A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and its Applications

Emmanuel Bresson[1], Dario Catalano[2], and David Pointcheval[2]

[1] Cryptology Department, CELAR, 35174 Bruz Cedex, France,
Emmanuel.Bresson@polytechnique.org
[2] École normale supérieure, Laboratoire d'Informatique,
45 rue d'Ulm, 75230 Paris Cedex 05, France,
{dario.catalano,david.pointcheval}@ens.fr

Abstract. At Eurocrypt '02 Cramer and Shoup [7] proposed a general paradigm to construct practical public-key cryptosystems secure against adaptive chosen-ciphertext attacks as well as several concrete examples. Among the others they presented a variant of Paillier's [21] scheme achieving such a strong security requirement and for which two, independent, decryption mechanisms are allowed. In this paper we revisit such scheme and show that by considering a different subgroup, one can obtain a different scheme (whose security can be proved with respect to a different mathematical assumption) that allows for interesting applications. In particular we show how to construct a perfectly hiding commitment schemes that allows for an on-line / off-line efficiency tradeoff. The scheme is computationally binding under the assumption that factoring is hard, thus improving on the previous construction by Catalano *et al.* [5] whose binding property was based on the assumption that inverting RSA$[N, N]$ (i.e. RSA with the public exponent set to N) is hard.

1 Introduction

Secrecy of communication is clearly one of the most important goal of cryptography, therefore many secret-key and public-key cryptosystems have been proposed to solve it. It is furthermore widely admitted that the main security notion to be achieved is the semantic security [11] (a.k.a. indistinguishability of ciphertexts). Actually, a semantically secure public-key cryptosystem is not only important for secret communications, but it is also a fundamental primitive for many more complex protocols such as electronic voting, electronic auctions and secret evaluation of functions to cite some of them. However, having a "secure" cryptosystem is in general not sufficient to construct efficient solution for the above mentioned problems. In general more specific properties, such as a kind of malleability, or even homomorphic relations, are very useful to obtain practical constructions.

Roughly speaking, a public-key encryption scheme allows someone to encrypt a message for a unique recipient, the one who owns the corresponding private key (a.k.a. decryption key). But in practice, there is often a natural hierarchy, either for security or for safety reasons: the head of a group may want to be able to read any message sent to the members of the group, people may want to be able to recover the plaintexts even if they loose their private key. Therefore, it is highly desirable to provide schemes that enable to deal with intermediate scenarios, in which users are allowed to process their own data, but not those of other users.

Moreover, in practice, there are many situations on which we need more than a plain encryption function. In particular, it is often useful to have a provably secure encryption primitive that allows to perform some computation on the plaintexts without revealing them explicitly.

In this paper we propose a simple cryptosystem achieving both the above goals.

1.1 Related Work

El Gamal's scheme [8] was the first scheme based on the discrete logarithm problem, more precisely on the Diffie-Hellman problem. Furthermore, it enjoys a multiplicative homomorphic property (as the RSA cryptosystem [22]) by which one can easily obtain an encryption of $m_1 \cdot m_2$ by simply multiplying encryptions of m_1 and m_2. This feature, however, is not very convenient for practical purposes. Indeed for many applications one may desire an efficient cryptosystem equipped with an *additive homomorphic* property, i.e. such that from encryptions of m_1 and m_2 one can obtain the encryption of m_1+m_2 by simply combining the corresponding ciphertexts. The first additively homomorphic cryptosystem was proposed by Goldwasser and Micali [11] in their seminal paper on probabilistic encryption. The Goldwasser-Micali's scheme is based on quadratic residues. Given an RSA modulus N, to encrypt a bit b one chooses a pseudo-square $g \in \mathbb{Z}_N^*$ (i.e. a non quadratic residue having Jacobi symbol equal to 1) and computes $g^b r^2 \bmod N$ for random $r \in \mathbb{Z}_N^*$. The security of the cryptosystem is based on the so-called *quadratic residuosity assumption*. To improve on bandwidth Benaloh and Fisher [1, 6] proposed a generalization of Goldwasser-Micali cryptosystem based on the *prime residuosity assumption*. The basic idea of their scheme is to consider $\mathbb{Z}_e$ (instead of $\mathbb{Z}_2$) as underlying message space (where e is a small prime such that it divides $\phi(N)$ but e^2 does not). To encrypt a message m one then sets $g^m r^e \bmod N$, where, in this case, g is a non e-*residue* (i.e. an element whose order is a multiple of e). The main drawback of this scheme however is that decryption is rather inefficient as it requires some kind of exhaustive search to recover the message (and thus it imposes e to be very small). A more efficient variant of the Benaloh-Fischer scheme was proposed in 1998 by Naccache and Stern [18], who observed that in order to make the decryption procedure faster one can consider a value e that is not prime but instead obtained as the product of several small primes $e_1, \ldots, e_n$ such that e divides $\phi(N)$ but none of the e_i^2's does.

At the same time a completely different approach was proposed by Okamoto and Uchiyama [20] who suggested to work on the group $\mathbb{Z}_N^*$ where $N = p^2 q$. The resulting scheme is very efficient and allows for a pretty large bandwidth (they use $\mathbb{Z}_p$ as underlying message space), but unfortunately it is vulnerable to a simple chosen-ciphertext attack that permits to factor the modulus.

More recently Paillier [21] proposed a generalization of the Okamoto-Uchiyama cryptosystem that works in the multiplicative group $\mathbb{Z}_{N^2}^*$ and allows to consider N as a standard RSA modulus. Details of Paillier's scheme are presented below, but its basic idea is that to encrypt a message $m \in \mathbb{Z}_N$ one selects a random value y in $\mathbb{Z}_N^*$ and sets the ciphertext as $g^m y^N \bmod N^2$ (where g is an element whose order is a multiple of N in $\mathbb{Z}_{N^2}^*$). The semantic security of the scheme is proved with respect to the *decisional N-th residuosity assumption*: given a random value $x \in \mathbb{Z}_N^*$ it is computationally infeasible to decide if there exists another element z in $\mathbb{Z}_{N^2}^*$ such that $x \equiv z^N \bmod N^2$. Paillier's scheme is more efficient (in terms of bandwidth) than all previously described schemes, moreover no adaptive chosen ciphertext attack recovering the factorization of the modulus is known. For these reasons Paillier's proposal is the best solution presented so far in terms of additively homomorphic cryptosystems.

At Eurocrypt'02 Cramer and Shoup [7] proposed a very general and beautiful methodology to obtain security against adaptive chosen-ciphertext attacks from a certain class of cryptosystems with some well-defined algebraic properties. In particular they showed how to modify Paillier's original scheme in order to achieve such a strong security goal. The resulting variant, moreover, allows for a double decryption mechanism: one can decrypt either if the factorization of the modulus is available or if some specific discrete logarithm is known.

