The Polynomial Composition Problem in $(\mathbb{Z}/n\mathbb{Z})[X]$

Marc Joye[1], David Naccache[2], and Stéphanie Porte[1]

[1] Gemplus Card International
Avenue du Jujubier, ZI Athélia IV, 13705 La Ciotat Cedex, France
{marc.joye, stephanie.porte}@gemplus.com
[2] Gemplus Card International
34 rue Guynemer, 92447 Issy-les-Moulineaux Cedex, France
david.naccache@gemplus.com

Abstract. Let n be an RSA modulus and let $P, Q \in (\mathbb{Z}/n\mathbb{Z})[X]$. This paper explores the following problem: Given Q and $Q(P)$, find P. We shed light on the connections between the above problem to the RSA problem and derive from it new zero-knowledge protocols.

1 Introduction

In this paper we introduce a new cryptographic problem, the *Polynomial Composition Problem* (PCP), which can be stated as follows.

> *Let P and Q be two polynomials in $(\mathbb{Z}/n\mathbb{Z})[X]$ where n is an RSA modulus. Given polynomials Q and $S := Q(P)$, find P.*

Most public-key cryptographic schemes base their security on the difficulty of solving a hard mathematical problem. As the number of hard problems with applications to cryptography is rather limited the investigation of new problems is of central importance in cryptography. To clearly understand the polynomial composition problem and related problems, we clarify the way in which it relates to the celebrated RSA problem.

The Polynomial Composition Problem in $(\mathbb{Z}/n\mathbb{Z})[X]$ does not imply the RSA Problem, that is, the computation of roots in $\mathbb{Z}/n\mathbb{Z}$. Nevertheless, we exhibit a related problem that we call *Reducible Polynomial Composition Problem* (RPCP) and prove that RPCP $\Leftrightarrow$ RSA problem. In particular, we prove that when $Q(X) = X^q$ then the Polynomial Composition Problem is equivalent to the problem of extracting q^{th} roots in $\mathbb{Z}/n\mathbb{Z}$.

These new problems allow to broaden the view of existing cryptographic constructions. So, we derive a higher level PCP-based protocol of which the Fiat-Shamir [3] and the Guillou-Quisquater protocols [4] are particular instances. Namely, if s denotes the secret, [3] and [4] respectively correspond to the cases $Q(X) = vX^2$ and $Q(X) = vX^\nu$ ($\nu \geq 3$), with $Q(s) = 1$.

The rest of this paper is organized as follows. In Section 2, we formally define the Polynomial Composition Problem and introduce the notations used throughout this paper. The security of the problem and its comparison with the RSA-problem are analyzed in Section 3. Finally, in Section 4 we show how the PCP problem generalizes several zero-knowledge protocols.

2 The Polynomial Composition Problem

We suggest the following problem as a basis for building cryptographic protocols.

Problem 1 (Polynomial Composition Problem). Let P and Q be two polynomials in $(\mathbb{Z}/n\mathbb{Z})[X]$ where n is an RSA modulus. Given polynomials Q and $S := Q(P)$, find P.

Throughout this paper p and q denote the degrees of P and Q, respectively. Let

$$P(X) = \sum_{i=0}^{p} u_i X^i$$

where the u_i's denote the unknowns we are looking for. We assume that

$$Q(Y) = \sum_{j=0}^{q} k_j Y^j$$

is known. Hence,

$$S(X) = \sum_{j=0}^{q} k_j \left(\sum_{i=0}^{p} u_i X^i \right)^j .$$

If, given polynomials $Q'(Y) := Q(Y) - k_0$ and $S'(X) := Q'(P(X))$, an attacker can recover polynomial P then the same attacker can also recover P from polynomials Q and S by first forming polynomials $Q'(Y) = Q(Y) - k_0$ and $S'(X) = S(X) - k_0$. Therefore the problem is reduced to that of decomposing polynomials where Q has no free term, *i.e.*, $Q(Y) = \sum_{j=1}^{q} k_j Y^j$. Similarly, once this has been done, the attacker can divide Q

by a proper constant and replace one of the coefficients k_j by one. Consequently and without loss of generality we restrict our attention to monic polynomials Q with no free term, that is,

$$Q(Y) = Y^q + k_{q-1}Y^{q-1} + \cdots + k_1 Y \quad . \tag{1}$$

Noting that $q = 1$ implies $S = Q(P) = P$, we also assume that $q \geq 2$.

