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

Marc Joye[1], David Naccache[2], and Stéphanie Porte[3]

[1] Thomson R&D, Security Competence Center
1 avenue de Belle Fontaine, 35576 Cesson-Sévigné Cedex, France
marc.joye@thomson.net
[2] Ecole normale supérieure, Département d'informatique
45 rue d'Ulm, 75230 Paris Cedex 05, France
david.naccache@ens.fr
[3] Smart Consulting
2 rue Louis Vignol, 13600 La Ciotat, France
stef.porte@gmail.com

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

Keywords: Polynomial composition, zero-knowledge protocols, Fiat-Shamir protocol, Guillou-Quisquater protocol, smart cards.

1 Introduction

Smart cards play an active role in security systems. Their salient features make them attractive in numerous applications, including — to name a few — the areas of banking, telephone, health, pay TV, home computers, and communication networks.

One of the primary use of smart cards resides in authentication. There exist basically two families of methods currently used for authenticating purposes. The first family relies on secret-key one-way functions while the second one makes use of public-key techniques. Both families have their own advantages. This paper focuses on the second family. In particular, we study zero-knowledge techniques. In a typical scenario, the smart card, characterized by a set of credentials, plays the role of the proving entity.

The first practical zero-knowledge protocol is due to Fiat and Shamir [3]. Remarkably, the protocol is rather efficient, computation-wise. It amounts to at most two modular multiplications per interaction. However, in order to reach a level of confidence of $(1 - 2^{-k})$, the basic protocol has to be repeated k times — a typical value for k is $k = 40$. In order to reduce the communication overhead, there is also a multiple-key protocol, at the expense of more key material. Another zero-knowledge protocol well suited to smart-card applications is the

Guillou-Quisquater protocol [4] (a.k.a. GQ protocol). The GQ protocol features small storage requirements and needs a single interaction.

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

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

Most public-key cryptographic schemes base their security on the difficulty of solving a hard mathematical problem. Given that the number of hard problems harnessable to cryptographic applications is rather limited, the investigation of new problems is of central importance in cryptography. To understand the Polynomial Composition Problem and its variants, we explore in the following sections the way in which the PCP 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. In particular, we prove that when $\mathscr{Q}(X) = X^q$ 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 us to broaden the view of existing cryptographic constructions. Namely, we describe a general PCP-based zero-knowledge protocol of which the Fiat-Shamir and the Guillou-Quisquater protocols are particular instances. As will be seen later, if s denotes the secret, they respectively correspond to the cases $\mathscr{Q}(X) = vX^2$ and $\mathscr{Q}(X) = vX^v$ ($v \geq 3$), with $\mathscr{Q}(s) = 1$.

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 hardness of the problem and its comparison with RSA are analyzed in Section 3. Finally, in Section 4, we show that the PCP allows one to generalize 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 (PCP)). Let $\mathscr{P}$ and $\mathscr{Q}$ be two polynomials in $(\mathbb{Z}/n\mathbb{Z})[X]$ where n is an RSA modulus. Given polynomials $\mathscr{Q}$ and $\mathscr{S} := \mathscr{Q}(\mathscr{P})$, find $\mathscr{P}$.

Throughout this paper p and q denote the degrees of $\mathscr{P}$ and $\mathscr{Q}$, respectively. Let

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

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

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

is known. Hence,

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

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

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

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

3 Analyzing the Polynomial Composition Problem

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

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

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 ⇒ Polynomial Composition Problem

We define polynomials $\mathscr{P}_0, \ldots, \mathscr{P}_{pq} \in (\mathbb{Z}/n\mathbb{Z})[U_0, \ldots, U_p]$ as

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

Note that $\mathscr{P}_t(u_0, \ldots, u_p) = 0$ for all $0 \leq t \leq pq$.

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

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

Fix r in $[1, p)$. By contradiction, suppose that $\mathscr{P}_{pq-r} \notin (\mathbb{Z}/n\mathbb{Z})[U_{p-r}, \ldots, U_p]$. 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 $i_1 + 2i_2 + \cdots + pi_p \leq j \cdot 1 + p \cdot (q-1) < pq - r$; a contradiction because $i_1 + 2i_2 + \cdots + pi_p = pq - r$ for polynomial $\mathscr{P}_{pq-r}$.

