel-RSA$[N, e, \ell + 2]$ is hard if and only if RSA$[N, e]$ is hard. Note that any valid RSA public encryption exponent e can be written in the form $e = fN^\ell$ (where $\gcd(f, \lambda(N^2)) = 1$), unless $\gcd(e, N)$ is a non-trivial factor of N, in which case the public exponent e would disclose the RSA private key. As already mentioned our proof will focus on showing that under the RSA$[N, e]$ assumption, Hensel-RSA$[N, e, \ell + 2]$ is hard. Interestingly, our reduction only calls the oracle twice, as opposed to at least $2 \log N$ for the (flawed) one proposed by [12].

Theorem 1. *Given an integer N and an integer e of the form $e = fN^\ell$ where f is coprime with $\lambda(N^2)$ and $\ell \geq 0$, then Hensel-RSA$[N, e, \ell + 2]$ is hard if and only if the RSA$[N, e]$ assumption holds.*

Proof. Assume, for the sake of contradiction, that Hensel-RSA$[N, e, \ell + 2]$ is not hard. This means that there exists an oracle $\mathcal{O}$ that, on input a random challenge $w = r^e \bmod N$, computes $r^e \bmod N^{\ell+2}$ with some non-negligible probability ε. Here we will show how to use this oracle to construct a probabilistic polynomial time algorithm $\mathcal{I}$ that succesfully inverts RSA with a polynomially related probability.

[1] For example it may very well happen that the non-negligible set of inputs for which the oracle answers correctly does not contain any couple of the form $(r^e \bmod N, (2^{-1}r)^e \bmod N)$, and, in such a case, the success probability of the algorithm would be 0.

Assume that we are given as input a random element $w = r^e \bmod N$: our goal is to compute r. We start by choosing a random a uniformly in $\mathbb{Z}_N^*$. We then call the oracle $\mathcal{O}$ twice, on inputs w and $(a^e w) \bmod N$. Since the queries w and $(a^e w) \bmod N$ are independent and uniformly distributed over $\mathbb{Z}_N$ (by definition of r, w and a), we obtain with probability ε^2 the integers $r^e \bmod N^{\ell+2}$ and $\mu^e \bmod N^{\ell+2}$ where μ is defined by $\mu = ar \bmod N$.

We may assume that $r \in \mathbb{Z}_N^*$, otherwise either $r = 0$ or we are able to factor N. Then μ is invertible modulo $N^{\ell+2}$, and there therefore exists $z \in \mathbb{Z}_{N^{\ell+1}}$ such that :

$$ar \equiv \mu(1 + zN) \pmod{N^{\ell+2}} \tag{1}$$

Raising to the power $e = fN^\ell$, we obtain :

$$a^e r^e \equiv \mu^e (1 + zfN^{\ell+1}) \pmod{N^{\ell+2}}.$$

In this congruence, we know a, $r^e \bmod N^{\ell+2}$ and $\mu^e \bmod N^{\ell+2}$: we can thus compute zf modulo N. Since f is coprime with N, we derive $z_0 = z \bmod N$. Taking equation (1) modulo N^2, we obtain:

$$ar \equiv \mu(1 + z_0 N) \pmod{N^2}, \tag{2}$$

where only r and μ are unknowns both in $\{1, \ldots, N-1\}$.

To complete the proof, we solve this linear congruence by a lattice reduction argument (see for instance the survey [10] for references on lattice theory). Consider indeed the following set

$$L = \{(R, U) \in \mathbb{Z}^2 : aR \equiv U(1 + z_0 N) \pmod{N^2}\}.$$

Since L is a subgroup of $\mathbb{Z}^2$, L is a lattice, whose dimension is obviously equal to two. The vector (r, μ) belongs to L and to $[1, N-1]^2$. Therefore $L \cap [1, N-1]^2$ is not empty. A classical lattice reduction result (which can be viewed as a particular case of integer programming in fixed dimension, see [9]) then states that one can compute a vector $(r', \mu') \in L \cap [1, N-1]^2$ in time polynomial in $\log N$ (because one obviously knows a basis of L whose size is polynomial in $\log N$). Because (r, μ) and (r', μ') both belong to L, equation (2) implies :

$$r\mu' \equiv r'\mu \pmod{N^2}.$$

Since r, μ, r', μ' all lie in $[1, N-1]$, the congruence is in fact an equality over $\mathbb{Z}$: $r\mu' = r'\mu$. ¿From r' and μ', we can therefore compute the integers r and μ up to a multiplicative factor, namely $\gcd(r, \mu)$.