3 Analyzing the Polynomial Composition Problem

As before, let $P(X) = \sum_{i=0}^{p} u_i X^i$ and let $Q(Y) = Y^q + \sum_{j=1}^{q-1} k_j Y^j$. Generalizing Newton's binomial formula and letting $k_q := 1$, we get

$$\begin{aligned} S(X) &= \sum_{j=1}^{q} k_j \left(\sum_{i=0}^{p} u_i X^i \right)^j \\ &= \sum_{t=0}^{pq} \underbrace{\left(\sum_{\substack{1 \leq i_0 + \cdots + i_p \leq q \\ i_1 + 2i_2 + \cdots + p i_p = t}} k_{i_0 + \cdots + i_p} \tfrac{(i_0 + \cdots + i_p)!}{i_0! \ldots i_p!} \, {u_0}^{i_0} \cdots {u_p}^{i_p} \right)}_{:= c_t} X^t \, , \end{aligned} \tag{2}$$

where the second sum is extended over all nonnegative integers i_j satisfying $1 \leq \sum_{j=0}^{p} i_j \leq q$ and $\sum_{j=0}^{p} j\, i_j = t$.

3.1 RSA Problem $\Rightarrow$ Polynomial Composition Problem

We define polynomials $P_0, \ldots, P_{pq} \in (\mathbb{Z}/n\mathbb{Z})[U_0, \ldots, U_p]$ as

$$P_t(U_0, \ldots, U_p) := \sum_{\substack{1 \leq i_0 + \cdots + i_p \leq q \\ i_1 + 2i_2 + \cdots + p i_p = t}} k_{i_0 + \cdots + i_p} \frac{(i_0 + \cdots + i_p)!}{i_0! \ldots i_p!} \, {U_0}^{i_0} \cdots {U_p}^{i_p} - c_t \quad . \tag{3}$$

Note that $P_t(u_0, \ldots, u_p) = 0$ for all $0 \leq t \leq pq$.

Proposition 1. *For all $0 \leq r \leq p$, $P_{pq-r} \in (\mathbb{Z}/n\mathbb{Z})[U_{p-r}, \ldots, U_p]$. Furthermore, for all $1 \leq r \leq p$, P_{pq-r} is of degree exactly one in variable U_{p-r}.*

Proof. For $r = 0$, we have $P_{pq}(U_0, \ldots, U_p) = {U_p}^q - c_{pq}$. For $r = p$, the condition $P_{pq-r} \in (\mathbb{Z}/n\mathbb{Z})[U_{p-r}, \ldots, U_p]$ is trivially satisfied.

Fix r in $[1, p)$. By contradiction, suppose that $P_{pq-r} \notin (\mathbb{Z}/n\mathbb{Z})[U_{p-r}, \ldots, U_p]$. So from Eq. (3), there exists some $i_j \neq 0$ with $0 \leq j \leq p - r - 1$. Since $1 \leq i_0 + \cdots + i_p \leq q$, it follows that $i_1 + 2i_2 + \cdots + p i_p \leq j \cdot 1 + p \cdot (q - 1) <$

$pq-r$; a contradiction because $i_1+2i_2+\cdots+pi_p = pq-r$ for polynomial P_{pq-r}.

Moreover, for all $1 \le r \le p$, P_{pq-r} is of degree one in variable U_{p-r} since we cannot simultaneously have $1 \le \sum_{j=0}^{p} i_j \le q$, $\sum_{j=0}^{p} j\, i_j = pq-r$, and $i_{p-r} \ge 2$. Indeed, $i_{p-r} \ge 2$ implies $i_1+2i_2+\cdots+pi_p \le (p-r)\cdot 2 + p\cdot(q-2) < pq-r$, a contradiction. When $i_{p-r}=1$, $i_1+2i_2+\cdots+pi_p = pq-r$ if $i_p = q-1$ and $i_j = 0$ for all $0 \le (j \ne p-r) \le p-1$. This implies that the only term in U_{p-r} appearing in polynomial P_{pq-r} is $qU_{p-r}\,{U_p}^{q-1}$, whatever the values of variables k_i-s are. □

Corollary 1. *If the value of u_p is known then the Polynomial Composition Problem is trivial.*

