Trapdoor Permutation Polynomials of $\mathbb{Z}/n\mathbb{Z}$ and Public Key Cryptosystems
(Extended Abstract)

Guilhem Castagnos[1] and Damien Vergnaud*[2]

[1] DMI-XLIM, Université de Limoges,
123, avenue Albert Thomas, 87060 Limoges CEDEX, France
guilhem.castagnos@unilim.fr
[2] École normale supérieure
Département d'informatique, 45 rue d'Ulm, 75230 Paris cedex 05, France
damien.vergnaud@di.ens.fr

Abstract. We define new algorithmic problems and discuss their properties (in particular, we present a careful study of their computational complexity). We apply the new problems to design public key encryption protocols with semantic security relative to their decisional variants. We then show how to provide efficient schemes that are semantically secure under adaptive chosen ciphertext attacks in the random oracle model. Finally, we show that the ideas developed in this extended abstract can be used to design the most efficient known cryptosystem with semantic security under non-adaptive chosen ciphertext attacks in the standard security model.

Keywords: Public Key Encryption, Semantic Security, Standard Model, Random Oracle Model, Chosen-Ciphertext Attacks, Polynomial Diffie-Hellman Problems.

1 Introduction

This paper describes new algorithmic problems using trapdoor permutation polynomials of $\mathbb{Z}/n\mathbb{Z}$ and new constructions of semantically secure public key cryptosystem based on these problems, relative to different scenarios of attacks.

Background. A *trapdoor permutation* is a one-to-one function f that anyone can compute efficiently; however, inverting f is hard unless some "trapdoor" information is also given. Naively, a trapdoor permutation defines a simple public key encryption scheme: the description of f is the public key and the trapdoor is the secret key.

In 1978, Rivest, Shamir, and Adleman proposed the first candidate trapdoor permutation [24]. The RSA setup consists of choosing two distinct large prime

* This work was done while this author was a postdoctoral fellow in the Computer Security group of the Bonn/Aachen International Center for Information Technology.

numbers p and q, and computing the RSA modulus $n = pq$. The public key is n together with an exponent e (relatively prime to $\varphi(n) = (p-1)(q-1)$). The secret key d is defined to be the multiplicative inverse of e modulo $\varphi(n)$. Encryption and decryption of a message in $(\mathbb{Z}/n\mathbb{Z})^\times$ are defined as follows: $\mathcal{E}(m) = m^e \mod n$ and $\mathcal{D}(c) = c^d \mod n$. Hence, the encryption function is an evaluation of the polynomial X^e of $\mathbb{Z}/n\mathbb{Z}[X]$ and the decryption is performed with the polynomial X^d.

In 1993, in [12], Demytko has suggested to replace the monomial X^e by division polynomials of "elliptic curves" defined over the ring $\mathbb{Z}/n\mathbb{Z}$. The same year, Smith and Lennon [26] have proposed a system, LUC, which uses a special type of Lucas sequences. The encryption and decryption functions can also be seen as an evaluation of a polynomial of $\mathbb{Z}/n\mathbb{Z}[X]$, a Dickson polynomial (cf. [19]). As a consequence, the LUC cryptosystem is very similar to a system already proposed by Müller and Nöbauer (cf. [20, 21]).

The security goal for a public key encryption scheme is to guarantee that no partial information about a plaintext message is revealed from its ciphertext, a notion often called *semantic security* or *indistinguishability of ciphertexts* [15]. Unfortunately, the naive public key system built from the three mentioned trapdoor functions is deterministic and hence cannot achieve this security notion. Under a slightly stronger assumption than the intractability of the integer factorization, these primitives give a cryptosystem that is only One-Way under Chosen-Plaintext Attacks (a very weak level of security). The main purpose of the present paper is to propose new combinations of these three polynomial functions giving rise to semantically secure public key cryptosystem.

Several models of attacks have been defined. An encryption scheme that is semantically secure under a Chosen-Plaintext Attack (*resp.* a non-Adaptive Chosen-Ciphertext Attack, *resp.* an Adaptive Chosen-Ciphertext Attack) is said to be IND-CPA secure (*resp.* IND-CCA1 secure, *resp.* IND-CCA2 secure). In [2], it is shown that IND-CCA2 is strictly the strongest notion of semantic security, IND-CCA1, the intermediary notion, and IND-CPA strictly the weakest. Indistinguishably against Chosen-Ciphertext attack is considered to be the correct notion of security for general-purpose public key encryption schemes.

Contributions of the paper. In this paper, we present new algorithmic problems (in section 2, after some notations). Then, in section 3, we give some arguments to validate the cryptographic purpose of those problems, with a careful study of their difficulty and their relations. It is possible to apply the new problems to design public key encryption protocols with semantic security (IND-CPA) relative to the decisional variant of them. This is done in section 4. We then show, in section 5, how these schemes can also be made IND-CCA2 secure assuming the intractability of our decisional RSA variants by using well-known techniques (in the random oracle model [5]). Finally, in section 6, we explain how the ideas developed in this extended abstract can be used to design encryption schemes with higher security in the standard model: for instance, we show that it is pos-

sible to construct the most efficient known IND-CCA1 secure cryptosystem with security analysis in the standard model.

2 Permutation polynomials and new algorithmic problems

2.1 Notations

Let $\mathcal{A}$ be a probabilistic Turing machine running in expected polynomial time (a PPT, for short), and let x be an input for $\mathcal{A}$. The probability space that assigns to a string σ the probability that $\mathcal{A}$, on input x, outputs σ is denoted by $A(x)$. Given a probability space S, a PPT that samples a random element according to S is denoted by $x \xleftarrow{R} S$. For a finite set X, $x \xleftarrow{R} X$ denotes a PPT that samples a random element uniformly at random from X. We will use poly and negl to denote respectively unspecified polynomial any negligible[3] functions.

For any integer $k \geq 2$, we denote $\mathsf{Primes}(k) = \{p \in \mathbb{N}, 2^k < p < 2^{k+1}, p \text{ is prime}\}$ and $2\mathsf{Factor}(k) = \{n \in \mathbb{N}, n = pq, \text{ with } p < q < 2p \text{ and } p, q \in \mathsf{Primes}(k)\}$. For $n \in \mathbb{N}$, $\varphi(n) = \#(\mathbb{Z}/n\mathbb{Z})^\times$ denotes the Euler totient value of n.

For two algorithmic problems A and B, we denote $A \xleftarrow{\mathcal{P}} B$ whenever A is polynomial-time reducible to B, and $A \wedge B$ the problem of solving together A and B.

2.2 RSA and LUC

Let $k \geq 2$, $n = pq \in 2\mathsf{Factor}(k)$ and e be an integer relatively prime to $\varphi(n)$. It is well-known that the polynomial X^e of $\mathbb{Z}/n\mathbb{Z}[X]$ induces a permutation of $(\mathbb{Z}/n\mathbb{Z})^\times$. The RSA encryption corresponds to an evaluation of this polynomial. Moreover, this polynomial has a trapdoor: knowing d such that $ed \equiv 1 \mod \varphi(n)$ allows one to invert the evaluation of this polynomial at any point.