1.2 Our Contribution

As described above all the additively homomorphic cryptosystems known so far base their security on some assumption relying on deciding residuosity.

In this paper we further investigate on the basic Cramer-Shoup variant and show that by slightly modifying the underlying structure of the scheme we obtain a new cryptosystem that allows for some more useful applications, maintaining, at the same time, all the "good" properties and with security based on a different (non residuosity-related) decisional assumption[3]. Our new public-key encryption scheme, as the proposal in [7] allows for a double decryption mechanism based either on the factorization of the modulus, or on the knowledge of a discrete logarithm. The former trapdoor can be seen as the *master* one, while the latter is a *local* one: the knowledge of a discrete logarithm helps to decrypt ciphertexts which have been encrypted with a *specific* key only, while the factorization of the modulus helps to decrypt any ciphertext, whatever the key is (as long as the underlying modular group remains the same). The basic version

[3] Here, by *non-residuosity related assumption*, we mean a *decisional assumption* which claims something different from the intractability of deciding memberships in a high-residues set.

of our scheme enjoys an additive homomorphic property (similarly to the Paillier's scheme [21]). Furthermore, it is semantically secure in the standard model, based on the decisional Diffie-Hellman assumption modulo a square composite number. Thus our proposal is the first additively homomorphic cryptosystem that can be proved semantically secure with respect to a non residuosity-related decisional assumption.

We emphasize that by applying the Cramer-Shoup [7] general methodology, our scheme can be proved secure against adaptive chosen-ciphertext attacks in the standard model.

Interestingly enough, given the master key, a kind of gap group [19] appears in which the computational Diffie-Hellman problem is hard, while the corresponding decisional problem turns out to be easy — thanks to the easiness of computing the partial discrete logarithm problem (see below). This is the first gap group structure known not based on elliptic curves and pairings.

As an additional result we show how to construct a new, efficient, perfectly hiding / computationally binding commitment scheme based on factoring. A useful property of such a commitment scheme is that it allows for an on-line/off-line efficiency trade-off, by which, one may perform the most expensive part of the work, *before* knowing the message to commit to. To our knowledge no other trapdoor commitment scheme with this property, based on factoring, is known.

1.3 Organization of the Paper

2 Preliminaries

2.1 Definitions and Notations

Let $N = pq$ be a safe-prime modulus, meaning with this that p and q are primes of the form $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also primes. In the remaining of this paper, we denote by $\mathcal{SP}(\ell)$ the sets of safe prime numbers of length ℓ. We consider $\mathbb{G} = QR_{N^2}$ the cyclic group of quadratic residues modulo N^2. We have $\mathrm{ord}(\mathbb{G}) = \lambda(N^2)/2 = pp'qq' = N\lambda(N)/2$, with $\lambda(N) = 2p'q'$. The maximal order of an element in this group is $N\lambda(N)/2$, and every element of order N is of the form $\alpha = (1 + kN)$.

The latter statement is not so trivial, but it will be very useful rewritten as follows: there are exactly N elements of order N in $\mathbb{Z}^\star_{N^2}$, and they are all of the form $\alpha = 1 + kN$. Furthermore, since N is odd, if one denotes by t the inverse of 2 modulo N:

$$\alpha = 1 + kN = (1 + tkN)^2 \bmod N^2.$$

Therefore, they are all in $\mathbb{G}$ too.

2.2 The Partial Discrete Logarithm Problem

Let g be an element of maximal order in $\mathbb{G}$. For simplicity, we assume that $g^{\lambda(N)} \bmod N^2 = (1 + N) \bmod N^2$, that is $k = 1$. Given g and $h = g^a \bmod N^2$

(for some $a \in [1, \mathrm{ord}(\mathbb{G})]$), Paillier [21] defined the *Partial Discrete Logarithm Problem* as the computational problem of computing $a \bmod N$. We assume this problem is difficult (without the factorization of the modulus), as stated in the following assumption.

Assumption 1 (Partial Discrete Logarithm over $\mathbb{Z}^{\star}_{N^2}$). *For every probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function* negl() *such that for sufficiently large ℓ*

$$\Pr\left[\mathcal{A}(N,g,h) = a \bmod N \;\middle|\; \begin{array}{c} p,q \leftarrow \mathcal{SP}(\ell/2);\;\; N = pq; \\ g \leftarrow \mathbb{G};\;\; a \leftarrow [1, ord(\mathbb{G})]; \\ h = g^a \bmod N^2; \end{array}\right] = \mathsf{negl}(\ell).$$

Moreover Paillier proved that, when the factorization of the modulus is available, such a problem is efficiently solvable.

Theorem 2 (See [21]). *Let N be a composite modulus product of two large primes. Let $\mathbb{G}$ be the cyclic group of quadratic residues modulo N^2. The Partial Discrete Logarithm problem (in $\mathbb{G}$) cannot be harder than factoring.*

Proof. It is easy to see that we can solve the PDL problem if the factorization of N is provided, by using the following algorithm,

1. Compute $C = h^{\lambda(N)} \bmod N^2 = (1+N)^a \bmod N^2 = (1+aN) \bmod N^2$;
2. Return the integer $(C - 1 \bmod N^2)/N$.

$\square$

2.3 Details of Paillier's Cryptosystem

Let $N = pq$ be an RSA modulus and g an element having order αN ($\alpha \geq 1$) in the multiplicative group $\mathbb{Z}^{*}_{N^2}$. To encrypt a message $m \in \mathbb{Z}_N$ Paillier proposed the following mechanism

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

for some random $y \in \mathbb{Z}^{*}_{N}$ and he proved that:

- $\mathcal{P}_g$ is a bijection between $\mathbb{Z}_N \times \mathbb{Z}^{*}_{N}$ and $\mathbb{Z}^{*}_{N^2}$.
- $\mathcal{P}_g$ is a trapdoor function equivalent to $RSA[N,N]$.
- The above encryption scheme is semantically secure against chosen-plaintext attack under the N-residuosity assumption (see [21] for details).

Since $\mathcal{P}_g$ is a bijection, given g, for an element $w \in \mathbb{Z}^{\star}_{N^2}$ there exists an unique pair $(c,z) \in \mathbb{Z}_N \times \mathbb{Z}^{*}_{N}$ such that $w = g^c z^N \bmod N^2$. We say that c is the *class* of w relative to g. Informally, (see [21] for more details) Paillier defined the *Computational Composite Residuosity Class Problem* as the problem of computing c given w and assumed that it is hard to solve.