Proof. Solving for U_{p-1} the relation $P_{pq-1}(U_{p-1}, u_p) = 0$ (which is a univariate polynomial of degree exactly one in U_{p-1} by virtue of the previous proposition), the value of u_{p-1} is recovered. Next, the root of $P_{pq-2}(U_{p-2}, u_{p-1}, u_p)$ gives the value of u_{p-2} and so on until the value of u_0 is found. □

This means that the Polynomial Composition Problem in $\mathbb{Z}/n\mathbb{Z}$ is easier than the problem of computing q^{th} roots in $\mathbb{Z}/n\mathbb{Z}$ because if an attacker is able to compute a q^{th} modular root (*i.e.*, to solve the RSA Problem) then she can find u_p from $P_{pq}(u_p) = {u_p}^q - c_{pq} = 0$ and then apply the technique explained in the proof of Corollary 1 to recover $u_{p-1}, \ldots, u_0$. In other words,

Corollary 2. *RSA Problem $\Rightarrow$ Polynomial Composition Problem.* □

There is a proposition similar to Proposition 1. It says that once u_0 is known, $u_1, \ldots, u_p$ can be found successively thanks to polynomials $P_1, \ldots, P_p$, respectively.

Proposition 2. *For all $0 \le r \le p$, $P_r \in (\mathbb{Z}/n\mathbb{Z})[U_0, \ldots, U_r]$. Furthermore, for all $1 \le r \le p$, P_r is of degree exactly one in variable U_r.*

Proof. We have $P_0(U_0) = \sum_{j=1}^{q} k_j {U_0}^j - c_0$.

For $r \in [1, p]$, suppose that $P_r \notin (\mathbb{Z}/n\mathbb{Z})[U_0, \ldots, U_r]$. Therefore, $i_1 + 2i_2 + \cdots + pi_p \ge (r+1)\cdot 1 > r$; a contradiction since $i_1 + 2i_2 + \cdots + pi_p = r$. Moreover, we can easily see that $P_r(U_0, \ldots, U_r) = qU_0^{q-1}U_r + \sum_{j=1}^{q-1} k_j\, j\, {U_0}^{j-1} U_r + Q_r(U_0, \ldots, U_{r-1})$ for some polynomial $Q_r \in (\mathbb{Z}/n\mathbb{Z})[U_0, \ldots, U_{r-1}]$. □

3.2 Reducible Polynomial Composition Problem $\Rightarrow$ RSA Problem

The Polynomial Composition Problem *cannot* be equivalent to the RSA Problem. Consider for example the case $p = 2$ and $q = 3$: we have $P(X) = u_2X^2 + u_1X + u_0$ and $Q(X) = X^3 + k_2X^2 + k_1X$, and

$$S(X) = c_6X^6 + c_5X^5 + c_4X^4 + c_3X^3 + c_2X^2 + c_1X + c_0$$

$$\text{with}\begin{cases} c_0 = k_1u_0 + k_2u_0^2 + u_0^3\,, \\ c_1 = k_1u_1 + 2k_2u_0u_1 + 3u_0^2u_1\,, \\ c_2 = k_2u_1^2 + 3u_0u_1^2 + k_1u_2 + 2k_2u_0u_2 + 3u_0^2u_2\,, \\ c_3 = u_1^3 + 2k_2u_1u_2 + 6u_0u_1u_2\,, \\ c_4 = 3u_1^2u_2 + k_2u_2^2 + 3u_0u_2^2\,, \\ c_5 = 3u_1u_2^2\,, \\ c_6 = u_2^3\,. \end{cases}$$

We define the polynomials $P_0(U_0) := k_1U_0 + k_2U_0^2 + U_0^3 - c_0$, $P_1(U_0, U_1) := k_1U_1 + 2k_2U_0U_1 + 3U_0^2U_1 - c_1$, and $P_5(U_1, U_2) := 3U_1U_2^2 - c_5$. Now we first compute the resultant of P_0 and P_1 with respect to variable U_0 and obtain a univariate polynomial in U_1, say $R_0 = \mathrm{Res}_{U_0}(P_0, P_1)$. Next we compute the resultant of R_0 and P_5 with respect to variable U_1 and get a univariate polynomial in U_2, say $R_1 = \mathrm{Res}_{U_1}(R_0, P_5)$. After computation, we get

$$\begin{aligned} R_1(U_2) = 27c_1^3U_2^6 &+ (27c_1^2c_5k_1 - 9c_1^2c_5k_2^2)U_2^4 \\ &+(-4c_5^3k_1^3 + c_5^3k_2^2b^2 - 18c_0c_5^3k_1k_2 + 4c_0c_5^3k_2^3 - 27c_0^2c_5^3)\ . \end{aligned}$$