Moreover, for all $1 \leq r \leq p$, $\mathscr{P}_{pq-r}$ is of degree one in variable U_{p-r} since we cannot simultaneously have $1 \leq \sum_{j=0}^{p} i_j \leq q$, $\sum_{j=0}^{p} j\, i_j = pq - r$, and $i_{p-r} \geq 2$. Indeed, $i_{p-r} \geq 2$ implies $i_1 + 2i_2 + \cdots + pi_p \leq (p-r) \cdot 2 + p \cdot (q-2) < pq - r$, 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 $0 \leq (j \neq p - r) \leq p - 1$. This implies that the only term in U_{p-r} appearing in polynomial $\mathscr{P}_{pq-r}$ is $q U_{p-r}\, {U_p}^{q-1}$, whatever the values of variables k_i's are. □

Corollary 1. *If the value of* u_p *is known then the Polynomial Composition Problem can be solved in time* $O(p)$.

Proof. Solving for U_{p-1} the relation $\mathscr{P}_{pq-1}(U_{p-1}, u_p) = 0$ (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 $\mathscr{P}_{pq-2}(U_{p-2}, u_{p-1}, u_p)$ gives the value of u_{p-2} and so on until the value of u_0 is found.

Note that the running time of the resolution process is $O(p)$ and is thus exponential in the bit-length of p. □

This means that for low degree polynomials, 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 $\mathscr{P}_{pq}(u_p) = {u_p}^q - c_{pq} = 0$ 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 $\mathscr{P}_1, \ldots, \mathscr{P}_p$, respectively.

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

Proof. We have $\mathscr{P}_0(U_0) = \sum_{j=1}^{q} k_j {U_0}^j - c_0$.

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

3.2 Reducible Polynomial Composition Problem ⇒ 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 $\mathscr{P}(X) = u_2X^2 + u_1X + u_0$ and $\mathscr{Q}(X) = X^3 + k_2X^2 + k_1X$, and

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

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

We define the polynomials $\mathscr{P}_0(U_0) := k_1U_0 + k_2U_0^2 + U_0^3 - c_0$, $\mathscr{P}_1(U_0, U_1) := k_1U_1 + 2k_2U_0U_1 + 3U_0^2U_1 - c_1$, and $\mathscr{P}_5(U_1, U_2) := 3U_1U_2^2 - c_5$. Now we first compute the resultant of $\mathscr{P}_0$ and $\mathscr{P}_1$ with respect to variable U_0 and obtain a univariate polynomial in U_1, say $\mathscr{R}_0 = \mathrm{Res}_{U_0}(\mathscr{P}_0, \mathscr{P}_1)$. Next we compute the resultant of $\mathscr{R}_0$ and $\mathscr{P}_5$ with respect to variable U_1 and get a univariate polynomial in U_2, say $\mathscr{R}_1 = \mathrm{Res}_{U_1}(\mathscr{R}_0, \mathscr{P}_5)$. After computation, we get

$$\begin{aligned} \mathscr{R}_1(U_2) = 27c_1^3U_2^6 &+ (27c_1^2c_5k_1 - 9c_1^2c_5k_2^2)U_2^4 \\ &+(-4c_5^3k_1^3 + c_5^3k_2^2b^2 - 18c_0c_5^3k_1k_2 + 4c_0c_5^3k_2^3 - 27c_0^2c_5^3)\ . \end{aligned}$$

Since u_2 is a root of both $\mathscr{R}_1(U_2)$ and $\mathscr{P}_6(U_2) := U_2^3 - c_6$, 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 $\mathscr{P}$ and $\mathscr{Q}$ be two polynomials in $(\mathbb{Z}/n\mathbb{Z})[X]$ where n is an RSA modulus. Given $\mathscr{Q}$ and the $\deg(\mathscr{P}) + 1$ most significant coefficients of $\mathscr{S} := \mathscr{Q}(\mathscr{P})$, find $\mathscr{P}$.