Another permutation polynomial is the LUC function used in the system of [26]. Given two integers a and b such that $a^2 - 4b$ is a non-square, the Lucas sequence V is given by a second-order linear recurrence relation:

$$\forall k \geqslant 1,\ V_{k+1}(a,b) = aV_k(a,b) - bV_{k-1}(a,b),\ V_1(a,b) = a,\ V_0(a,b) = 2.$$

Let e be an integer relatively prime to $(p^2-1)(q^2-1)$. The LUC function, $x \mapsto V_e(x,1) \pmod n$, is a permutation of the set $\{x \in \mathbb{N},\ 0 < x < n,\ \gcd(x,n) = 1,\ \gcd(x^2-4,n) = 1\}$, whose inverse is $x \mapsto V_d(x,1) \pmod n$, where d is the multiplicative inverse of e modulo $(p^2-1)(q^2-1)$ (see [6] for more details on the LUC function). One can see that $V_e(X,1)$ is in fact a polynomial of degree e,

$$V_e(X,1) = \sum_{i=0}^{\lfloor e/2 \rfloor} \frac{e}{e-i}\binom{e-i}{i} X^{e-2i},$$

[3] *i. e.*, $\forall c \geq 0, \exists K_c \in \mathbb{N}, \forall k \in [\![K_c, +\infty[\![, \mathsf{negl}(k) \leq k^{-c}$

which is a special type of Dickson polynomial (cf. [19]) and, for that reason, a permutation polynomial.

These two polynomials of degree e derived from the RSA and LUC cryptosystems can be evaluated at low cost. If we denote $|e|$ the size of e in bits, the evaluation of the RSA polynomial needs $(3/2)|e|$ multiplications modulo n on the average, with the square and multiply algorithm, and $2|e|$ multiplications are needed for the evaluation of the LUC polynomial, using the algorithm of [17].

In the following, we combine these two polynomials to define new algorithmic problems and build new systems. We will define our new problems in a general setting, by considering arbitrary permutation polynomials of $(\mathbb{Z}/n\mathbb{Z})^\times$. The study will be identical as if we were working directly with the RSA and LUC polynomials except for one thing: as the RSA polynomial induces a morphism of $(\mathbb{Z}/n\mathbb{Z})^\times$ some extra reductions will be possible. Consequently, this specific case of polynomial Q such that $Q(xy) = Q(x)Q(y)$ for all elements x, y of $(\mathbb{Z}/n\mathbb{Z})^\times$ will be considered in the study of our problems.

Remark 1. In order to design cryptosystems, one can also follow the ideas of Schwenk and Huber (cf. [25]), and use more general permutation polynomials for which the inverse function is not explicit.

2.3 Permutation polynomials and pointwise inversion

In this paragraph, we define the problem of pointwise inversion of an arbitrary permutation polynomial (PP).

Definition 1. *A* PP generator *is a PPTM that takes a security parameter k as input and outputs a 4-tuple (n, p, q, P) where $n = pq \in 2\mathsf{Factor}(k)$ and $P \in \mathbb{Z}/n\mathbb{Z}[X]$ is a permutation of $(\mathbb{Z}/n\mathbb{Z})^\times$ which can be evaluated at any value of $(\mathbb{Z}/n\mathbb{Z})^\times$ in polynomial time in k. Let $e : \mathbb{N} \longrightarrow \mathbb{N}$. A PP generator Gen is said to be a PP generator of* degree e *if for any $k \in \mathbb{N}$ and any $(n, p, q, P) \leftarrow \mathsf{Gen}(k)$, $\deg(P) \leq e(k)$.*

The next definition quantifies the resistance to pointwise inversion for PP generators (*i. e.*, the problem: given $\alpha = P(a) \in (\mathbb{Z}/n\mathbb{Z})^\times$, compute $a \in (\mathbb{Z}/n\mathbb{Z})^\times$ denoted $P^{-1}(n)$).

Definition 2. *Let Gen be a PP generator. Let $\mathcal{A}$ be a PPTM that takes as input a triple $(n, P, y) \in \mathbb{N} \times \mathbb{Z}/n\mathbb{Z}[X] \times (\mathbb{Z}/n\mathbb{Z})^\times$ and outputs an element $x \in (\mathbb{Z}/n\mathbb{Z})^\times$. We consider the following random experiments, where k is a security parameter:*

$$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{P}^{-1}}_{\mathsf{Gen},\mathcal{A}}(k)}$$
$$(n, p, q, P) \xleftarrow{R} \mathsf{Gen}(k)$$
$$y \xleftarrow{R} (\mathbb{Z}/n\mathbb{Z})^\times$$
$$x \leftarrow \mathcal{A}(n, P, y)$$
$$\textit{Return } 1 \textit{ if } P(x) = y, \ 0 \textit{ otherwise}$$

The success *of $\mathcal{A}$ in solving the pointwise inversion problem is*

$$\mathbf{Succ}^{\mathsf{P}^{-1}}_{\mathsf{Gen},\mathcal{A}}(k) = \Pr[\mathbf{Exp}^{\mathsf{P}^{-1}}_{\mathsf{Gen},\mathcal{A}}(k) = 1].$$

Let τ be an integer and ε a real in $[0,1]$. Gen is said to be (k,τ,ε)-P^{-1}-secure if no adversary $\mathcal{A}$ running in time τ has success $\mathbf{Succ}^{\mathsf{P}^{-1}}_{\mathsf{Gen},\mathcal{A}}(k) \geq \varepsilon$.

2.4 Permutation polynomials and polynomial Diffie-Hellman problem

Definition 3. *A* PDH generator *is a PPTM that takes a security parameter k as input and outputs a 5-tuple (n,p,q,P,Q,R) where $n = pq \in 2\mathsf{Factor}(k)$, $P \in \mathbb{Z}/n\mathbb{Z}[X]$ and $Q \in \mathbb{Z}/n\mathbb{Z}[X]$ are permutations of $(\mathbb{Z}/n\mathbb{Z})^\times$ which can be evaluated at any value of $(\mathbb{Z}/n\mathbb{Z})^\times$ in polynomial time in k and $R \in \mathbb{Z}/n\mathbb{Z}[X,Y]$ is a bivariate polynomial which can be evaluated at any value of $(\mathbb{Z}/n\mathbb{Z})^{\times 2}$ in polynomial time in k. Let $e_P : \mathbb{N} \longrightarrow \mathbb{N}$, $e_Q : \mathbb{N} \longrightarrow \mathbb{N}$ and $e_R : \mathbb{N} \longrightarrow \mathbb{N}$. A PDH generator Gen is said to be a PDH generator of* degree (e_P, e_Q, e_R) *if for any $k \in \mathbb{N}$ and any $(n,p,q,P,Q,R) \leftarrow \mathsf{Gen}(k)$, $\deg(P) \leq e_P(k)$, $\deg(Q) \leq e_Q(k)$, $\deg_X(R) \leq e_R(k)$.*

We now define a new family of algorithmic problems: the *computational polynomial Diffie-Hellman problems* that generalizes the pointwise inversion problem:

Computational Polynomial DH: C-POL-DH(n,P,Q,R)
Given: $\alpha = P(a) \in (\mathbb{Z}/n\mathbb{Z})^\times$ and $\beta = Q(b) \in (\mathbb{Z}/n\mathbb{Z})^\times$;
Find: $R(a,b) \in (\mathbb{Z}/n\mathbb{Z})^\times$.