Since u_2 is a root of both $R_1(U_2)$ and $P_6(U_2)$, u_2 will be a root of their greatest common divisor in $(\mathbb{Z}/n\mathbb{Z})[U_2]$, which is given by

$$\begin{aligned} &(27c_1^2c_5k_1 - 9c_1^2c_5k_2^2)c_6U_2 \\ &\quad + (27c_1^3c_6^2 - 4c_5^3k_1^3 + c_5^3k_1^2k_2^2 - 18c_0c_5^3k_1k_2 + 4c_0c_5^3k_2^3 - 27c_0^2c_5^3)\,, \end{aligned}$$

from which we derive the value of u_2. Once u_2 is known, the values of u_1 and u_0 trivially follow by Corollary 1.

We now introduce a harder problem: the *Reduced* Polynomial Composition Problem in $(\mathbb{Z}/n\mathbb{Z})[X]$.

Problem 2 (Reduced Polynomial Composition Problem (RPCP)). Let P and Q be two polynomials in $(\mathbb{Z}/n\mathbb{Z})[X]$ where n is an RSA modulus. Given Q and the $(\deg(P)+1)$ most significant coefficients of $S := Q(P)$, find P.

Definition 1. *When the Polynomial Composition Problem is equivalent to the Reduced Polynomial Composition Problem, it is said to be* reducible*.*

Equivalently, the Polynomial Composition Problem is reducible when the values of $c_0, \ldots, c_{p(q-1)-1}$ can be derived from $c_{p(q-1)}, \ldots, c_{pq}$ and $k_1, \ldots, k_{q-1}$. This is for example the case when $p = q = 2$, that is, when $P(X) = u_2X^2 + u_1X + u_0$, $Q(X) = X^2 + k_1X$, and

$$S(X) = c_4X^4 + c_3X^3 + c_2X^2 + c_1X + c_0$$

$$\text{with} \begin{cases} c_0 = k_1u_0 + u_0^2\,, \\ c_1 = k_1u_1 + 2u_0u_1\,, \\ c_2 = k_1u_2 + 2u_0u_2 + u_1^2\,, \\ c_3 = 2u_1u_2\,, \\ c_4 = u_2^2\,. \end{cases}$$

An astute algebraic manipulation yields:

$$c_1 = \frac{4c_2c_3c_4 - c_3^3}{8c_4^2} \pmod{n} \quad \text{and} \quad c_0 = \frac{4c_1^2c_4 - c_3^2k_1^2}{4c_3^2} \pmod{n}\ .$$

If follows that we can omit the first two relations (the information included therein is anyway contained in the remaining three as we had just shown) and the decomposition problem amounts to solving the Reduced Polynomial Composition Problem:

$$\begin{cases} c_2 = k_1u_2 + 2u_0u_2 + u_1^2\,, \\ c_3 = 2u_1u_2\,, \\ c_4 = u_2^2\,. \end{cases}$$

Theorem 1. *Reducible Polynomial Composition Problem $\Rightarrow$ RSA Problem.*

Proof. Assume that we are given an oracle $\mathcal{O}^{\mathsf{PCP}}(k_1, \ldots, k_{q-1}; c_0, \ldots, c_{pq})$ which on input polynomials $Q(X) = X^q + \sum_{j=1}^{q-1} k_j X^j$ and $S(X) = \sum_{t=0}^{pq} c_t X^t$ returns the polynomial $P(X) = \sum_{i=0}^{p} u_i X^i$ such that $S(X) = Q(P(X))$. When the polynomial composition is reducible, oracle $\mathcal{O}^{\mathsf{PCP}}$ can be used to compute a q^{th} root of a given $x \in \mathbb{Z}/n\mathbb{Z}$, *i.e.*, compute a y satisfying $y^q \equiv x \pmod{n}$.