2.4 The "lite" Cramer-Shoup variant

Let N be a product of two safe primes p and q and g an element of order $\lambda(N)$ in $\mathbb{Z}_{N^2}^*$. Such a g can be found by randomly selecting a $\mu \in \mathbb{Z}_{N^2}^\star$ and setting $g = -\mu^{2N}$. It is not hard to show that this results in a generator with overwhelming probability (see [7] for more details). Then we produce the remaining part of the public key h as follows. Randomly choose a secret key $z \in [0, N^2/2]$ and set $h = g^z \bmod N^2$. (Note that for the purposes of this paper, we are considering a *very* simplified version of the Cramer-Shoup scheme, achieving semantic security *only* with respect to a passive adversary. The reader is referred to [7] for the complete solution achieving full security properties).

To encrypt a message $m \in \mathbb{Z}_N$ one chooses a random value $r \in [0, N/4]$ and computes the ciphertext (A, B) where $A = g^r \bmod N^2$ and $B = h^r(1 + mN) \bmod N^2$.

Conversely to decrypt a ciphertext (A, B) two methods are possible: either by computing $(1+mN)$ as $B/A^z \bmod N^2$ or by using the decryption procedure described by Paillier [21] for his scheme. Note that for this second mechanism to work, knowing the value of B is sufficient. Indeed m can be retrieved from $B = h^r(1+mN) \bmod N^2$ as follows. We denote by π the inverse of $\lambda(N) \bmod N$ (note that $\gcd(N, \lambda(N)) = 1$):

$$m = \frac{B^{\lambda(N)} - 1 \bmod N^2}{N} \cdot \pi \pmod N \quad \text{since} \quad B^{\lambda(N)} = 1 + m\lambda(N)N$$

2.5 The Decisional Diffie-Hellman Problem over $\mathbb{Z}_{N^2}^\star$

Informally speaking, the Decisional Diffie-Hellman Problem consists, when given two random Diffie-Hellman "public keys" $A = g^a$ and $B = g^b$, in distinguishing the resulting shared key g^{ab} from a random value (see [11] for the definition of computational indistinguishability). Of course, this is to be done without possessing neither any secret keys a, b nor the factorization of the modulus.

We thus state the *Decisional Diffie-Hellman Assumption* (DDH) over a squared composite modulus of the form $N = pq$.

Assumption 3 (DDH Assumption over $\mathbb{Z}_{N^2}^\star$). *For every probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}()$ *such that for sufficiently large ℓ*

$$\Pr\left[\begin{array}{c|c} \mathcal{A}(N, X, Y, & p, q \leftarrow \mathcal{SP}(\ell/2);\quad N = pq; \\ Z_b \bmod N) = b & g \leftarrow \mathbb{G};\quad x, y, z \leftarrow [1, ord(\mathbb{G})]; \\ & X = g^x \bmod N^2; Y = g^y \bmod N^2; \\ & Z_0 = g^z \bmod N^2; Z_1 = g^{xy} \bmod N^2; \\ & b \leftarrow \{0, 1\}; \end{array}\right] - \frac{1}{2} = \mathsf{negl}(\ell).$$

The Decisional Diffie-Hellman Assumption is related to the regular Diffie-Hellman assumption that says that given g^a and g^b one cannot compute g^{ab} in polynomial time. Clearly this assumption relies on the hardness of computing discrete logs. Reductions in the inverse direction are not known. Interestingly enough, if the factorization of the modulus is available solving the decisional Diffie-Hellman problem (over $\mathbb{Z}_{N^2}$) turns out to be easy.

Theorem 4. *Let N be a composite modulus product of two large primes. Let $\mathbb{G}$ be the cyclic group of quadratic residues modulo N^2. The decisional Diffie-Hellman problem (in $\mathbb{G}$) cannot be harder than factoring.*

Proof. Assume the factorization of the modulus is provided, we are given a challenge triplet $\mathcal{G} = (g^a, g^b, g^c)$ and we have to determine if it is a Diffie-Hellman triplet or not. Our strategy is as follows. Using the factorization of the modulus we compute $a \bmod N$, $b \bmod N$ and $c \bmod N$, then we check whether the following relation holds:

$$ab \equiv c \bmod N. \tag{1}$$

Note that if $\mathcal{G}$ is a Diffie-Hellman triplet, the relation (1) is in fact satisfied with probability 1. On the other hand if $\mathcal{G}$ is not a Diffie-Hellman triplet, the probability that the relation (1) is verified is:

$$\Pr[ab \equiv c \bmod N \ \wedge \ ab \not\equiv c \bmod p'q'N].$$

Since a, b and c are random elements in $\mathbb{Z}^\star_{N^2}$ they can be written as $a = a_1 + a_2N$, $b = b_1 + b_2N$ and $c = c_1 + c_2N$ where a_1, a_2, b_1, b_2, c_1, $c_2 \in Z_N$. Thus denoting $\delta = a_2b_1 + a_1b_2 + a_2b_2N$ the above probability becomes

$$\begin{aligned}&\Pr[a_1b_1 \equiv c_1 \bmod N \ \wedge \ \delta \not\equiv c_2 \bmod \phi(N)]\\ &\qquad\qquad = \Pr[a_1b_1 \equiv c_1 \bmod N] \times \Pr[\delta \not\equiv c_2 \bmod \phi(N)].\end{aligned}$$

The probability that $a_1b_1 = c_1 \bmod N$ for randomly chosen a_1, b_1 and c_1 is clearly $\frac{1}{N}$. On the other hand the probability that the event $\delta \not\equiv c_2 \bmod \phi(N)$ happens is bounded by $1 - \frac{1}{\phi(N)}$. In total the above probability can be bounded by $\frac{1}{N} - \frac{1}{\phi(N)N}$ and thus our strategy succeeds with probability approximately $1 - \frac{1}{N}$. □

Remark 5. A *Gap-Group* is a group in which a computational problem is hard, but the corresponding decisional one is "easy". In other words, the computational and the decisional problems are strictly separated in such a group. This implies that the corresponding *Gap-Problem* [19] is computationally hard. The first example of gap group was proposed by Joux and Nguyen in [15]. The above result shows that, when the factorization of N is provided, $\mathbb{Z}^\star_{N^2}$ can be seen as a some kind of gap group for the Diffie-Hellman problem.

3 The Scheme

Our scheme can be seen as an additively homomorphic variant of the well-known El Gamal cryptosystem [8]. Let h and g be two elements of maximal order in $\mathbb{G}$. Note that, if h is computed as g^x, where $x \in_R [1, \lambda(N^2)]$, then x is coprime with $\mathrm{ord}(\mathbb{G})$ with high probability, and thus h is of maximal order. The message space here is $\mathbb{Z}_N$.