The problem is named after the Diffie-Hellman key exchange [13] because of its similarity with it in the special case where $Q = P$ and $R(X,Y) = P(XY)$. In this paper, we deal only with the cases $R(X,Y) = XY$, $R(X,Y) = P\big((XY)^\ell\big)$ and $R(X,Y) = Q(X)$ that we denote respectively C-POL1(n,P,Q), C-POL2(n,ℓ,P,Q) and C-DPOL(n,P,Q).

The next definition quantifies the resistance to the computational polynomial Diffie-Hellman problems for PDH generators.

Definition 4. *Let Gen be a PDH generator. Let $\mathcal{A}$ be a PPTM that takes as input a 6-tuple $(n,P,Q,R,y,z) \in \mathbb{N} \times \mathbb{Z}/n\mathbb{Z}[X]^2 \times \mathbb{Z}/n\mathbb{Z}[X,Y] \times (\mathbb{Z}/n\mathbb{Z})^{\times 2}$ and outputs an element $z \in (\mathbb{Z}/n\mathbb{Z})^\times$. We consider the following random experiments, where k is a security parameter:*

$\boxed{\textit{Experiment } \mathbf{Exp}^{\text{C-POL-DH}}_{\mathsf{Gen},\mathcal{A}}(k)}$
$(n,p,q,P,Q,R) \xleftarrow{R} \mathsf{Gen}(k)$
$y \xleftarrow{R} (\mathbb{Z}/n\mathbb{Z})^\times$, $y' \xleftarrow{R} P(y)$
$z \xleftarrow{R} (\mathbb{Z}/n\mathbb{Z})^\times$, $z' \xleftarrow{R} Q(z)$
$x \leftarrow \mathcal{A}(n,P,Q,R,y',z')$
Return 1 *if* $x = R(y,z)$, 0 *otherwise*

The success *of $\mathcal{A}$ in solving the computational polynomial DH problem is*

$$\mathbf{Succ}^{\text{C-POL-DH}}_{\mathsf{Gen},\mathcal{A}}(k) = \Pr[\mathbf{Exp}^{\text{C-POL-DH}}_{\mathsf{Gen},\mathcal{A}}(k) = 1].$$

Let τ be an integer and $\varepsilon \in [0,1]$. Gen *is said to be (k,τ,ε)-*C-POL-DH*-secure if no adversary $\mathcal{A}$ running in time τ has success $\mathbf{Succ}^{\text{C-POL-DH}}_{\mathsf{Gen},\mathcal{A}}(k) \geq \varepsilon$.*

Now, we define the decision problem D-POL-DH(n, P, Q, R) where an element from $(\mathbb{Z}/n\mathbb{Z})^\times$ is given and the algorithm has to decide whether it is a valid candidate for the C-POL-DH(n, P, Q, R) problem.

Decision Polynomial DH: D-POL-DH(n, P, Q, R)
Given: $\alpha = P(a) \in (\mathbb{Z}/n\mathbb{Z})^\times$, $\beta = Q(b) \in (\mathbb{Z}/n\mathbb{Z})^\times$ and $\gamma \in (\mathbb{Z}/n\mathbb{Z})^\times$;
Decide whether: $\gamma = R(a, b)$.

We define three decision problems D-POL1(n, P, Q), D-POL2(n, ℓ, P, Q) and finally D-DPOL(n, P, Q) for the cases $R(X, Y) = XY$, $R(X, Y) = P\big((XY)^\ell\big)$ and $R(X, Y) = Q(X)$ (respectively).

Remark 2. The D-DPOL(n, P, Q) decision problem can be rewritten (the same holds for the computational problem) as follows: given $P(a)$ and $\gamma \in (\mathbb{Z}/n\mathbb{Z})^\times$ decide whether $\gamma = Q(a)$. Hence, the C-DPOL(n, P, Q) and D-DPOL(n, P, Q) problems are generalisations of the Dependent-RSA problems defined in [23].

3 Relations among the new problems

In this section, we discuss the problems defined in the previous section with a careful study of both their difficulty and their relations. For clarity reasons, when the reductions are simple, the theorems are stated with less formalism than the definitions of the previous section. Throughout this section, for a security parameter k, n, P and Q will correspond to the output of a *PDH generator* on input k. For short, we will denote e_p and e_q the degrees of P and Q.

We define an extraction problem, E-POL-DH(n, P, Q, R): Given $P(a)$, $Q(b)$ and $R(a, b)$, find a and b. We denote as before E-POL1, E-POL2, E-DPOL the extraction problems for the special values of R.

We first study the C-POL1 and C-POL2 classes of problems, then we will analyse the C-DPOL class of problem and finally the relation between these three classes.

3.1 The C-POL1 and C-POL2 problems

For the C-POL1 problem, we have the straightforward theorem:

Theorem 1. D-POL1$(n, P, Q) \overset{\mathcal{P}}{\Longleftarrow}$ C-POL1$(n, P, Q) \overset{\mathcal{P}}{\Longleftrightarrow} P^{-1}(n) \wedge Q^{-1}(n)$.

Proof. All the reductions follow from the definition of the C-POL1(n, P, Q) problem except C-POL1$(n, P, Q) \overset{\mathcal{P}}{\Longrightarrow} P^{-1}(n) \wedge Q^{-1}(n)$. Suppose that we know $P(a)$ and we want to compute a. We choose a random b in $(\mathbb{Z}/n\mathbb{Z})^\times$ and we give the value $P(a)$ and $Q(b)$ to an oracle for C-POL1(n, P, Q) which gives the value ab in reply, so we can recover b. We can invert Q with a symmetric process. □

For the C-POL2 problem, we use the extraction problem to state a similar theorem.

Theorem 2. *For an RSA integer n, and two permutation polynomials P and Q of $(\mathbb{Z}/n\mathbb{Z})^\times$,*

$$\text{C-POL2} \wedge \text{E-POL2} \overset{\mathcal{P}}{\Longleftrightarrow} P^{-1} \wedge Q^{-1} \overset{\mathcal{P}}{\Longrightarrow} \begin{matrix}\text{C-POL2}\\ \text{E-POL2}\end{matrix} \overset{\mathcal{P}}{\Longrightarrow} \text{D-POL2}\,.$$

□

We now examine the difficulty of the decision problems.

Difficulty of D-POL1 and D-POL2 The best known way to solve these problems is to solve the corresponding extraction problem (cf. [8]). We know the values of $P(a)$, $Q(b)$ and $R(a, b)$ and we want to find the values of a and b. Suppose that $e_Q \leqslant e_P$ (else the attack is done with the symmetric method). We compute the resultant with respect to the variable Y:

$$S(X) = \operatorname*{Res}_Y(R(X, Y) - R(a, b),\ Q(Y) - Q(b)).$$

This gives a polynomial $S(X)$ of degree $e_R e_Q$ with $S(a) = 0$, so

$$(X - a) \mid \gcd(S(X), P(X) - P(a)).$$

In fact, in many cases[4], we will have $(X - a) = \gcd(S(X), P(X) - P(a))$, and this method allows to recover a. If we are trying to solve the E-POL1 problem we know ab, so we have also recovered b. Else, for the E-POL2 problem, we can recover b by computing $\gcd(R(a, Y) - R(a, b), Q(Y) - Q(b))$.