1. Choose $p+q-1$ random values $k_1, \ldots, k_{q-1}, c_{p(q-1)}, \ldots, c_{pq-1} \in \mathbb{Z}/n\mathbb{Z}$;
2. Compute $c_0, \ldots, c_{p(q-1)-1}$;
3. Run $\mathcal{O}^{\mathsf{PCP}}(k_1, \ldots, k_{q-1}; c_0, \ldots, c_{pq-1}, x)$;
4. Get $u_0, \ldots, u_p$;
5. Set $y := u_p$ and so $y^q \equiv x \pmod{n}$.

Note that Step 2 can be executed since the composition is supposed to be reducible. Furthermore, note that the values of $c_{pq-1}, \ldots, c_{p(q-1)}$ *uniquely* determine the values of $u_{p-1}, \ldots, u_0$, respectively. Indeed, from Proposition 1, $P_{pq-r}(U_{p-r}, u_{p-r+1}, \ldots, u_p) \in (\mathbb{Z}/n\mathbb{Z})[U_{p-r}]$ is a polynomial of degree exactly one of which u_{p-r} is root, for all $1 \leq r \leq p$. $\square$

3.3 A Practical Criterion

In this paragraph, we present a simple criterion allowing to decide if a given composition problem is reducible.

During the course of proving Proposition 1, we have shown that there exists a polynomial $Q_{pq-r} \in (\mathbb{Z}/n\mathbb{Z})[U_{p-r+1}, \ldots, U_p]$ such that

$$P_{pq-r}(U_{p-r}, \ldots, U_p) = qU_{p-r}{U_p}^{q-1} + Q_{pq-r}(U_{p-r+1}, \ldots, U_p)$$

for all $1 \leq r \leq p$. From $c_{pq} = (u_p)^q$, we then infer:

$$u_{p-r} = \frac{-Q_{pq-r}(u_{p-r+1}, \ldots, u_p)}{q\, c_{pq}}\, u_p\,, \quad (1 \leq r \leq p)\ . \tag{4}$$

Using Eq. (4), for $r = 1, \ldots, p$, we now iteratively compute $u_{p-1}, \ldots, u_0$ as a polynomial function in u_p. We let Υ_{p-r} denote this polynomial function, *i.e.*, $u_{p-r} = \Upsilon_{p-r}(u_p)$ for all $1 \leq r \leq p$. We then respectively replace $u_0, \ldots, u_{p-1}$ by $\Upsilon_0(u_p), \ldots, \Upsilon_{p-1}(u_p)$ in the expressions of $c_0, \ldots, c_{pq-p-1}$. If, for each c_i $(0 \leq i \leq pq - p - 1)$, the powers of u_p cancel thanks to $(u_p)^{q-1} = c_{pq}$ then the problem is reducible.

We illustrate the technique with the example $P(X) = u_3X^3 + u_2X^2 + u_1X + u_0$ and $Q(Y) = Y^3$. Then $S(X) = \sum_{t=0}^{9} c_t X^t$ with

$$\begin{cases} c_0 = u_0^3\,, \\ c_1 = 3u_0^2u_1\,, \\ c_2 = 3u_0^2u_2 + 3u_0u_1^2\,, \\ c_3 = 3u_0^2u_3 + 6u_0u_1u_2 + u_1^3\,, \\ c_4 = 6u_0u_1u_3 + 3u_0u_2^2 + 3u_1^2u_2\,, \\ c_5 = 6u_0u_2u_3 + 3u_1^2u_3 + 3u_1u_2^2\,, \\ c_6 = 3u_0u_3^2 + 6u_1u_2u_3 + u_2^3\,, \\ c_7 = 3u_1u_3^2 + 3u_2^2u_3\,, \\ c_8 = 3u_2u_3^2\,, \\ c_9 = u_3^3\,. \end{cases}$$

From the respective expressions of c_8, c_7 and c_6, we successively find

$$\Upsilon_2(u_3) = \frac{c_8}{3c_9}\,u_3\,, \quad \Upsilon_1(y_3) = \frac{3c_7c_9 - c_8}{9c_9^2}\,u_3\,, \quad \text{and}$$
$$\Upsilon_0(u_3) = \frac{27c_6c_9^2 - 6c_8(3c_7c_9 - c_8^2) - c_8^3}{81c_9^3}\,u_3 \enspace .$$

Since $c_0, \ldots, c_5$ are homogeneous in u_0, u_1, u_2, u_3 and of degree three, they can be evaluated by replacing u_0, u_1, u_2 by $\Upsilon_0(u_3), \Upsilon_1(u_3), \Upsilon_2(u_3)$, respectively, and then replacing $(u_3)^3$ by c_9. Consequently, the composition is reducible: the values of $c_0, \ldots, c_5$ can be inferred from $c_6, \ldots, c_9$ and the problem amounts to computing cubic roots in $\mathbb{Z}/n\mathbb{Z}$.