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 $\mathscr{P}(X) = u_2X^2 + u_1X + u_0$, $\mathscr{Q}(X) = X^2 + k_1X$, and

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

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

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 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 $O^{\mathsf{PCP}}(k_1, \ldots, k_{q-1}; c_0, \ldots, c_{pq})$ which on input polynomials $\mathscr{Q}(X) = X^q + \sum_{j=1}^{q-1} k_jX^j$ and $\mathscr{S}(X) = \sum_{t=0}^{pq} c_tX^t$ returns the polynomial $\mathscr{P}(X) = \sum_{i=0}^{p} u_iX^i$ such that $\mathscr{S}(X) = \mathscr{Q}(\mathscr{P}(X))$. When the polynomial composition is reducible, oracle O^{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 $O^{\mathsf{PCP}}(k_1, \ldots, k_{q-1}; c_0, \ldots, c_{pq-1}, x)$;
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,

$$\mathscr{P}_{pq-r}(U_{p-r}, u_{p-r+1}, \ldots, u_p) \in (\mathbb{Z}/n\mathbb{Z})[U_{p-r}]$$

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

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

$$\mathscr{P}_{pq-r}(U_{p-r}, \ldots, U_p) = qU_{p-r}U_p{}^{q-1} + \mathscr{Q}_{pq-r}(U_{p-r+1}, \ldots, U_p)$$

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

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

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

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

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 $\mathcal{Q}(Y) = Y^q$, 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 $\mathcal{S}(X) = \sum_{t=0}^{pq} c_t X^t$ 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 Applications

Loosely speaking, a zero-knowledge protocol allows a prover to demonstrate the knowledge of a secret without revealing any useful information about the secret. We show how to construct such a protocol thanks to composition of polynomials.

4.1 A PCP-Based Zero-Knowledge Protocol

A trusted third party selects and publishes an RSA modulus n. Each prover $\mathcal{P}$ chooses two polynomials $\mathscr{P}, \mathscr{Q}$ in $(\mathbb{Z}/n\mathbb{Z})[X]$ and computes $\mathscr{S} = \mathscr{Q}(\mathscr{P})$. $\{\mathscr{Q}, \mathscr{S}\}$ is $\mathcal{P}$'s public key given to the verifier $\mathcal{V}$ so as to ascertain $\mathcal{P}$'s knowledge of the secret key $\mathscr{P}$.

Execute ℓ times the following protocol:

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

PCP-Based Protocol.

4.2 Improvements

Efficiency can be increased by using the following trick:

$\mathcal{P}$ chooses ν polynomials $\mathscr{P}_1, \ldots, \mathscr{P}_{\nu-1}, \mathscr{Q}$ in $(\mathbb{Z}/n\mathbb{Z})[X]$, with $\nu \ge 3$. Her secret key is the set $\{\mathscr{P}_1, \ldots, \mathscr{P}_{\nu-1}\}$ while her public key consists of the set $\{\mathscr{S}_0 =$

$\mathcal{Q}, \mathcal{S}_1 = \mathcal{Q}(\mathcal{P}_{v-1}), \mathcal{S}_2 = \mathcal{Q}(\mathcal{P}_{v-1}(\mathcal{P}_{v-2})), \ldots, \mathcal{S}_j = \mathcal{Q}(\mathcal{P}_{v-1}(\ldots(\mathcal{P}_{v-j}))), \ldots, \mathcal{S}_{v-1} = \mathcal{Q}(\mathcal{P}_{v-1}(\ldots(\mathcal{P}_1)))\}$.

The protocol is shown below:

- $\mathcal{P}$ selects a random $r \in \mathbb{Z}/n\mathbb{Z}$
- $\mathcal{P}$ evaluates $c = \mathcal{S}_{v-1}(r)$ and sends c to $\mathcal{V}$.
- $\mathcal{V}$ sends to $\mathcal{P}$ a random integer $0 \leq b \leq v-1$.
- If $b = 0$, $\mathcal{P}$ reveals $t = r$ and $\mathcal{V}$ checks that $\mathcal{S}_{v-1}(t) = c$.
- If $b \neq 0$, $\mathcal{P}$ reveals $t = \mathcal{P}_b(\ldots(\mathcal{P}_1(r)))$ and $\mathcal{V}$ checks that $\mathcal{S}_{v-b-1}(t) = c$.

Nested PCP Protocol.