Remark 3. In the previous description, "the" resultant and "the" gcd of two polynomials with coefficients in $\mathbb{Z}/n\mathbb{Z}$ are understood as the results of the classical algorithms (described in [14] for instance) which compute the resultant and the gcd of polynomials over a field. In the unlikely event, that a non-trivial factor of n appears during this computation, the adversary simply aborts the computation and uses this knowledge to solve the instance of its problem.

[4] Some experimentations confirm that, but, we have not been able to prove this fact. Note that in [8, 23] a similar fact is stated, without proof. Anyway, as our goal is to estimate the size of the degrees to make our cryptosystems secure, only the possibility of an attack matters.

The resultant can be computed with $\mathcal{O}(e_R^2 e_Q \log^2(e_R e_Q) \log\log(e_R e_Q))$ operations in $\mathbb{Z}/n\mathbb{Z}$, according to [14, Corollary 11.18, p. 310]. Note that $e_R = 1$ for E-POL1 and $e_R = \ell e_P$ for E-POL2, so, if ℓ is large enough, this method will be infeasible even if e_P is small.

According to [14, Corollary 11.6, p. 304], the computation of the first gcd can be done in $\mathcal{O}(e \log^2 e \log\log e)$ operations in $\mathbb{Z}/n\mathbb{Z}$, where $e = \max(e_R e_Q, e_P)$ and the computation of the second gcd in $\mathcal{O}(e \log^2 e \log\log e)$ operations in $\mathbb{Z}/n\mathbb{Z}$, where $e = \max(e_R, e_Q)$. This complexity of attacks on the extraction problems will be used in the next section to set key sizes for the cryptosystems that we will built.

Now, suppose that the polynomial Q induces a morphism of $(\mathbb{Z}/n\mathbb{Z})^\times$ (in particular, if Q is associated to the RSA function). In this case, it is possible to make another reduction from $Q^{-1}(n)$ to C-POL2(n, ℓ, P, Q) when $\ell = 1$.

Theorem 3. *Let $e_P : \mathbb{N} \longrightarrow \mathbb{N}$, $e_Q : \mathbb{N} \longrightarrow \mathbb{N}$ and $e_R : \mathbb{N} \longrightarrow \mathbb{N}$ and let Gen be a PDH generator of degree (e_P, e_Q, e_R). Suppose that, for any $k \in \mathbb{N}$ and any $(n, p, q, P, Q, R) \in \mathsf{Gen}(k)$, Q is a morphism of $(\mathbb{Z}/n\mathbb{Z})^\times$ and P is not a polynomial in X^i for any $i > 1$. Let $\tau \in \mathbb{N}^{\mathbb{N}}$, $\varepsilon \in [0,1]^{\mathbb{N}}$ and $\mathcal{A}$ be an adversary that $(k, \tau(k), \varepsilon(k))$-solves the* C-POL2$(n, 1, P, Q)$ *problem for any integer $k \in \mathbb{N}$. There exists an algorithm $\mathcal{B}$ that $(k, \tau'(k), \varepsilon'(k))$-solves the $Q^{-1}(n)$ problem such that*

$$\varepsilon' \geq \varepsilon^{e_P} - \mathsf{negl} \text{ and } \tau' \leq e_P \cdot \tau + e_P^3 \cdot \mathsf{poly}.$$

Proof. The algorithm $\mathcal{A}$ takes as input an instance of the C-POL2$(n, 1, P, Q)$ problem: for any $k \in \mathbb{N}$, given $(n, p, q, P, Q, R) \in \mathsf{Gen}(k)$, $P(a)$ and $Q(b)$ in $(\mathbb{Z}/n\mathbb{Z})^\times$, he will return $P(ab) \in (\mathbb{Z}/n\mathbb{Z})^\times$ in time at most $\tau(k)$ with probability at least $\varepsilon(k)$.

Let $\mathcal{I}$ be the subset of $\{1, \ldots, e_P(k)\}$ of cardinality $m > 1$ such that $P(X) = \sum_{i \in \mathcal{I}} p_i X^i$, where all the $(p_i)_{i \in \mathcal{I}}$ are non-zero elements of $(\mathbb{Z}/n\mathbb{Z})^\times$. Since P is not a polynomial in X^i for any $i > 1$, the gcd of $\mathcal{I}$ is equal to 1.

Given an element $Q(b) \in (\mathbb{Z}/n\mathbb{Z})^\times$, the algorithm $\mathcal{B}$ will recover b. It starts by choosing randomly m couples $(s_j, t_j) \in (\mathbb{Z}/n\mathbb{Z})^\times \times (\mathbb{Z}/n\mathbb{Z})^\times$, with $j \in \{1, \ldots, m\}$ so that all the s_j and the t_j with $j \in \{1, \ldots, m\}$ are distinct.

For each $j \in \{1, \ldots, m\}$, $\mathcal{B}$ gives the values $P(s_j)$ and $Q(bt_j) = Q(b)Q(t_j)$ to the algorithm $\mathcal{A}$ which returns the value of $P(s_j t_j b)$ with probability at least $\varepsilon(k)^m$ (since the m queries are independent). After these queries, $\mathcal{B}$ gets the m equations:

$$\sum_{i \in \mathcal{I}} p_i (s_j t_j)^i b^i = P(s_j t_j b), \text{ for } j \in \{1, \ldots, m\}$$

with the m unknowns $(b^i)_{i \in \mathcal{I}}$. If we denote $\mathcal{I} := \{i_1, i_2, \ldots, i_m\}$, with $0 < i_1 < i_2 < \cdots < i_m = e_P$, the system of equations is associated with the following matrix:

$$M := \begin{bmatrix} p_{i_1}(s_1 t_1)^{i_1} & p_{i_2}(s_1 t_1)^{i_2} & \cdots & p_{i_m}(s_1 t_1)^{i_m} \\ \vdots & \vdots & & \vdots \\ p_{i_1}(s_m t_m)^{i_1} & p_{i_2}(s_m t_m)^{i_2} & \cdots & p_{i_m}(s_m t_m)^{i_m} \end{bmatrix}$$

The method successes if $\det(M) \in (\mathbb{Z}/n\mathbb{Z})^\times$. We focus on the study of $\det(M) \neq 0$ (another value of $(\mathbb{Z}/n\mathbb{Z}) \setminus (\mathbb{Z}/n\mathbb{Z})^\times$ will reveal the factorisation of n).

We have

$$\det(M) = \left(\prod_{j=1}^{m} p_{i_j} c_j^{i_1}\right) \begin{vmatrix} 1 & c_1^{i_2-i_1} & \cdots & c_1^{i_m-i_1} \\ \vdots & \vdots & & \vdots \\ 1 & c_m^{i_2-i_1} & \cdots & c_m^{i_m-i_1} \end{vmatrix}$$

where $c_j := s_j t_j$ for $j = 1, \ldots, m$. This last determinant, D, is a generalized Vandermonde determinant. One can see that

$$D = \left(\prod_{1 \leq i < j \leq m} (c_j - c_i)\right) T(c_1, c_2, \ldots, c_m),$$

where T is a polynomial of degree $i_m - i_1 - m + 1$ in c_m (see [11], for example, for details on this polynomial). So, if all the $(c_j)_{j=1,\ldots,m}$ are distinct, once all the $(s_j)_{j=1,\ldots,m}$, all the $(t_j)_{j=1,\ldots,m-1}$ have been chosen, less than $(i_m - i_1 - m + 1)^2$ values of t_m can make the method fail.