This is not fortuitous and can easily be generalized as follows.

Corollary 3. *For $Q(Y) = Y^q$, the Polynomial Composition Problem in $\mathbb{Z}/n\mathbb{Z}$ is equivalent to the RSA Problem, i.e. to the problem of extracting q^{th} roots in $\mathbb{Z}/n\mathbb{Z}$.*

Proof. From Eq. (2), it follows that $S(X) = \sum_{t=0}^{pq} c_t X^t$ with

$$c_t = \sum_{\substack{i_0+\cdots+i_p=q \\ i_1+2i_2+\cdots+pi_p=t}} \frac{q!}{i_0!\cdots i_p!}\, {u_0}^{i_0} \cdots {u_p}^{i_p}\,,$$

which is homogeneous in $u_0, \ldots, u_p$ and of degree $i_0+\cdots+i_p = q$. Moreover since by induction, for $1 \le r \le p$, $\Upsilon_{p-r}(u_p) = K_{p-r} \cdot u_p$ for some constant K_{p-r}, the corollary follows. □

4 Cryptographic Application

4.1 A Simple PCP-Based Identification Protocol

For setting up the system, a trusted third party (TTP) selects and publishes an RSA modulus n. Each user chooses two polynomials P, Q in $(\mathbb{Z}/n\mathbb{Z})[X]$ and computes $S = Q(P)$. She then registers Q and S as her public key with the TTP. Her secret key is P.

To prove knowledge of P, a user (referred to as the prover) executes ℓ times the following protocol.

- The prover selects a random $r \in \mathbb{Z}/n\mathbb{Z}$, evaluates $c = S(r)$ and sends c to the verifier.
- The verifier sends to the prover a random bit b.
- If $b = 0$, the prover reveals $t = r$ and the verifier checks that $S(t) = c$.
  If $b = 1$, the prover reveals $t = P(r)$ and the verifier checks that $Q(t) = c$.

Fig. 1. Basic Protocol.

4.2 Improvements

Efficiency can be increased by using the following trick. The user chooses ν polynomials $P_1, \dots, P_{\nu-1}, Q$ in $(\mathbb{Z}/n\mathbb{Z})[X]$, with $\nu \geq 3$. Her secret key is the set $\{P_1, \dots, P_{\nu-1}\}$ while her public key is the set $\{S_0 = Q, S_1 = Q(P_{\nu-1}), S_2 = Q(P_{\nu-1}(P_{\nu-2})), \dots, S_j = Q(P_{\nu-1}(\dots(P_{\nu-j}))), \dots, S_{\nu-1} = Q(P_{\nu-1}(\dots(P_1)))\}$.

The resulting protocol is given in Fig. 2.

- The prover selects a random $r \in \mathbb{Z}/n\mathbb{Z}$, evaluates $c = S_{\nu-1}(r)$ and sends c to the verifier.
- The verifier sends to the prover a random integer $0 \leq b \leq \nu - 1$.
- If $b = 0$, the prover reveals $t = r$ and the verifier checks that $S_{\nu-1}(t) = c$.
  If $b \neq 0$, the prover reveals $t = P_b(\dots(P_1(r)))$ and the verifier checks that $S_{\nu-b-1}(t) = c$.

Fig. 2. Improved Protocol.

4.3 Comparison with Other Identification Protocols

It is interesting to note that our basic protocol (Fig. 1) coincides with the (simplified) Fiat-Shamir protocol [3] (see also [5, Protocol 10.24]) when $P(X) = sX$ and $Q(X) = vX^2$ where $vs^2 \equiv 1 \pmod{n}$.

The multiple polynomials variant (Fig. 2) may be seen as a generalization of the Guillou-Quisquater protocol [4] by taking $P_1(X) = P_2(X) = \cdots = P_{\nu-1}(X) = sX$ where s is a secret value and $Q(X) = vX^\nu$ so that $vs^\nu \equiv 1 \pmod{n}$. Indeed, in this case we have $P_{\nu-1}(\ldots(P_{\nu-j}(X))) = s^j X$ and hence $S_j(X) = v^{1-j}X^\nu$.