4.3 Relations with Other Zero-Knowledge Protocols

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

The nested variant may be seen as a generalization of the Guillou-Quisquater protocol [4] by taking $\mathcal{P}_1(X) = \mathcal{P}_2(X) = \cdots = \mathcal{P}_{v-1}(X) = sX$ where s is a secret value and $\mathcal{Q}(X) = vX^v$ so that $vs^v \equiv 1 \pmod{n}$. Indeed, in this case we have $\mathcal{P}_{v-1}(\ldots(\mathcal{P}_{v-j}(X))) = s^jX$ and hence $\mathcal{S}_j(X) = v^{1-j}X^v$.

An interesting research direction would be to extend the above protocols to Dickson polynomials.

5 Conclusion

This paper introduced the Polynomial Composition Problem (PCP) and the related Reducible Polynomial Composition Problem (RPCP). Relations between these two problems and the RSA Problem were explored. Further, two concrete zero-knowledge protocols suited to smart-card applications were given as particular instances of PCP-based constructs.

Acknowledgments We are grateful to Jesper Buus Nielsen (ETHZ) for attracting our attention to an important detail in the proof of Corollary 1. We are also grateful to the anonymous referees for useful comments.

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 $\mathscr{A}, \mathscr{B} \in \mathcal{R}[X]$, the* resultant of $\mathscr{A}$ and $\mathscr{B}$, *denoted by* $\mathrm{Res}(\mathscr{A}, \mathscr{B})$, *is defined as*

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

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

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

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

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

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

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

A *multivariate polynomial* $\mathscr{A} \in \mathcal{R}[X_1, \dots, X_k]$ (with $k \ge 2$) may be viewed as a 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 $\mathscr{A}, \mathscr{B} \in \mathcal{R}[X_1, \ldots, X_k]$, we let $\mathrm{Res}_{X_k}(\mathscr{A}, \mathscr{B})$ denote the resultant of $\mathscr{A}$ and $\mathscr{B}$ with respect to X_k.

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

B Additional Examples

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

Using the previous notations and simplifications, we write $\mathscr{P}(X) = u_3X^3 + u_2X^2 + u_1X + u_0$ and $\mathscr{Q}(Y) = Y^2 + k_1Y$. 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 PCP 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 $\mathscr{P}(X) = u_3X^3 + u_2X^2 + u_1X + u_0$ and $\mathscr{Q}(X) = X^3 + k_2X^2 + k_1X$. Defining polynomials $\mathscr{P}_i$ as in Eq. (3), we successively compute $\mathscr{R}_0 := \mathrm{Res}_{U_0}(\mathscr{P}_0, \mathscr{P}_1)$, $\mathscr{R}_1 := \mathrm{Res}_{U_1}(\mathscr{R}_0, \mathscr{P}_7)$, and $\mathscr{R}_2 = \mathrm{Res}_{U_2}(\mathscr{R}_1, \mathscr{P}_8)$ wherefrom

$$\begin{aligned} \mathscr{R}_2(u_3) = {} & 19683c_1^3u_3^{18} + (-6561c_1^2c_7k_2^2 + 19683c_1^2c_7k_1)u_3^{16} \\ & + (2187c_1^2c_8^2k_2^2 - 6561c_1^2c_8^2k_1)u_3^{13} \\ & + (2916c_0c_7^3k_2^3 + 729c_7^3k_1^2k_2^2 - 13122c_0c_7^3k_2k_1 - 2916c_7^3k_1^3 \\ & \qquad - 19683c_7^3c_0^2)u_3^{12} \\ & + (-2916c_0c_7^2c_8^2k_2^3 - 729c_8^2c_7^2k_2^2k_1^2 + 13122c_8^2c_7^2c_0k_2k_1 + 2916c_8^2c_7^2k_1^3 \\ & \qquad + 19683c_8^2c_7^2c_0^2)u_3^9 \\ & + (972c_8^4c_7c_0k_2^3 + 243c_8^4c_7k_1^2k_2^2 - 4374c_8^4c_7c_0k_1k_2 - 972c_8^4c_7k_1^3 \\ & \qquad - 6561c_8^4c_7c_0^2)u_3^6 \\ & + (-108c_8^6c_0k_2^3 - 27c_8^6k_1^2k_2^2 + 486c_8^6c_0k_1k_2 + 108c_8^6k_1^3 + 729c_8^6c_0^2)u_3^3 \\ = {} & 0\ . \end{aligned}$$

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.

Note 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 $\mathbb{Z}/n\mathbb{Z}$.