So with standard Gauss elimination, $\mathcal{B}$ can recover the $(b^i)_{i \in \mathcal{I}}$ with $\mathcal{O}(e_P(k)^3)$ operations in $\mathbb{Z}/n\mathbb{Z}$ and m independent queries to the oracle. As $\gcd(\mathcal{I}) = 1$, there exists a linear combination of the elements of $\mathcal{I}$ that equals 1, therefore $\mathcal{B}$ can recover b. □

Remark 4. If $\ell > 1$, with the method used in the proof, we can only recover the value of b^ℓ. Then, we can recover b by computing $\gcd(Q(Y) - Q(b), Y^\ell - b^\ell)$.

3.2 The C-DPOL problem

As shown in Remark 2, the C-DPOL can be rewritten: Given $P(a)$, find $Q(a)$; and the extraction problem, E-DPOL, can be rewritten: Given $P(a)$ and $Q(a)$, find a. We then have the following (straightforward) theorem, that generalizes ([23], Theorem 3):

Theorem 4. *Let* Gen *be a PDH generator. We have:*

$$\text{C-DPOL} \wedge \text{E-DPOL} \overset{\mathcal{P}}{\Longleftrightarrow} P^{-1} \overset{\mathcal{P}}{\Longrightarrow} \begin{matrix} \text{C-DPOL} \\ \text{E-DPOL} \end{matrix} \overset{\mathcal{P}}{\Longrightarrow} \text{D-DPOL}\,.$$

Now let's try to solve the E-DPOL problem. We know the values of $P(a)$ and $Q(a)$ and we want to compute the value of a. We have $(X - a)$ divides $\gcd(P(X) - P(a), Q(X) - Q(a))$ and again, in many cases, we will have an equality. The complexity of the computation of the gcd is $\mathcal{O}(e \log^2 e \log\log e)$ operations in $\mathbb{Z}/n\mathbb{Z}$, where $e = \max(e_Q, e_P)$. If e_Q and e_P are greater than, say 2^{60}, this method will fail. This method for solving E-DPOL problem allows to break the D-DPOL problem. In conjunction with Theorem 4, this method also leads to a reduction from the $P^{-1}(n)$ problem to the C-DPOL(n, P, Q) problem in $\mathcal{O}(e \log^2 e \log\log e)$ operations in $\mathbb{Z}/n\mathbb{Z}$.

Again, suppose that the polynomial P induces a morphism of $(\mathbb{Z}/n\mathbb{Z})^\times$: we can also make another reduction from C-DPOL(n, P, Q) to P^{-1}.

Theorem 5. *Let $e_P : \mathbb{N} \longrightarrow \mathbb{N}$, $e_Q : \mathbb{N} \longrightarrow \mathbb{N}$ and $e_R : \mathbb{N} \longrightarrow \mathbb{N}$ and let* Gen *be a PDH generator of degree (e_P, e_Q, e_R). Suppose that, for any $k \in \mathbb{N}$ and any $(n, p, q, P, Q, R) \in \mathsf{Gen}(k)$, P is a morphism of $(\mathbb{Z}/n\mathbb{Z})^\times$ and Q is not a polynomial in X^i for any $i > 1$. Let $\tau \in \mathbb{N}^{\mathbb{N}}$, $\varepsilon \in [0,1]^{\mathbb{N}}$ and $\mathcal{A}$ be an adversary that $(k, \tau(k), \varepsilon(k))$-solves the* C-DPOL$(n, P, Q)$ *problem for any integer $k \in \mathbb{N}$. There exists an algorithm $\mathcal{B}$ that $(k, \tau'(k), \varepsilon'(k))$-solves the $P^{-1}(n)$ problem such that*

$$\varepsilon' \geq \varepsilon^{e_Q} - \mathsf{negl} \text{ and } \tau' \leq e_Q \cdot \tau + e_Q^3 \cdot \mathsf{poly}.$$

□

The proof is analogous to that of Theorem 3.

3.3 Relations among the three classes of problems.

It is trivial to see that for all $\ell \geq 1$,

$$\begin{array}{ccc} \text{C-POL1}(n, P, Q) & \stackrel{\mathcal{P}}{\Longrightarrow} & \text{D-POL1}(n, P, Q) \\ \Downarrow \mathcal{P} & & \Uparrow \mathcal{P} \\ \text{C-POL2}(n, \ell, P, Q) & \stackrel{\mathcal{P}}{\Longrightarrow} & \text{D-POL2}(n, \ell, P, Q) \end{array}$$

In the special case where the polynomial P induces a morphism of $(\mathbb{Z}/n\mathbb{Z})^\times$, as $P((ab)^\ell) = P(ab)^\ell$ and $P(ab) = P(a)P(b)$, we have the following theorem.

Theorem 6. *If P induces a morphism of $(\mathbb{Z}/n\mathbb{Z})^\times$,*

$$\text{C-POL2}(n, \ell, P, Q) \stackrel{\mathcal{P}}{\Longleftarrow} \text{C-POL2}(n, 1, P, Q) \stackrel{\mathcal{P}}{\Longleftrightarrow} \text{C-DPOL}(n, Q, P),$$

$$\text{D-POL2}(n, \ell, P, Q) \stackrel{\mathcal{P}}{\Longrightarrow} \text{D-POL2}(n, 1, P, Q) \stackrel{\mathcal{P}}{\Longleftrightarrow} \text{D-DPOL}(n, Q, P).$$

□

In fact the idea of the proof of the previous theorem can be used to make a reduction from D-POL1(n, P, Q) to D-DPOL(n, P, Q) if Q is a morphism. Suppose that we know the values of $P(a)$ and $Q(b)$ and that we want to decide if an element c equals ab. If this is the case, $Q(c) = Q(a)Q(b)$. So we can submit $P(a)$ and $Q(c)/Q(b)$ to an oracle of the D-DPOL(n, P, Q) problem to solve the D-POL1(n, P, Q) problem.

As D-DPOL$(n, Q, P) \stackrel{\mathcal{P}}{\Longleftrightarrow}$ D-DPOL(n, P, Q), we have proved the following theorem:

Theorem 7. *If P or Q induces a morphism of $(\mathbb{Z}/n\mathbb{Z})^\times$,*

$$\text{D-POL1}(n, P, Q) \stackrel{\mathcal{P}}{\Longleftarrow} \text{D-DPOL}(n, P, Q).$$

□

4 IND-CPA-secure public key cryptosystems

Let f be a trapdoor permutation and g be another function with the following pseudo-randomness property: the distribution of $(f(k), g(k))$ induced by a random k cannot be distinguished (by a polynomially bounded adversary) from a randomly distributed $(f(k), r)$. Then the encryption $E(m) = (f(k), g(k) \oplus m)$ is semantically secure (cf. [23, 7]). In this section, we revisit this approach by using for the function g a trapdoor permutation.