Key Generation - Choose a random element $\alpha \in \mathbb{Z}^\star_{N^2}$, a random value $a \in [1, \mathrm{ord}(\mathbb{G})]$ and set $g = \alpha^2 \bmod N^2$ and $h = g^a \bmod N^2$. The public key is given by the triplet (N, g, h) while the corresponding secret key is a.

Encrypt - Given a message $m \in \mathbb{Z}_N$, a random pad r is chosen uniformly and at random in $\mathbb{Z}_{N^2}$ the ciphertext (A, B) is computed as

$$A = g^r \bmod N^2 \qquad B = h^r(1 + mN) \bmod N^2.$$

First Decryption Procedure - Knowing a, one can compute m as follows

$$m = \frac{B/(A^a) - 1 \bmod N^2}{N}.$$

Alternate Decryption Procedure - If the factorization of the modulus is provided, one can compute $a \bmod N$ and $r \bmod N$ as seen in the previous section. Let $ar \bmod \mathrm{ord}(\mathbb{G}) = \gamma_1 + \gamma_2 N$, thus $\gamma_1 = ar \bmod N$ is efficiently computable. Note that

$$D = \left(\frac{B}{g^{\gamma_1}}\right)^{\lambda(N)} = \frac{(g^{ar}(1 + mN))^{\lambda(N)}}{g^{\gamma_1 \lambda(N)}} = 1 + m\lambda(N)N \bmod N^2.$$

So, still denoting by π the inverse of $\lambda(N)$ in $\mathbb{Z}^\star_N$, one can compute m as

$$m = \frac{D - 1 \bmod N^2}{N} \cdot \pi \pmod N.$$

Remark 6. Note that even though the two described decryption procedures produce the same result when applied to correctly generated ciphertext they are not equivalent from a computational point of view. Indeed knowing the discrete logarithm a of h with respect to the base g in $\mathbb{Z}^\star_{N^2}$ allows to decrypt any valid ciphertext generated using g and h as underlying public key. More precisely knowledge of a allows to decrypt any ciphertext generated with respect of a public key in $\{N\} \times \mathcal{G} \times \mathcal{H}$ where $\mathcal{G} \times \mathcal{H}$ is the set of the couples (g, h) such that $h = g^a \bmod N^2$. On the other hand knowing the factorization of the modulus allows to decrypt ciphertexts generated with respect to *any* public key in $\{N\} \times \mathbb{G} \times \mathbb{G}$.

Remark 7. Another interesting comparison is regarding the invalid (that is, not correctly generated) ciphertexts. Namely, if a ciphertext is not correctly generated, the fault can be detected when decrypting using the secret discrete logarithm. On the other hand, however, if the ciphertext is decrypted using the factorization of the modulus, the resulting - invalid - plaintext cannot be recognized as such. To illustrate this, consider the following example. Let (A, B) a given ciphertext, with $A \in \mathbb{G}$. Since g is a generator of $\mathbb{G}$ there exists r, and thus K, m, such that:

$$A = g^r \text{ where } r \in [1, \mathrm{ord}(\mathbb{G})],$$
$$B = h^r(K + mN) \text{ where } K, m \in \mathbb{Z}_N.$$

If decrypted with the discrete logarithm trapdoor, this leads to a failure, since B/A^a differs from 1 mod N. Then, the incorrect encryption is detected. Conversely if one decrypts using the factorization, one gets a mod N and r mod N and thus (let us denote $ar = \gamma_1 + \gamma_2 N$):

$$\begin{aligned} D = \left(\frac{B}{g^{\gamma_1}}\right)^{\lambda(N)} &= g^{ar\lambda(N)-\gamma_1\lambda(N)}(K+mN)^{\lambda(N)} = (K+mN)^{\lambda(N)} \bmod N^2 \\ &= K^\lambda + \lambda K^{\lambda-1}mN = K^\lambda + \lambda(K^{-1} \bmod N)mN \pmod{N^2} \\ &= 1 + \alpha N + mL\lambda N = 1 + (\alpha\pi + mL \bmod N)\lambda(N)N \quad (\mathrm{mod}\ N)^2, \end{aligned}$$

where one can write $K^{\lambda(N)} = 1 + \alpha N \bmod N^2$, $L = K^{-1} \bmod N$ and where π is the inverse of λ mod N. Thus, the output plaintext is $m' = \alpha\lambda^{-1} + mK^{-1}$ mod N.

4 Security Requirements

4.1 One-Wayness

In this section we prove that the one-wayness of the scheme presented in section 3 can be related to the *Lift Diffie-Hellman* problem that we are about to define.

Let g, X, Y, $Z \in \mathbb{G}$ where $X = g^x \bmod N^2$, $Y = g^y \bmod N^2$ and $Z = g^{xy} \bmod N^2$. The well known (computational) Diffie-Hellman (modulo N^2) asks to compute Z when X, Y, g and N are provided. Similarly we define the *Lift Diffie-Hellman* problem as the one to compute Z when X, Y, g, N and Z mod N are given. Of course it cannot be harder than the Computational Diffie-Hellman problem, but we don't know if the two problems are actually equivalent.

Definition 8 (Lift Diffie-Hellman Problem). *We say that the* Lift Diffie-Hellman *computational problem is hard if, for every probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}()$ *such that for sufficiently large ℓ*

$$\Pr\left[\begin{array}{c|c} \mathcal{A}(N, X, Y, Z \bmod N) & p, q \leftarrow \mathcal{SP}(\ell/2);\ \ N = pq; \\ = Z\ (\mathrm{mod} N^2) & g \leftarrow \mathbb{G};\ \ x, y \leftarrow [1, \mathit{ord}(\mathbb{G})]; \\ & X = g^x \bmod N^2; Y = g^y \bmod N^2; \\ & Z = g^{xy} \bmod N^2; \end{array}\right] = \mathsf{negl}(\ell).$$

Theorem 9 (One-wayness). *The scheme presented in section 3, is one-way if and only if the Lift Diffie-Hellman problem is hard.*

Proof. For $g, h \in \mathbb{G}$, let (N, g, h) be a public key, and $(A, B) = (g^r, h^r(1 + mN)) \bmod N^2$ an encryption of a random message m. If one can efficiently solve the Lift Diffie-Hellman problem then, on input $X = A = g^r$, $Y = h$ and $z = h^r(1+mN) \bmod N = h^r \bmod N$, one can compute the quantity $Z = h^r \bmod N^2$ from which retrieving m is trivial.