We now show that with overwhelming probability, this gcd will be sufficiently small that it can be exhaustively searched in polynomial time. To see this, notice that the number of pairs $(\alpha, \beta) \in [0, N-1]^2$ which have a common divisor d is $O(N^2/d^2)$ as N grows, therefore, for any B, the number of pairs $(\alpha, \beta) \in [0, N-1]^2$ which have a a gcd $> B$ is at most $O(\sum_{d>B} N^2/d^2) = O(N^2/B)$. Since μ and r are both uniformly distributed over $\mathbb{Z}_N$, the probability that $\gcd(\mu, r) \geq (\log N)/\varepsilon$ is $O(\varepsilon^2/\log N)$ by taking $B = (\log N)/\varepsilon^2$. Finally, we proved that with probability at least $\varepsilon^2 - O(\varepsilon^2/\log N) = \varepsilon^2(1 - o(1))$ over the choice of (a, r), we can compute in polynomial time r an μ up to the factor $\gcd(r, \mu)$ which is $\leq (\log N)/\varepsilon^2$. Thus, we can compute r in time polynomial in $\log N$ and $1/\varepsilon$, thanks to an exhaustive search over $\gcd(r, \mu)$, since the value of r can be checked with $w = r^e \bmod N$.

□

As an immediate consequence of Theorem 1 and Lemma 1, we obtain:

Corollary 1. *Given an RSA modulus N together with a public exponent e such that $gcd(e, \lambda(N^2)) = 1$, the RSAP encryption function is one-way if and only if RSA[N, e] is a one-way function.*

Observe that, by setting $f = \ell = 1$ in the parameters of Theorem 1, we get that the hardness of Hensel-RSA$[N, N, 3]$ is actually equivalent to that of RSA$[N, N]$. To complete the picture, with the next theorem we make explicit the relation existing between the one-wayness of Paillier's encryption function and the problem of computing Hensel-RSA with parameters $N, N, 2$.

Theorem 2. *Given an RSA modulus N, then Hensel-RSA[N,N,2] is hard if and only if $Class_g$ is hard.*

Proof. Since, for all g such that $ord(g) \propto N$, all the intances of $Class_g(\cdot)$ are computationally equivalent, we will prove the theorem for the case in which $g = 1 + N$ (note that $1 + N$ has order N in $\mathbb{Z}^*_{N^2}$).
First assume that a random ciphertext $c = (1 + mN)r^N \bmod N^2$ is given. Our goal is to compute m using an oracle that, when receiving an input of the form $y^N \bmod N$ returns as output the value $y^N \bmod N^2$, with probability ε (non negligible). Thus when the oracle is given the value $c \bmod N$, it will answer $(\sqrt[N]{c \bmod N})^N \bmod N^2$ with probability ε. Note that this value corresponds to $r^N \bmod N^2$ (Observe that this is true even in the case in which r is greater than N). From $r^N \bmod N^2$ and $r^N \bmod N$ it is easy to compute m.

Conversely, assume we are given an oracle than on input a random $c \in \mathbb{Z}^*_{N^2}$ computes the class of c with respect to the base $(1 + N)$ (again we denote by ε the probability of success of the oracle). Now we would like to compute, for a random challenge $r^N \bmod N$, the corresponding $r^N \bmod N^2$, using the provided oracle.
Let us consider the value

$$d = (r^N \bmod N) + kN$$

Where $k \leftarrow \mathbb{Z}_N$. Note that, since $r^N \bmod N$ is uniformly distributed in $\mathbb{Z}^*_N$ and, being $\mathbb{Z}^*_{N^2}$ isomorphic to $\mathbb{Z}^*_N \times \mathbb{Z}_N$ [11], d is uniformly distributed in $\mathbb{Z}^*_{N^2}$ and can be written (univoquely) as

$$d = r^N(1 + mN) \bmod N^2$$

extracting m from d (via the given oracle) thus leads to compute $r^N \bmod N^2$.

□

Remark 1. At PKC'01 Damgård and Jurik [6] presented a generalized (and still homomorphic) version of Paillier's basic cryptosystem in which the expansion factor is reduced and the block length of the scheme may be changed without altering the public key. Moreover they show that such a variant is as secure as Paillier's construction.
The result presented in Theorem 2 above, can be generalized to connect the one-wayness of the Damgård-Jurik construction and the hardness of Hensel-RSA with appropriate parameters. Details are deferred to the final version of this paper.