Following this paradigm, we define three new encryption schemes where the public key is (n, P, Q) or (n, P, Q, R) and the corresponding secret key is P^{-1} or (P^{-1}, Q^{-1}), with the notations of the previous section (*i. e.*, n, P, Q, R correspond to the output of a *PDH generator* for a given security parameter). To encrypt a message $m \in (\mathbb{Z}/n\mathbb{Z})^\times$, a user picks at random $r \in (\mathbb{Z}/n\mathbb{Z})^\times$ (or $(r_0, r_1) \in (\mathbb{Z}/n\mathbb{Z})^{\times 2}$) and uses one of the three following encryption functions:

Function 1: $(m, r_0, r_1) \mapsto \big(P(r_0), Q(r_1), mR(r_0, r_1)\big)$

Function 2: $(m, r) \mapsto \big(P(r), mQ(r)\big)$

Function 3: $(m, r) \mapsto \big(P(mr), Q(r^{-1})\big)$

To decrypt, a user uses his knowledge P^{-1} or (P^{-1}, Q^{-1}) to recover r or (r_0, r_1) then m.

Theorem 8. *The previous schemes are One-Way and semantically secure under Chosen Plaintext Attack relative to the following problems:*

Encryption function	*One-Wayness*	*Semantic security*
Function *1,* $R(X, Y) = XY$	C-POL1(n, P, Q)	D-POL1(n, P, Q)
Function *1,* $R(X, Y) = P\big((XY)^\ell\big)$	C-POL2(n, ℓ, P, Q)	D-POL2(n, ℓ, P, Q)
Function *2*	C-DPOL(n, P, Q)	D-DPOL(n, P, Q)
Function *3*	C-POL1(n, P, Q)	D-POL1$(n, P, Q)^{(*)}$

$^{(*)}$ *If P or Q is a morphism.*

Proof (Sketch). For the first three schemes, the proof relies on the analysis done in [7]. The fourth encryption scheme mixes the one-time-pad masking approach used above with the trapdoor property of the function induced by P. The semantic security of this scheme can be rewritten: there is no polynomial algorithm that can choose a value $m \in (\mathbb{Z}/n\mathbb{Z})^\times$ and then recognize the couples $(P(a), Q(b)) \in (\mathbb{Z}/n\mathbb{Z})^\times \times (\mathbb{Z}/n\mathbb{Z})^\times$, satisfying $ab = m$. It's easy to see that if P or Q is a morphism, this assertion is equivalent to the intractability of the D-POL1(n, P, Q) decision problem. □

Efficiency considerations. From the encryption functions above, we design five practical cryptosystems, three with Function 1, by setting $R(X,Y) = XY$, $R(X,Y) = P(XY)$ and $R(X,Y) = P\big((XY)^\ell$ with $\ell > 1$; one with Function 2; and one with Function 3.

For the polynomial P we use the LUC polynomial $V_e(X,1)$ and for the polynomial Q, the RSA polynomial of the same degree, *i. e.*, $Q(X) = X^e$. In order to compare the efficiency of these schemes, we use an RSA modulus of 1024 bits and we adjust the parameter e (and ℓ) in order to achieve a 2^{80} security. For this, we use Theorem 8 and the analysis done in Section 3. These new cryptosystems and the corresponding values of the parameters are given in the following table.

Scheme	Ciphertext	Public keys
Scheme 1	$V_e(r_o,1), {r_1}^e, mr_0r_1$	$e = 2^{67} + 3$.
Scheme 2	$V_e(r_o,1), {r_1}^e, mV_e(r_0r_1)$	$e = 2^{23} + 9$.
Scheme 3	$V_e(r_o,1), {r_1}^e, mV_e\big((r_0r_1)^\ell\big)$	$e = 5$ and $\ell = 2^{31} + 65$
Scheme 4	$V_e(r,1), mr^e$	$e = 2^{67} + 3$.
Scheme 5	$V_e(mr,1), r^{-e}$	$e = 2^{67} + 3$.

Now, we compare the concrete efficiency of our new schemes with the one from [7, 23]. For the D-RSA scheme of [23] we use $e = 2^{67} + 3$ and for the scheme of Catalano *et al.* ([7]), we use $e = 2^{16} + 1$. The unity of complexity is the cost of a multiplication modulo n. We use the following estimations: a multiplication modulo n^2 costs as much as three multiplications modulo n, an inversion costs 10 multiplications, a multiplication modulo p costs 1/3 multiplication modulo n and a multiplication modulo p^2 costs one multiplication modulo n. We use the Chinese Remainder Theorem for the decryption process of all schemes. The comparison is done in the following table.

Scheme	D-RSA	Catalano	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5
Input	1024						
Output	2048		3072			2048	
Encryption	139	52	205	119	**44**	204	214
Decryption	567	570	1204	1234	1228	736	1196

One can remark that the new cryptosystems appear to be quite practical. If the decryption phase of these schemes (except Scheme 4) suffers from the cost of the simultaneous inversions of the LUC and RSA function, the encryption process is very fast and Scheme 3 (which is an improvement of Scheme 2) can

encrypt *faster* than the D-RSA and Catalano *et al.* cryptosystems. Schemes 1 and 5 have a similar complexity and are the most efficient semantically secure cryptosystems with One-Wayness proved equivalent to the problem of inverting *simultaneously* RSA *and* LUC.

5 IND-CCA2-secure public key cryptosystems in the ROM

In this section, we apply standard techniques to obtain chosen ciphertext security (from these new primitives) in the random oracle model formalized by Bellare and Rogaway in 1993 [5], in which cryptographic protocols are designed and proved secure under the additional assumption that publicly available functions that are chosen truly at random exist. These random oracles can only be accessed in a black-box way, by providing an input and obtaining the corresponding output. A similar method is used in [23] for instance.

Let h be a cryptographic hash function (seen like a random oracle). With the previous notations, the public key is now (n, P, Q, h) or (n, P, Q, R, h), and the corresponding secret key is P^{-1} or (P^{-1}, Q^{-1}). To encrypt a message m of $(\mathbb{Z}/n\mathbb{Z})^\times$, a user picks at random $r \in (\mathbb{Z}/n\mathbb{Z})^\times$ (or $(r_0, r_1) \in (\mathbb{Z}/n\mathbb{Z})^{\times 2}$) and uses one of the three following encryption functions:

Function 1: $(m, r_0, r_1) \mapsto (P(r_0), Q(r_1), mR(r_0, r_1), h(m||r_0||r_1))$

Function 2: $(m, r) \mapsto (P(r), mQ(r), h(m||r))$

Function 3: $(m, r) \mapsto (P(mr), Q(r^{-1}), h(m||r))$

The decryption process is done as in Section 4 except that the message is returned only if the hash value is correct.

Theorem 9. *The previous schemes are semantically secure against Adaptive Chosen Ciphertext Attack in the Random Oracle Model relative to the following problems:*

Encryption function	*Semantic security*
Function *1,* $R(X, Y) = XY$	D-POL1(n, P, Q)
Function *1,* $R(X, Y) = P\big((XY)^\ell\big)$	D-POL2(n, ℓ, P, Q)
Function *2*	D-DPOL(n, P, Q)
Function *3*	D-POL1$(n, P, Q)^{(*)}$

$^{(*)}$ *If P or Q is a morphism.*

Proof (Sketch). The proof is standard. The random oracle model is simulated in the standard way and the instance of the decision problem is embedded in the challenge ciphertext without updating the hash table. The hash table is used to

simulate the decryption oracle: when the adversary makes a decryption query, the reduction looks in this table to get the pair (m, r) or the triple (m, r_0, r_1) corresponding to the last element of the ciphertext. It returns the message m only if the encryption of m with the value r or the pair (r_0, r_1) produces the same ciphertext; otherwise, it returns the reject symbol.