Conversely if one can correctly extract m from a correctly generated ciphertext, then such a capability can be used to solve the Lift Diffie-Hellman problem as follows. Assume we are given g, $X = g^x \bmod N^2$, $Y = g^y \bmod N^2$ and $z = g^{xy} \bmod N$. For a randomly chosen message m, we generate a ciphertext (A, B) as follows: we set the public key $(N, g, h = Y)$, $A = X$ and $B = z(1 + mN) \bmod N^2$. Our goal is to retrieve $Z = g^{xy} \bmod N^2$.

Let M be the extracted plaintext corresponding to (A, B). We have by definition:

$$B = Z(1 + MN) = Z + ZMN = Z + (Z \bmod N)MN = Z + zMN \bmod N^2.$$

On the other hand, from the construction of B, it follows that $z + zmN = Z + zMN \bmod N^2$. Thus, we can efficiently compute $Z = z(1+(m-M)N) \bmod N^2$. □

With the following theorem we make explicit the relation existing between the lift Diffie-Hellman problem and the partial Discrete Logarithm problem.

Theorem 10. *If the Partial Discrete Logarithm problem is hard then so is the Lift Diffie-Hellman problem.*

Proof. The proof goes by a standard reduction argument. Assume we are given an oracle $\mathcal{O}$ for the lift Diffie-Hellman problem that on input a triplet of the form $(X, Y, Z) = (g^x \bmod N^2, g^y \bmod N^2, g^{xy} \bmod N)$ returns the value $g^{xy} \bmod N^2$ with some non negligible probability ϵ. Our goal is to use the provided oracle to compute the partial discrete logarithm of a given challenge $h = g^{a_1+a_2N}$ in $\mathbb{Z}^\star_{N^2}$ with respect to the base g (we assume g is a generator of the group $\mathbb{G}$ of quadratic residues in $\mathbb{Z}^\star_{N^2}$). Since g is a generator of $\mathbb{G}$ any quadratic residue c can be written as $c = g^{r_1+r_2\lambda(N)}$ for some $r_1 \in \mathbb{Z}_{\lambda(N)}$ and $r_2 \in \mathbb{Z}_N$. Moreover $g^{\lambda(N)/2} = (1 + \alpha N)$ for some $\alpha \in \mathbb{Z}_N$.

Now we set $X = h$ and $Y = g^{r_1}(1 + r_2N) \bmod N^2$ where r_1 is a random value in $[0 \ldots (N + 1)/4]$, and r_2 a random element in $\mathbb{Z}_N$. Note that Y is not uniformly distributed over $\mathbb{G}$, but its distribution is statistically close to uniform (the statistical difference is of order $O(2^{-|p|})$). Finally we set $Z = X^{r_1} \bmod N$.

Observe that

$$Y = g^{r_1}(1 + r_2N) = g^{r_1}(1 + \alpha r_2\alpha^{-1}N) = g^{r_1+\beta r_2\lambda(N)/2} \pmod{N}^2$$

where $\beta = \alpha^{-1} \bmod N$.

Now we query the oracle $\mathcal{O}$ on input (X, Y, Z) and with probability ϵ it will provide the correct answer Z' such that

$$Z' = g^{(a_1+a_2N)(r_1+\beta r_2\lambda/2)} \bmod N^2 = X^{r_1} g^{a_1\beta r_2\lambda(N)/2} \bmod N^2$$

Thus

$$\frac{Z'}{X^{r_1}} = g^{a_1\beta r_2\lambda(N)/2} \bmod N^2 = (1 + a_1r_2N) \bmod N^2$$

from which we can get a_1 easily.

In [21] Paillier noted that when the order of g is maximal, and N is the product of two safe primes, then the partial discrete logarithm problem is equivalent to the problem of computing the composite residuosity class. This equivalence result can easily be extended to the case on which g is a generator of the group of quadratic residues modulo N^2. This implies that, in our case, the Lift DH problem is at least as hard as the computational class problem introduced by Paillier.

4.2 Semantic security

Theorem 11 (Semantic Security). *If Decisional Diffie-Hellman Assumption in $\mathbb{Z}^{\star}_{N^2}$ holds, then the scheme presented in section 3, is semantically secure.*

Proof. For the sake of contradiction assume the scheme is not semantically secure. This means that there is a polynomial time distinguisher $\mathcal{A}$ that can break semantic security. Our goal then is, given a quadruple $\mathcal{G} = (g, g^a, g^b, g^c)$, to use $\mathcal{A}$ to decide if it is a Diffie-Hellman or a random one (i.e. if $c = ab \bmod \mathrm{ord}(\mathbb{G})$ or not). The public key is first set as (N, g, h) where $h = g^a$; then once the adversary has chosen the messages m_0 and m_1, we flip a bit d and we encrypt m_d as follows: $E(m_d) = (A, B)$ where $A = g^b$ and $B = g^c(1 + m_d N) \bmod N^2$.

Clearly if $\mathcal{G}$ is a Diffie-Hellman quadruple, the above is a valid encryption of m_d and $\mathcal{A}$ will give the correct response with non negligible advantage. On the other hand, if $\mathcal{G}$ is not a Diffie-Hellman quadruple, we claim that even a polynomially unbounded adversary gains no information about m_d from $E(m_d)$ in a strong information-theoretic sense.

Let $c = ab + r \bmod \mathrm{ord}(\mathbb{G})$, we can note that r is random and uniformly distributed in $[1, \mathrm{ord}(\mathbb{G})]$ and can be written as $r_1 + r_2\lambda(N)/2$, with $r_1, r_2 \in \mathbb{Z}_N$. The information received by the adversary (together with the public key) is of the form

$$g^b \bmod N^2, \quad g^{ab+r}(1 + m_d N) \bmod N^2$$

Let us concentrate on the second value (for the sake of simplicity let us assume that $g^{\lambda(N)/2} = (1 + N) \bmod N^2$).

$$\begin{aligned} g^{ab+r}(1 + m_d N) &= g^{ab} g^{r_1} g^{r_2\lambda(N)/2}(1 + m_d N) \bmod N^2 \\ &= g^{ab+r_1}(1 + N)^{r_2}(1 + m_d N) \bmod N^2 \\ &= g^{ab+r_1}(1 + (r_2 + m_d)N) \bmod N^2. \end{aligned}$$

Note that, in the above relation, r_2 hides m_d perfectly and thus $\mathcal{A}$ cannot guess d better than at random.