5 Conclusion

This paper introduced the Polynomial Composition Problem (PCP) and the related Reducible Polynomial Composition Problem (RPCP). Relations among these two problems and the RSA Problem were explored. Finally, concrete zero-knowledge protocols were given as particular instances of PCP-based constructs.

References

1. Henri Cohen. *A Course in Computational Algebraic Number Theory*, GTM 138, Springer-Verlag, 1993.
2. Don Coppersmith, Matthew Franklin, Jacques Patarin, and Michael Reiter. Low-exponent RSA with related messages. In U. Maurer, ed., *Advances in Cryptology – EUROCRYPT '96*, LNCS 1070, pp. 1–9, Springer-Verlag, 1996.
3. Amos Fiat, and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, ed., *Advances in Cryptology – CRYPTO '86*, LNCS 263, pp. 186–194, Springer-Verlag, 1987.
4. Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In C. Günther, ed., *Advances in Cryptology – EUROCRYPT '88*, LNCS 330, pp. 123–128, Springer-Verlag, 1988.
5. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, 1997.

A Mathematical Background

Let $\mathcal{R}$ be an integral domain with quotient field $\mathbb{K}$.

Definition 2. *Given two polynomials $A, B \in \mathcal{R}[X]$, the* resultant of A and B*, denoted by* $\operatorname{Res}(A, B)$*, is defined as*

$$\operatorname{Res}(A, B) = (a_m)^n \, (b_n)^m \prod_{1 \le i \le m, 1 \le j \le n} (\alpha_i - \beta_j) \tag{5}$$

if $A(X) = a_m \prod_{1\le i\le m}(X-\alpha_i)$ *and* $B(X) = b_n \prod_{1\le j\le n}(X-\beta_j)$ *are the decompositions of* A *and* B *in the algebraic closure of* $\mathbb{K}$.

From this definition, we see that $\mathrm{Res}(A,B)=0$ if and only if polynomials A and B have a common root (in $\overline{\mathbb{K}}$); hence if and only if A and B have a (non-trivial) common factor. Equivalently, we have

$$\mathrm{Res}(A,B) = (a_m)^n \prod_{1\le i\le m} B(\alpha_i) = (b_n)^m \prod_{1\le j\le n} A(\beta_j) \ .$$

The resultant $\mathrm{Res}(A,B)$ can be evaluated without knowing the decomposition of A and B. Letting $A(X) = \sum_{1\le i\le m} a_i X^i$ and $B(X) = \sum_{1\le j\le n} b_j X^j$, we have

$$\mathrm{Res}(A,B) = \det \begin{pmatrix} a_m & a_{m-1} & \dots & a_0 & 0 & \dots & 0 \\ 0 & a_m & a_{m-1} & \dots & a_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & a_m & a_{m-1} & \dots & a_0 \\ b_n & b_{n-1} & \dots & b_0 & 0 & \dots & 0 \\ 0 & b_n & b_{n-1} & \dots & b_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & b_n & b_{n-1} & \dots & b_0 \end{pmatrix} \begin{array}{l} \left.\vphantom{\begin{matrix} a\\a\\a\\a \end{matrix}}\right\} n \text{ rows} \\ \left.\vphantom{\begin{matrix} a\\a\\a\\a \end{matrix}}\right\} m \text{ rows} \end{array} .$$

This clearly shows that $\mathrm{Res}(A,B) \in \mathcal{R}$.

A *multivariate polynomial* $A \in \mathcal{R}[X_1,\dots,X_k]$ (with $k \ge 2$) may be viewed as an univariate polynomial in $\mathcal{R}[X_1,\dots,X_{k-1}][X_k]$. Consequently, it makes sense to compute the resultant of two multivariate polynomials with respect to one variable, say X_k. If $A, B \in \mathcal{R}[X_1,\dots,X_k]$, we let $\mathrm{Res}_{X_k}(A,B)$ denote the resultant of A and B with respect to X_k.

Lemma 1. *Let* $A, B \in \mathcal{R}[X_1,\dots,X_k]$ *(with* $k\ge 2$*). Then* $(\alpha_1,\dots,\alpha_k)$ *is a common root (in* $\overline{\mathbb{K}}$*) of* A *and* B *if and only if* $(\alpha_1,\dots,\alpha_{k-1})$ *is a root of* $\mathrm{Res}_{X_k}(A,B)$.