4 The Discrete Log Case

In this section we extend our results to the discrete logarithm function. Let $\omega \in \mathcal{PRIMES}(k)$ and $g \in \mathbb{Z}_p$ an element of order ω in $\mathbb{Z}_p^*$, where p is a prime (note that ω must divide $p-1$). We introduce the following, computational, problem: Given p, g, ω and $h = g^x \bmod p$, compute $h' = g^x \bmod p^\ell$.

Formally we define the function:

$$\mathbf{Hensel}-\mathbf{Dlog}[p, g, \ell](g^x \bmod p) = g^x \bmod p^\ell$$

We will assume this function to be not computable in probabilistic polynomial time.

Definition 3. Let $n(\cdot)$ be a polynomial, we say that computing the function **Hensel-Dlog**$[p, g, \ell](g^x \bmod p)$ is hard if, for every probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function $\mathsf{negl}()$ such that

$$\Pr\begin{bmatrix} \omega \leftarrow \mathcal{PRIMES}(k) \\ p \leftarrow \mathcal{PRIMES}(n(k)) \; s.t. \; p-1 \propto \omega \\ g \leftarrow \mathbb{Z}_p^* \;\; s.t. \;\; ord(g) = \omega \\ x \leftarrow \mathbb{Z}_\omega; \;\; h = g^x \bmod p; \\ \mathcal{A}(N, g, h, \omega, \ell) = g^x \bmod p^\ell \end{bmatrix} = \mathsf{negl}(k)$$

With the following theorem we relate the hardness of the function Hensel-Dlog, to the hardness of the Discrete Logarithm function.

Theorem 3. *Let ω be a k-bit random prime and p, such that $p-1 \propto \omega$, a prime whose size is polynomially related with k. Given g of order ω in $\mathbb{Z}_p^*$, p and ω, Hensel-Dlog$[p, g, \ell]$ is hard if and only if the discrete logarithm in the subgroup spanned by g in $\mathbb{Z}_p^*$ is a one-way function, where ℓ is defined as the unique positive integer such that $g^\omega \not\equiv 1 \pmod{p^\ell}$ and $g^\omega \equiv 1 \pmod{p^{\ell-1}}$.*

Proof. We follow the proof of Theorem 1. Assume, for the sake of contradiction, that Hensel-Dlog$[p, g, \ell]$ is not hard. This means that there exists an oracle $\mathcal{O}$ that, on input a random challenge $h = g^x \bmod p$ uniformly distributed over the subgroup spanned by g, computes $g^x \bmod p^\ell$ with some non-negligible probability ε. Here we will show how to use this oracle to construct a probabilistic polynomial time algorithm $\mathcal{I}$ that succesfully extracts discrete logarithms in base g modulo p with a polynomially related probability.

We are given as input a random element $h = g^x \bmod p$: our goal is to compute x. We start by choosing a random a uniformly in $\mathbb{Z}_\omega^*$. We then call the oracle $\mathcal{O}$ twice, on inputs h and $h^a \bmod p$. Since the queries h and $h^a \bmod p$ are independent and uniformly distributed over the subgroup spanned by g (because ω is prime), we obtain with probability ε^2 the integers $g^x \bmod p^\ell$ and $g^\mu \bmod p^\ell$ where μ is defined by $\mu = ax \bmod \omega$.

Because $0 \leq a < \omega$ and $0 \leq x < \omega$, there exists an integer r such that $ax = \mu + r\omega$ and $0 \leq r < \omega$. We obtain :

$$g^{ax} \equiv g^\mu g^{r\omega} \pmod{p^\ell}.$$

¿From $g^x \bmod p^\ell$ and $g^\mu \bmod p^\ell$, we therefore derive $g^{r\omega} \bmod p^\ell$. Besides, ℓ is such that $g^\omega \not\equiv 1 \pmod{p^\ell}$ and $g^\omega \equiv 1 \pmod{p^{\ell-1}}$. One can therefore compute an integer $z \in \mathbb{Z}_p$ such that :

$$g^\omega \equiv 1 + p^{\ell-1} z \pmod{p^\ell}.$$

Then :

$$g^{r\omega} \equiv 1 + p^{\ell-1} rz \pmod{p^\ell}.$$

Hence, we can compute $r \bmod p$, and since $0 \leq r < \omega < p$, we know r exactly.