5 A First Application: Trapdoor Commitment

5.1 A New On-line/Off-line Trapdoor Commitment Scheme

In this section we present a new trapdoor commitment scheme based on the encryption function proposed in section 3. The security of the scheme can be proven to be equivalent to the hardness of factoring.

As sketched in the introduction an useful property of the proposed commitment function is that it allows for an on-line/off-line efficiency trade off, meaning with this that it becomes very efficient to compute when a preprocessing stage is allowed. On-line/off-line trapdoor commitment schemes were first proposed by [5]. In particular, to commit to a message m the sender has to compute only two modular multiplications (using a previously computed value). Such a value is completely independent of m and for this reason can be computed before even knowing to which message to commit to. Furthermore we point out that such a preprocessing step requires a single modular exponentiation. Thus even when the precomputation time is considered, our new scheme is basically as efficient as all the other trapdoor commitment schemes known in the literature.

5.2 Trapdoor Commitments

A trapdoor commitment scheme (a.k.a. *chameleon* commitment [16]) is a function with associated a pair of matching public and private keys (the latter also called the trapdoor of the commitment). The main property we want from such a function is collision-resistance: unless one knows the trapdoor, it is infeasible to find two inputs that map to the same value. On the other hand, knowledge of the trapdoor suffices to find collisions easily.

More formally, a trapdoor commitment scheme is a triplet $(\mathcal{K}, \mathcal{C}, \mathcal{D})$, where:

- $\mathcal{K}$ is a randomized key generation algorithm. On input a security parameter k it outputs a pair of public and private keys: $\mathcal{K}(1^k) = (pk, sk)$.
- The function $\mathcal{C}$ is the commitment function which depends on PK

$$\mathcal{C} : PK \times M \times R \longrightarrow C$$

  where PK, M, R, C are the public key, message, randomness and committed values spaces respectively.
- The function $\mathcal{D}$ is the collision-finding function,

$$\mathcal{D} : SK \times M \times R \times C \times M \longrightarrow R$$

  on input the trapdoor information, a committed value (with its inputs) and a message it finds the corresponding random string. That is, given m, r and $c = \mathcal{C}(pk, m, r)$, for any message m' we have $\mathcal{D}(sk, m, r, c, m') = r'$ such that $c = \mathcal{C}(pk, m', r')$.

We require that

1. $(\mathcal{K}, \mathcal{C}, \mathcal{D})$ are functions computable in polynomial time.
2. No efficient algorithm, taking as input the public key, should be able to find, with non negligible probability, two messages $m \neq m'$ and two random values $r \neq r'$ such that $\mathcal{C}(pk, m, r) = \mathcal{C}(pk, m', r')$.
3. For any message m, the distribution $\{c = \mathcal{C}(pk, m, r)\}_{r \in R}$ has to be indistinguishable from uniform.

Note that the term "indistinguishable" above can be intended as usual in three ways: either the distributions are identical, or they are statistically indistinguishable or computationally indistinguishable (see [12]).

5.3 Previous Work on Trapdoor Commitments

The notion of trapdoor commitments was first proposed by Brassard, Chaum and Crépeau [4] in the context of zero-knowledge arguments. It is well known that trapdoor commitments can be based on the existence of claw-free trapdoor permutations [13, 14].

A specific implementation based on factoring was presented in [13, 14] and it requires a number of modular squarings in $\mathbb{Z}_N^\star$ which is proportional to the length of the committed message.

A famous scheme based on the hardness of computing discrete logarithms has been presented by Boyar et al. [3]. This scheme requires a full modular exponentiation (or alternatively, once again, a number of multiplications which is proportional to the length of the message).

The first commitment scheme with the on-line/off-line property was proposed by [5]. The security of such scheme is based on the hardness of inverting the RSA function (with public exponent set to N).

5.4 Our Commitment Scheme

Key generation – The key generation algorithm, on input a security parameter ℓ produces a modulus N product of two safe primes of size $\ell/2$ together with a square h of maximal order in $\mathbb{G}$. The public key is given by N and h. The factorization of the modulus is the private key.

Committing a message – To commit to a message $m \in \mathbb{Z}_N$ the sender chooses $r \in_R \mathbb{Z}_{N\lambda(N)/2}$ and sets

$$\mathcal{C}(r, m) = h^r(1 + mN) \bmod N^2.$$

Then he sends B to the receiver. Notice that the sender can compute h^r in advance and without needing to know m. Once m is provided, only two more multiplications are required to commit.

Remark 12. As already pointed out in [5] we notice that any commitment $\mathcal{C}$ can be modified in order to obtain some on-line/off-line efficiency property. As a matter of fact such a "modified" commitment scheme $\mathcal{C}'$ would work as follows: during the off-line stage the sender commits to a random value s with randomness r using $\mathcal{C}$ as underlying commitment function. Let $a = \mathcal{C}(s, r)$ be the commitment value. Once m is known the sender commits to it by simply sending a and $c = m \oplus s$. The only problem with this approach is that it increases the length of the commitment. Here we denote by on-line/off-line commitment schemes those which achieve such an efficiency trade-off, *without* increasing the length of the committed value.

Theorem 13 (Security). *Under the assumption that factoring safe-prime moduli is hard the above function $\mathcal{C}$ is a perfectly hiding trapdoor commitment scheme.*

Proof. First notice that, for any m, if r is uniformly distributed in $\mathbb{Z}_{N\lambda(N)/2}$, then $\mathcal{C}(m,r)$ is uniformly distributed in $\mathbb{G}$ (this is because any $1+mN$ is in $\mathbb{G}$, and h^r is uniformly distributed in $\mathbb{G}$, since h is a generator.)

Now given a commitment $\mathcal{C}(m,r) \in \mathbb{G}$ together with the corresponding (m,r), knowing the factorization of the modulus, one can find collisions, for any message m' as follows. Let k be such that $h^{\lambda(N)} = (1+kN) \bmod N^2$, and d the inverse of k in $\mathbb{Z}_N^\star$. Thus we can write

$$\mathcal{C}(m,r) = h^r(1+mN) = h^r(1+kdmN) \bmod N^2 = h^{r+dm\lambda(N)} \bmod N^2.$$

This implies that we can find the required r' as follows

$$r' = r + (m-m')d\lambda(N) \bmod N\lambda(N)/2.$$

Finally to prove security we assume to have an algorithm $\mathcal{A}$ that can find, on input (N,h), two couples (m,r) and (m',r') such that $\mathcal{C}(m,r) = \mathcal{C}(m',r')$. Note that if $r = r'$ this implies that $m = m'$, thus we will assume that $r \neq r'$. From the two given couples one can write:

$$h^r(1+mN) = h^{r'}(1+m'N) \bmod N^2$$

and thus, letting $\Delta_r = r - r'$ and $\Delta_m = m' - m$,

$$h^{\Delta_r} = (1+\Delta_m N) \bmod N^2.$$

Since h has order $\lambda(N)N/2$ and $(1+\Delta_m N)$ has order (at most) N, this means that Δ_r is a multiple of $\lambda(N)/2$. This is enough to factor [17]. □