The simulation of the random oracle is perfect and the probability that a decryption query is incorrect is exponentially small in the size of the hash values. Details can be found in [23], for instance. □

Remark 5. As done in [23], it is also possible to modify our schemes in order to make them IND-CCA2 in the random oracle model relative to the corresponding computational problems. This transformation permits to use smaller degree polynomials P and Q in the encryption procedure but unfortunately the resulting schemes are IND-CPA-secure only in the Random Oracle Model.

6 IND-CCA1-secure public key cryptosystems in the standard model

6.1 Damgård's Elgamal

Let $\mathbb{G}$ be an additive group of prime order q, let k be the bit size of the elements of $\mathbb{G}$ and let P be a generator of $\mathbb{G}$. In 1991, Damgård [10] presented a simple variant of the Elgamal encryption scheme in $\mathbb{G}$. In his proposal, Alice publishes two public keys $A_1 = [a_1] \cdot P$ and $A_2 = [a_2] \cdot P$ and keeps secret their discrete logarithms a_1 and a_2. When Bob wants to send privately a message $m \in \{0,1\}^k$ to Alice, he picks uniformly at random an integer $r \in [\![1, q-1]\!]$ and transmits the triple (Q_1, Q_2, C) where $Q_1 = [r] \cdot P$, $Q_2 = [r] \cdot A_1$ and $C = m \oplus ([r] \cdot A_2)$. When she receives the ciphertext (Q_1, Q_2, C), Alice checks whether the equality $Q_2 = [a_1] \cdot Q_1$ holds: if it is the case, she retrieves the message m, as $m = C \oplus ([a_2] \cdot Q_1)$, otherwise she rejects the ciphertext.

Damgård proved that if the DDH problem is hard in $\mathbb{G}$, then this scheme is IND-CCA1-secure, if we assume the so-called *knowledge-of-exponent assumption* [22]. Intuitively this assumption states that, without the knowledge of a_1, the only way to generate couples $(Q_1, Q_2) \in \mathbb{G}^2$, satisfying $Q_2 = [a_1] \cdot Q_1$, is to choose an integer $r \in [\![1, q-1]\!]$ and to compute $Q_1 = [r] \cdot P$ and $Q_2 = [r] \cdot A_1$.

The knowledge-of-exponent assumption is a strong and non-standard one, but to date it has not been proven false. It has been criticized for assuming one can perform "reverse engineering" of an adversary. It should therefore be considered with caution, all the more since Bellare and Palacio [3] showed that a somehow similar assumption used in [16] is false.

The function that maps r to $([r] \cdot P, [r] \cdot A_1)$ is what Damgård called a *One-Way function with sparse image* (*i. e.*, only a very small fraction of $\mathbb{G}^2$ is in its image and it seems computationally infeasible to sample an element of this set without the knowledge of its preimage). Damgård predicts that such One-Way

functions would be extremely useful in other contexts. His proposal has indeed found applications in identification and zero-knowledge protocols ([3, 4, 1, 16]), but it has proved to be extremely difficult to find other examples that can be reduced to reasonable assumptions. The purpose of the next paragraph is to explain how our approach can be extended in order to propose such a function.

6.2 Knowledge of preimage assumption

Let Gen be a PDH generator. Suppose we are given (n, P_1, P_2, R, y, z) an element of $\mathbb{N} \times \mathbb{Z}/n\mathbb{Z}[X]^3 \times \mathbb{N}$ output by Gen and want to output a pair (x, y) of $((\mathbb{Z}/n\mathbb{Z})^\times)^2$, such that $P_1^{-1}(x) = P_2^{-1}(y)$. One way to do this is to pick some $a \in (\mathbb{Z}/n\mathbb{Z})^\times$ and let $x = P_1(a)$ and $y = P_2(a)$. Intuitively, the *knowledge of preimage assumption* (KPA) can be viewed as saying that this is the "only" way to produce such a pair.

There are many ways in which the formulation can be varied to capture the KPA. We will say that for any PPTM outputting a pair $(P_1(a), P_2(a))$, there is an "extractor" than can return the preimage a. For our purposes, it is necessary to allow the adversary to be randomized as in [1] (in that case, it is important that the extractor gets the coins of the adversary as an additional input, since otherwise the assumption is clearly false).

Definition 5. *Let* Gen *be a PDH generator and let* $\mathbf{A}$ *and* $\overline{\mathbf{A}}$ *be two PPTM's. We consider the following random experiments, where* $k \in \mathbb{N}$ *is a security parameter:*

Experiment $\mathbf{Exp}^{\mathsf{kpa}}_{\mathsf{Gen},\mathbf{A},\overline{\mathbf{A}}}(k)$

$(n, p, q, P_1, P_2, R) \xleftarrow{R} \mathsf{Gen}(k)$

$(x, y) \xleftarrow{r} \mathbf{A}(n, P_1, P_2)$

$\alpha \xleftarrow{r} \overline{\mathbf{A}}(n, P_1, P_2)$

Return 1 *if* $(x, y) \in ((\mathbb{Z}/n\mathbb{Z})^\times)^2$, $\exists a \in (\mathbb{Z}/n\mathbb{Z})^\times$ *s.t.*
$(x, y) = (P_1(a), P_2(a))$ *and* $a \neq \alpha$,

Return 0 *otherwise*

We define the advantage *of* $\mathbf{A}$ *relative to* $\overline{\mathbf{A}}$ *in these experiments* via

$$\mathbf{Adv}^{\mathsf{kpa}}_{\mathsf{Gen},\mathbf{A},\overline{\mathbf{A}}}(k) = \Pr\left[\mathbf{Exp}^{\mathsf{kpa}}_{\mathsf{Gen},\mathbf{A},\overline{\mathbf{A}}}(k) = 1\right].$$

Let $\varepsilon \in [0, 1]^{\mathbb{N}}$,

1. $\overline{\mathbf{A}}$ *is a* ε-kpa-extractor *for* $\mathbf{A}$ *if for all positive integers* k,
$$\mathbf{Adv}^{\mathsf{kpa}}_{\mathsf{Gen},\mathbf{A},\overline{\mathbf{A}}}(k) \leq \varepsilon(k).$$
2. *We say that the knowledge-of-preimage assumption holds for* Gen *if for every PPTM* $\mathbf{A}$*, there exists a PPTM* $\overline{\mathbf{A}}$ *and a negligible function* ε *such that* $\overline{\mathbf{A}}$ *is a* ε-KPA-extractor *for* $\mathbf{A}$.

3. *We say that the strong knowledge-of-preimage assumption (SKPA) holds for* Gen *if there exists a PPTM* $\mathcal{E}$ *such that for every PPTM* $\mathbf{A}$*, there exists a negligible function* ε *such that* $\mathcal{E}$ *is a* ε-KPA-extractor *for* $\mathbf{A}$*.*

6.3 New construction

Following Damgård's technique, we define a new encryption scheme where the public key is (n, P_1, P_2, Q) where P_1, P_2 and Q are One-Way permutations of $(\mathbb{Z}/n\mathbb{Z})^\times$ and the corresponding secret key is P^{-1}. To encrypt a message $m \in (\mathbb{Z}/n\mathbb{Z})^\times$, a user picks at random $r \in (\mathbb{Z}/n\mathbb{Z})^\times$ and uses the following encryption function:

Function 1: $(m, r) \mapsto \big(P_1(r), P_2(r), m \cdot Q(r)\big)$

When he receives a ciphertext (x, y, C), a user uses his knowledge of P^{-1} to check whether the equality $P_2(P_1^{-1}(x)) = y$ holds: if it is the case, he retrieves the message m, as $m = C/Q(P_1^{-1}(x))$, otherwise he rejects the ciphertext.