Now, in the equation $ax = \mu + r\omega$, only the integers $0 \leq x < \omega$ and $0 \leq \mu < \omega$ are unknown. We have:

$$\frac{r\omega}{a} \leq x < \frac{(r+1)\omega}{a}.$$

We thus obtain an interval of length ω/a containing x. We now show that with overwhelming probability, this interval will be sufficiently short to be exhaustively searched.

Indeed, with probability at least $1 - \varepsilon^2/\log\omega$ over the choice of a, we have $a \geq \varepsilon^2\omega/\log\omega$, which implies that $0 \leq \omega/a \leq (\log\omega)/\varepsilon^2$. It follows that with probability at least $\varepsilon^2 - \varepsilon^2/\log\omega = \varepsilon^2(1 - 1/\log\omega)$ over the choice of (r, a), we have $0 \leq \omega/a \leq (\log\omega)/\varepsilon^2$ and the outputs of the two oracle calls are correct. Then, by exhaustive search over at most $(\log\omega)/\varepsilon^2 \leq k/\varepsilon^2$ possibilities, we obtain x (the correct value can be recognized by the congruence $h \equiv g^x \pmod{p}$). Thus, with probability at least $\varepsilon^2(1 - 1/\log\omega) = \varepsilon^2(1 - o(1))$ (as k grows), we can compute x in time polynomial in k and $1/\varepsilon$. □

5 Conclusions

In this paper we introduced two new functions and we studied their computational properties by relating them to the problems of inverting RSA and computing discrete logarithms. Moreover we formally proved that the one-wayness of the RSA-Paillier scheme [4] is actually equivalent to that of RSA, thus fixing an incorrect proof recently proposed by Sakurai and Takagi [12].

There are several open questions arising from this research. It would be nice to know whether it is possible to further extend our results to discover the exact relation existing between Paillier's Class assumption and RSA$[N, N]$. Another intriguing direction may be to try to improve our understanding about the hardness of the Hensel-Dlog function, and to find cryptographic applications. We proved that if one can compute $g^x \bmod p^\ell$ from $g^x \bmod p$ (in the case when $g^\omega \equiv 1 \bmod p$ but $g^\omega \not\equiv 1 \bmod p^\ell$) then one could compute the discrete logarithm function over the subgroup spanned by g. This implies that computing $g^x \bmod p^{\ell-1}$ from $g^x \bmod p$ may be potentially easier than computing discrete logarithms in the subgroup spanned by g.

Acknowledgements. We would like to thank Igor Shparlinski for helpful discussions.

References

1. E. Bach and J. Shallit Algorithmic Number Theory, Vol.1: Efficient Algorithms. MIT Press, 1996.
2. M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. *SIAM Journal on Computing*, Vol. 13, No. 4:850-864, 1984.
3. D. Catalano, R. Gennaro and N. Howgrave-Graham. The Bit Security of Paillier's Encryption Scheme and its Applications. In *Advances in Cryptology - Eurocrypt '01*. LNCS vol.2045, Springer, 2001, pages 229-243.
4. D. Catalano, R. Gennaro, N. Howgrave-Graham and P. Q. Nguyen. Paillier's Cryptosystem Revisited. In *8th ACM Conference on Computer and Communication Security* pp.206-214, 2001.
5. H. Cohen. A Course in Computational Algebraic Number Theory. Graduate Texts in Mathematics, Vol 138, Springer, 1996.
6. I. Damgård and M. Jurik. A Generalization, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In *Public key Cryptography*, LNCS vol. 1992, 2001, pages 119-136.
7. R. Fischlin and C.P. Schnorr. Stronger Security Proofs for RSA and Rabin Bits. *J. of Cryptology*, 13(2):221–244, Spring 2000.
8. F. Gouvêa. p-adic numbers. Universitext, Springer, 1997.
9. M. Grötschel, L. Lovász and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1993.
10. P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Proc. of CALC '01*, volume 2146 of *LNCS*, Springer-Verlag, 2001.
11. P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology - Eurocrypt '99*, LNCS vol. 1592, Springer, 1997, pages 223-238.
12. K. Sakurai and T. Takagi. New Semantically Secure Public-Key Cryptosystems from the RSA Primitive In *Public key Cryptography*, LNCS vol. 2274, 2002, pages 1-16.
13. T. Takagi Fast RSA type Cryptosystems Using n-adic Expansion. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, Springer-Verlag, 1997.