5.5 Application to On-line/Off-line Signatures

On-line/Off-line signatures were introduced by Even, Goldreich and Micali [9]. The basic idea is to split the signature generation process in two stages: the first one, more expensive, is computed off-line before the message to sign is known. The second, more efficient, phase is performed once the message is available. The proposed method, however, is not very practical as it increases the length of the signature by a quadratic factor. More recently Shamir and Taumann [23] introduced a new paradigm — as well as several efficient constructions — based on chameleon commitments, which performs the above conversion more efficiently. Moreover, this technique, improves on the security of the underlying signature scheme which is used to sign only random strings chosen off-line by the signer.

The basic idea is as follows. During the off-line phase the signer computes a chameleon commitment function on input a random message m' and random string r' and signs the resulting value $H(m',r')$. Once the message m to sign is known, the signer use his knowledge of the trapdoor key to compute a value r such that $H(m,r) = H(m',r')$.

Using our new commitment scheme one can obtain a simple on-line/off-line signature scheme based on factoring.

6 Variants and other Applications

6.1 A Variant of the Cryptosystem

We propose a variant of our scheme in which the randomness is chosen in a smaller set, namely in $\mathbb{Z}_N$ rather than in $\mathbb{Z}_{N^2}$. Note, however, that we still consider an element g of maximal order in $\mathbb{G}$. To encrypt a message $m \in \mathbb{Z}_N$, the operations to perform remain the same:

$$A = g^r \bmod N^2, \qquad B = h^r(1 + mN) \bmod N^2$$

With this variant, the decryption procedure that makes use of the factorization is simplified, and in particular allows to detect some incorrectly generated ciphertext. More precisely, it becomes possible to check whether the underlying random exponent r belongs to the correct interval: before decrypting a ciphertext, the receiver first recover $\rho = \log_g A \bmod N$ using the factorization of the modulus; after that, it checks if $A = g^\rho \bmod N^2$ holds. If the equality does not hold, it rejects.

Of course, if the ciphertext is correctly generated, that is, $r \in \mathbb{Z}_N$, the recovered value ρ is actually r itself, and thus the equality holds. Whereas if A is not correctly generated, the relation $A = g^\rho$ holds with negligible probability only.

Note that decrypting such a ciphertext using the first decryption procedure (i.e., with the discrete logarithm of h to the base g), the decryption never "fail" at this step, simply because the receiver do not recover the value of r, and cannot check its range.

The decryption procedure continues as follows. If using the discrete logarithm trapdoor, the receiver computes h^r as $A^a \bmod N^2$; if using the factorization of N, he computes h^r as $h^\rho \bmod N^2$. Then in both cases, one checks whether $B/h^r = 1$ or not, and if yes, one recovers the plaintext.

6.2 The Small Diffie-Hellman Problem over $\mathbb{Z}^\star_{N^2}$

We introduce a new variant of the Diffie-Hellman Problem. In a nutshell, when given $(A, B) = (g^a, g^b)$ where b *is small*, i.e. $b \in \mathbb{Z}_N$, the computational (resp., decisional) problem consists in computing (resp., distinguishing from a random element in $\mathbb{G}$) the value $C = g^{ab} \bmod N^2$.

We thus state the *Small Decisional Diffie-Hellman Assumption* (S-DDH) over a squared composite modulus of the form $N = pq$.

Assumption 14 (Small-DDH Assumption over $\mathbb{Z}^\star_{N^2}$). *For every probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}()$ *such that for sufficiently large ℓ*

$$\Pr\left[\begin{array}{c|c} \mathcal{A}(N, X, Y, & p, q \leftarrow \mathcal{SP}(\ell/2);\;\; N = pq; \\ Z_b \bmod N) = b & g \leftarrow \mathbb{G}; x, z \leftarrow [1, ord(\mathbb{G})]; y \leftarrow \mathbb{Z}_N; \\ & X = g^x \bmod N^2; Y = g^y \bmod N^2; \\ & Z_0 = g^z \bmod N^2; Z_1 = g^{xy} \bmod N^2; \\ & b \leftarrow \{0, 1\}; \end{array}\right] - \frac{1}{2} = \mathsf{negl}(\ell)$$

One easily proves the following two theorems:

Theorem 15. *The Small (Computational) Diffie-Hellman Problem cannot be harder than factoring.*

Theorem 16. *The above variant of our cryptosystem is semantically secure under the Small Decisional Diffie-Hellman assumption.*

Indeed, knowing the factorization of N allows to fully retrieve the second exponent, thus making the computational problem trivial. The proof for second theorem is similar to the proof for the basic scheme (theorem 11).

6.3 A new Hierarchical Encryption Scheme

A hierarchical encryption scheme [10] can be simply based on our scheme by providing the authority with the master key (the factorization of the modulus) and by giving to each player a local key (an El Gamal-like private key.)

In such a scheme, anybody is able to encrypt a message for a particular player, in such way that only this player and the authority are able to decrypt properly. Moreover, by randomly choosing two elements g, h and encrypting with respect to such a "key", it is possible to design ciphertexts that can be decrypted by nobody but the authority.

Further work might consists in investigate such possibilities in the contexts of electronic voting or digital auctions.

7 Conclusion

This paper is a further investigation within the family of homomorphic cryptosystems modulo a squared composite number. As a first contribution, we provided a new variant of the Cramer-Shoup scheme whose main feature is to offer two different decryption procedures, based on two different trapdoors. In particular, this scheme is the first additively homomorphic cryptosystem whose security is not based on a residuosity-related assumption. Derived from this scheme is a new trapdoor commitment, whose security provably relies on the factorization problem. This commitment scheme allows for a very interesting on-line/off-line efficiency trade-off, without increasing the length of the commitment.

References

1. J. Benaloh. Verifiable Secret-Ballot Elections. PhD Thesis, Yale University, 1987.
2. D. Boneh. The decision Diffie-Hellman problem. In *Proc. of the 3^{rd} ANTS*, LNCS 1423, pp. 48–63, Springer-Verlag, June 1998.
3. J.F. Boyar, S.A. Kurtz, and M.W. Krentel. A Discrete Logarithm Implementation of Perfect Zero-Knowledge Blobs. *Journal of Cryptology*, 2(2):63–76, 1990.
4. G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer and System Sciences*, 37, 1988.