Theorem 10. *The previous schemes is One-Way and semantically secure under non-adaptive Chosen Ciphertext Attack relative to the following problems:*

Encryption function	*One-Wayness*	*Semantic security*
Function *1*	C-DPOL(n, P_1, Q)	D-DPOL(n, P_1, Q) *under SKPA*

Proof (Sketch). Theorem 8 insures that this scheme is one-way assuming the intractability of the C-DPOL(n, P_1, Q) problem and semantically secure under chosen-plaintext attacks if the D-DPOL(n, P_1, Q) problem is intractable. Following [4], it is straightforward to see that the scheme is plaintext-aware (PA1) assuming the strong knowledge of preimage assumption and *Theorem 1* from this paper implies that our new scheme is IND-CCA1-secure if the D-DPOL(n, P_1, Q) problem is intractable and the strong knowledge of preimage assumption holds for the underlying PDH generator. □

If one sets $P_1(X) = X^e$, $P_2(X) = (X + 1)^e$ and $Q(X) = (X + 2)^e$ with sufficiently large e in order to make the D-DPOL problem infeasible in reasonable time (see the analysis of subsection 3.2 and section 4) one obtains an IND-CCA1-secure system faster than the one of Damgård.

7 Conclusion

We have defined new algorithmic problems, derived from the RSA assumption, and discuss their computational difficulty. We have applied them to design public key encryption protocols with IND-CPA-security and IND-CCA2-security in the random oracle model under the assumption of the intractability of their decisional variants.

The ideas developed in this extended abstract can be used to design encryption schemes with higher security. For instance, by using the approach proposed by Cramer and Shoup in [9], we have been able to design a concrete encryption scheme that is proven IND-CCA2-secure in the standard model based on the difficulty of the new algorithmic problems. Details will appear elsewhere.

References

1. B. Barak, Y. Lindell, and S. Vadhan, *Lower Bounds for Non-Black-Box Zero Knowledge.*, Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS 2003) (M. Sudan, ed.), IEEE Computer Society, 2003, pp. 384–393.
2. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, *Relations Among Notions of Security for Public-Key Encryption Schemes.*, in Krawczyk [18], pp. 26–45.
3. M. Bellare and A. Palacio, *The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols.*, Advances in Cryptology - Crypto 2004 (M. K. Franklin, ed.), Lect. Notes Comput. Sci., vol. 3152, Springer, 2004, pp. 273–289.
4. M. Bellare and A. Palacio, *Towards Plaintext-Aware Public-Key Encryption Without Random Oracles.*, Advances in Cryptology - Asiacrypt 2004 (P. J. Lee, ed.), Lect. Notes Comput. Sci., vol. 3329, Springer, 2004, pp. 48–62.
5. M. Bellare and P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols.*, Proceedings of the First ACM Conference on Computer and Communications Security (D. Denning, R. Pyle, R. Ganesan, R. Sandhu, and V. Ashby, eds.), ACM Press, 1993, pp. 62–73.
6. G. Castagnos, *An efficient probabilistic public-key cryptosystem over quadratic fields quotients.*, Finite Fields Appl. **13** (2007), no. 3, 563–576.
7. D. Catalano, R. Gennaro, N. Howgrave-Graham, and P. Q. Nguyen, *Paillier's cryptosystem revisited.*, Proceedings of the 8th ACM Conference on Computer and Communications Security, 2001, pp. 206–214.
8. D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, *Low-Exponent RSA with Related Messages.*, Advances in Cryptology - Eurocrypt'96 (U. M. Maurer, ed.), Lect. Notes Comput. Sci., vol. 1070, Springer, 1996, pp. 1–9.
9. R. Cramer and V. Shoup, *Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack.*, SIAM J. Comput. **33** (2003), no. 1, 167–226.
10. I. B. Damgård, *Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks.*, Advances in Cryptology - Crypto'91 (J. Feigenbaum, ed.), Lect. Notes Comput. Sci., vol. 576, Springer, 1992, pp. 445–456.
11. S. De Marchi, *Polynomials arising in factoring generalized Vandermonde determinants: an algorithm for computing their coefficients.*, Math. and Comput. Modelling **34** (2001), no. 3–4, 271–281.
12. N. Demytko, *A New Elliptic Curve Based Analogue of RSA.*, Advances in Cryptology - Eurocrypt'93 (T. Helleseth, ed.), Lect. Notes Comput. Sci., vol. 765, Springer, 1994, pp. 40–49.
13. W. Diffie and M. E. Hellman, *New Directions in Cryptography.*, IEEE Trans. Inf. Theory **22** (1976), 644–654.
14. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press., 1999.

15. S. Goldwasser and S. Micali, *Probabilistic Encryption.*, J. Comput. Syst. Sci. **28** (1984), 270–299.
16. S. Hada and T. Tanaka, *On the Existence of 3-Round Zero-Knowledge Protocols.*, in Krawczyk [18], pp. 408–423.
17. M. Joye and J. Quisquater, *Efficient computation of full Lucas sequences.*, Electronics Letters 32, 6 (Mar. 1996), 1996, pp. 537–538.
18. H. Krawczyk (ed.), *Advances in Cryptology* - CRYPTO *'98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, Lect. Notes Comput. Sci., vol. 1462, Springer, 1998.
19. R. Lidl, G. L. Mullen, and G. Turnwald, *Dickson Polynomials.*, Pitman Monographs and Surveys in Pure and Applied Mathematics, vol. 65, Longman Scientific & Technical, 1993.
20. W. B. Müller and R. Nöbauer, *Some remarks on public-key cryptosystems.*, Sci. Math. Hungar. **16** (1981), 71–76.
21. W. B. Müller and R. Nöbauer, *Cryptanalysis of the Dickson-scheme.*, Proc. of Eurocrypt' 85, Springer-Verlag, 1986, pp. 50–61.
22. M. Naor, *On Cryptographic Assumptions and Challenges.*, Advances in Cryptology - CRYPTO 2003 (D. Boneh, ed.), Lect. Notes Comput. Sci., vol. 2729, Springer, 2003, pp. 96–109.
23. D. Pointcheval, *New Public Key Cryptosystems Based on the Dependent-RSA Problems.*, Advances in Cryptology - EUROCRYPT'99 (J. Stern, ed.), Lect. Notes Comput. Sci., vol. 1592, Springer, 1999, pp. 239–254.
24. R. L. Rivest, A. Shamir, and L. M. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.*, Comm. ACM **21** (1978), 120–126.
25. J. Schwenk and K. Huber, *Public key encryption and digital signatures based on permutation polynomials.*, Electronics Letters 34, 8 (Apr. 1998), 1998, pp. 759–760.
26. P. Smith and M. J .J Lennon, *LUC: A new public key system*, Proc. of the Ninth IFIP Int. Symp. on Computer Security (1993), 1993, pp. 103–117.