B Additional Examples

B.1 The case $p = 3$ and $q = 2$

Using the previous notations and simplifications, we write $P(X) = u_3X^3 + u_2X^2 + u_1X + u_0$ and $Q(Y) = Y^2 + k_1Y$. Expressing the c_i's we get:

$$\begin{cases} c_0 = k_1u_0 + u_0^2\,, \\ c_1 = k_1u_1 + 2u_0u_1\,, \\ c_2 = u_1^2 + k_1u_2 + 2u_0u_2\,, \\ c_3 = 2u_1u_2 + k_1u_3 + 2u_0u_3\,, \\ c_4 = u_2^2 + 2u_1u_3\,, \\ c_5 = 2u_2u_3\,, \\ c_6 = u_3^2\,. \end{cases}$$

Now using the criterion of § 3.3, we find $u_2 = \frac{c_5}{2c_6}u_3$, $u_1 = Vu_3$, and $u_0 = -\frac{k_1^2}{4} + Lu_3$ with $V := \frac{4c_4c_6 - c_5^2}{8c_6^2}$ and $L := \frac{8c_3c_6^2 - c_5(4c_4c_6 - c_5^2)}{16c_6^3}$. Hence, we derive:

$$c_2 = c_6V^2 + c_5L, \quad c_1 = 2c_6LV\,, \quad \text{and } c_0 = -\frac{k_1^2}{4} + L^2c_6 \;\;.$$

Being reducible, this proves that solving the polynomial decomposition problem for $p = 3$ and $q = 2$ amounts to computing square roots in $\mathbb{Z}/n\mathbb{Z}$.

B.2 The case $p = 3$ and $q = 3$

We have $P(X) = u_3X^3 + u_2X^2 + u_1X + u_0$ and $Q(X) = X^3 + k_2X^2 + k_1X$. Defining polynomials P_i as in Eq. (3), we successively compute $R_0 := \mathrm{Res}_{U_0}(P_0, P_1)$, $R_1 := \mathrm{Res}_{U_1}(R_0, P_7)$, and $R_2 = \mathrm{Res}_{U_2}(R_1, P_8)$ wherefrom

$$\begin{aligned} R_2(u_3) = {} & 19683c_1^3u_3^{18} + (-6561c_1^2c_7k_2^2 + 19683c_1^2c_7k_1)u_3^{16} \\ & + (2187c_1^2c_8^2k_2^2 - 6561c_1^2c_8^2k_1)u_3^{13} \\ & + (2916c_0c_7^3k_2^3 + 729c_7^3k_1^2k_2^2 - 13122c_0c_7^3k_2k_1 - 2916c_7^3k_1^3 \\ & \qquad - 19683c_7^3c_0^2)u_3^{12} \\ & + (-2916c_0c_7^2c_8^2k_2^3 - 729c_8^2c_7^2k_2^2k_1^2 + 13122c_8^2c_7^2c_0k_2k_1 + 2916c_8^2c_7^2k_1^3 \\ & \qquad + 19683c_8^2c_7^2c_0^2)u_3^9 \\ & + (972c_8^4c_7c_0k_2^3 + 243c_8^4c_7k_1^2k_2^2 - 4374c_8^4c_7c_0k_1k_2 - 972c_8^4c_7k_1^3 \\ & \qquad - 6561c_8^4c_7c_0^2)u_3^6 \\ & + (-108c_8^6c_0k_2^3 - 27c_8^6k_1^2k_2^2 + 486c_8^6c_0k_1k_2 + 108c_8^6k_1^3 + 729c_8^6c_0^2)u_3^3 \\ = {} & 0\;\;. \end{aligned}$$

So, we obtain the value of u_3 by exploiting the additional relation $c_9 = u_3^3$ and hence the values of u_2, u_1, and u_0.

Remark that if we choose $k_1 = k_2^2/3$ then the terms in u_3^{16} $(= c_9^5\, u_3)$ and in u_3^{13} $(= c_9^4\, u_3)$ disappear and consequently the value of u_3 cannot be recovered. In this case, the criterion shows again that the problem is equivalent to that of computing cubic roots in $\mathbb{Z}/n\mathbb{Z}$.