5. D. Catalano, R. Gennaro, N. Howgrave-Graham, and P. Q. Nguyen. Paillier's Cryptosystem Revisited. In *Proc. of the 8th CCS*, pages 206–214. ACM Press, New York, 2001.
6. J. Cohen, M. Fisher. A robust and Verifiable cryptographically secure election scheme. In *Proc. of the 26th FOCS*. IEEE, 1985.
7. R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Eurocrypt '02*, LNCS 2332, pages 45–64. Springer-Verlag, Berlin, 2002.
8. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.
9. S. Even, O. Goldreich and S.Micali. On-line/Off-line Digital Signatures. In *Crypto '89*, pages 263–277. Springer-Verlag, Berlin, 1989.
10. C. Gentry and A. Silverberg. Hierarchical ID-Based Encryption. In *Asiacrypt '02*, LNCS 2501, pages 548–566. Springer-Verlag, Berlin, 2002.
11. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
12. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proc. of the 17th STOC*, pages 291–304. ACM Press, New York, 1985.
13. S. Goldwasser, S. Micali, and R. Rivest. A "Paradoxical" Solution to the Signature Problem. In *Proc. of the 25th FOCS*, pages 441–448. IEEE, New York, 1984.
14. S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM Journal of Computing*, 17(2):281–308, April 1988.
15. A. Joux and K. N Guyen. Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups.
Cryptology eprint archive. `http://eprint.iacr.org/2001/003/`, 2001.
16. H. Krawczyk and T. Rabin. Chameleon Hashing and Signatures. In *Proc. of NDSS '2000*. Internet Society, 2000.
17. G. Miller. Riemann's Hypothesis and Tests for Primality. *Journal of Computer and System Sciences*, 13:300–317, 1976.
18. D. Naccache and J. Stern. A new public key cryptosystem based on higher residues. In *Proc. of 5th Symposium on Computer and Communications Security*. ACM, 1998.
19. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *CT – RSA '01*, LNCS 2020, pages 159–175. Springer-Verlag, Berlin, 2001.
20. T. Okamoto and S. Uchiyama. The Gap-Problems: A new class of problems for the security of cryptographic schemes. In *Proc. of PKC '01*, volume 1992 of *LNCS*. IACR, Springer-Verlag, 1998.
21. P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. In *Eurocrypt '99*, LNCS 1592, pages 223–238. Springer-Verlag, Berlin, 1999.
22. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
23. A. Shamir and Y. Taumann. Improved On-line/Off-line Signature Schemes. In *Crypto '01*, LNCS 2139, pages 355–367. Springer-Verlag, Berlin, 2001.

A Details for theorem 10

In that theorem we use the fact that the distribution of the oracle input is statistically close from the uniform one. Here we prove this fact with more details.

More formally, we want to evaluate the statistical distance δ between the two following distributions:

$$\left\{g^{r_1+r_2\lambda/2}\middle|(r_1,r_2)\in\mathbb{Z}_{\lambda/2}\times\mathbb{Z}_N\right\} \text{ and } \left\{g^{r_1}(1+r_2N)\middle|(r_1,r_2)\in\mathbb{Z}_{\frac{N+1}{4}}\times\mathbb{Z}_N\right\}$$

First we note that the map $\mathbb{Z}_{\lambda_2}\times\mathbb{Z}_N\to\mathbb{G}:(c_1,c_2)\mapsto c=g^{c_1+c_2\lambda/2} \bmod N^2$ is a bijection. Thus we have to compute:

$$\begin{aligned}
\delta &= \sum_{c\in\mathbb{G}}\left|\Pr_{\substack{r_1\in_R\mathbb{Z}_{\lambda/2}\\ r_2\in_R\mathbb{Z}_N}}\left[g^{r_1+r_2\lambda/2}=c\right]-\Pr_{\substack{r_1\in_R\mathbb{Z}_{(N+1)/4}\\ r_2\in_R\mathbb{Z}_N}}\left[g^{r_1}(1+r_2N)=c\right]\right|\\
&= \sum_{c\in\mathbb{G}}\left|\Pr_{r_1\in_R\mathbb{Z}_{\lambda/2}}[r_1=c_1]\Pr_{r_2\in_R\mathbb{Z}_N}[r_2=c_2]-\Pr_{\substack{r_1\in_R\mathbb{Z}_{(N+1)/4}\\ r_2\in_R\mathbb{Z}_N}}\left[g^{r_1}(1+r_2N)=c\right]\right|\\
&= \sum_{c\in\mathbb{G}}\left|\frac{2}{\lambda}\times\frac{1}{N}-\Pr_{\substack{r_1\in_R\mathbb{Z}_{(N+1)/4}\\ r_2\in_R\mathbb{Z}_N}}\left[g^{r_1}(1+r_2N)=c\right]\right|
\end{aligned}$$

Denoting $g^{\lambda/2}=1+\alpha N \bmod N^2$ and $\beta=\alpha^{-1} \bmod N$, we have $g^{r_1}(1+r_2N)=g^{r_1+r_2\beta\lambda/2} \bmod N^2$. Then we observe that for $\lambda/2\le r_1<\frac{N+1}{4}$, we have the following "collision":

$$g^{r_1+r_2\beta\lambda/2}=g^{(r_1-\lambda/2)+(r_2\beta+1)\lambda/2}\pmod{N}^2$$

Hence, two cases appear when summing up (of course, the probabilities that r_2 or $r_2\beta$ or $r_2\beta+1$ equals a given c_2 are all $1/N$):

$$\Pr\left[g^{r_1+r_2\beta\lambda/2}=g^{c_1+c_2\lambda/2}\right]=\begin{cases}2\cdot\frac{4}{N+1}\times\frac{1}{N} & \text{if } 0\le c<\frac{N+1}{4}-\frac{\lambda}{2}\\[2ex] 1\cdot\frac{4}{N+1}\times\frac{1}{N} & \text{if } \frac{N+1}{4}-\frac{\lambda}{2}\le c<\frac{\lambda}{2}\end{cases}$$

Consequently, we gets (recall that $\frac{N+1}{4}-\frac{\lambda}{2}=\frac{p+q}{4}$):

$$\delta=\frac{p+q}{4}\left|\underbrace{\frac{2}{\lambda N}-\frac{8}{N(N+1)}}_{\le 0}\right|+\left(\frac{\lambda}{2}-\frac{p+q}{4}\right)\left|\underbrace{\frac{2}{\lambda N}-\frac{4}{N(N+1)}}_{\ge 0}\right|$$

This is easily seen negligible. $\square$