PKP-Based Signature Scheme

Jean-Charles FAUGÈRE [1], Eliane KOUSSA [2], Gilles MACARIO-RAT[3], Jacques PATARIN [4], and Ludovic PERRET[5]

[1,5]INRIA and Sorbonne Universities/UPMC Uni Paris 6, jean-charles.faugere@inria.fr , Ludovic.Perret@lip6.fr
[2] Versailles Laboratory of Mathematics, UVSQ, EJKoussa@outlook.com
[3]Orange, gilles.macariorat@orange.com
[4]Versailles Laboratory of Mathematics, UVSQ, CNRS, University of Paris-Saclay, jpatarin@club-internet.fr

Abstract: In this document, we introduce PKP-DSS: a Digital Signature Scheme based on the so-called Permuted Kernel Problem (PKP) [Sha89]. PKP is an NP-complete [GJ79] algebraic problem that consists of finding a kernel vector with particular entries for a publicly known matrix. It's simple, and needs only basic linear algebra. Hence, this problem was used to develop the first Identification Scheme (IDS) which has an efficient implementation on low-cost smart cards. We construct PKP-DSS from a Zero-Knowledge Identification Scheme (ZK-IDS) based on PKP [Sha89]. We derive the signature scheme PKP-DSS by using the traditional Fiat-Shamir (FS) transform [FS86]. Thus, PKP-DSS has a security that can be provably reduced, in the *(classical) random oracle model*, to essentially the hardness of random instances of PKP.
Following the State-of-the-art attacks of PKP, we propose several sets of parameters for different security levels. Each parameter set arises signatures of length smaller than the other signatures derived from Zero-Knowledge identification schemes. In particular, PKP-DSS-128 gives a signature size approximately about 14 KBytes for 128 bits of classical security, while the best known signature schemes built from a ZK-IDS (such as MQDSS [CHR$^+$18], Picnic [CDG$^+$17],...) give bigger signatures ($\geq$ 16 KB).

Keywords: public-key cryptography, post-quantum cryptography, Fiat-Shamir, 5-pass identification scheme, Permuted Kernel Problem.

1 Introduction

The construction of large quantum computers would break all public-key cryptographic schemes in use today based on the traditional number-theoretic problems: the discrete logarithm (DLOG) and the integer factorization (FACT), like RSA public key encryption and Diffie-Hellman key exchange. Despite the fact that it isn't clear when and

even if enormous quantum computations would be feasible, it is important to anticipate a technological breakthrough and design new public key cryptosytems that are resistant to quantum attacks.

Therefore, the effort to develop new schemes is now being intensified, and the most significant sign is certainly the standardization process initiated by the American organization NIST (`https://www.nist.gov/`).

Due to the call for post quantum standards of the NIST, there has been renewed interest in the transformed Zero-Knowledge Identification Schemes into Digital Signatures Schemes (DSS) via the Fiat-Shamir paradigm [FS86]. This transformation method is important since it yields to efficient signature schemes in terms of minimal and sufficient security assumptions.

Particularly, we are interested in the post-quantum cryptographic schemes which belongs to the post quantum branch whose security relies on the fact that there is no quantum algorithms known to solve NP-Complete problems [BBBV97]. Namely, the Permuted Kernel Problem: the problem of finding a permutation of a known vector such that the resulting vector is in the kernel of a given matrix.

Here, we study the application in cryptography of the PKP problem over a finite field. We are essentially concerned in this problem because it can be used to build a post quantum signature scheme based on the hardness of solving random instances of PKP. It is an old-time combinatorial NP-Complete problem. It requires simple operations which involve basic linear algebra computations. For a little long time no new attacks on PKPwere reported which makes the construction of schemes based on hard instances of this problem more applicable.

Previous work and State-of-the-art. Since quantum computers are known to be incapable to solve *NP*-Complete problems [BBBV97], the Zero-knowledge Identification schemes (ZK-IDS), based on such problems, are very interesting nowadays. The Fiat-Shamir transform [FS86] is a technique to convert a zero knowledge authentication scheme (ZK scheme) into a signature scheme. Its principle is to turn the exchanged elements during authentication into a signature [NPV12, SSH11].

Here, we focus on recent signature schemes built from Zero-knowledge Identification schemes by applying the Fiat-Shamir transform. Lately, a secure signature scheme was introduced in [CHR$^+$18] with concrete parameters and detailed implementation. It has opened the doors to consider other Identifaction schemes based on NP-Complete problems.

In [CHR$^+$18], a new multivariate-based digital signature scheme called MQDSS and utilizing the Fiat-Shamir paradigm was presented. MQDSS is based on the MQ problem *i.e.*the problem of solving systems of multivariate quadratic polynomials.

The authors of [CHR$^+$18] have introduced MQDSS-31-48 for a security of 128 bits (*resp.* MQDSS-31-64 for a security of 192 bits) coming with a public key of 46 Bytes (*resp.* 64), a secret key of 16 Bytes (*resp.* 24) and a signature size of approximately 16.15 K-Bytes (*resp.* 33.23).

As well and besides zero knowledge proof, Picnic [CDG$^+$17] is a digital signature scheme whose security relies on hash functions, symmetric cryptography, and block ciphers. In Picnic, suitable parameters give a signature size, for the security level $L1$ identified by NIST (which is equivalent to the security level of AES128 [NIS]), about

approximately 33 K-Bytes (*resp.* 75 K-Bytes for the level $L3$), with a public key of 32 Bytes (*resp.* 48), and a secret key of 16 Bytes (*resp.* 24).

Additionally, we can cite the lattice-based signature scheme presented in Fiat-Shamir with aborts [Lyu09]. It also includes the Fiat-Shamir method to transform the IDS into a signature scheme. The resulting schemes gives signatures of small sizes, while the public/secret keys are large. Moreover, Dilithium [DKL$^+$18] is a scheme based on the Fiat-Shamir with aborts approach. This lattice-based signature scheme provides signatures of approximately 2.6 K-Bytes for the security level $L1$ [NIS], coming with a large public key of 1472 Bytes.

The results give post-quantum schemes in the strong sense, and this opens the way to consider other algebraic problems like PKP. However, in order to compare with our scheme, we keep the digital signatures converted from Zero-knowledge Identification schemes.

Main results. The main contribution of this paper is to present a new post-quantum signature scheme. After the complexity analysis of the PKP problem, we are particularly interested in the design of a signature scheme. Similarly to the approaches cited above, by applying the Fiat-Shamir transform, we study the design of post-quantum signature constructed from a 5-pass authentication scheme based on the PKP problem.

Our objective is to define the most optimal parameters for hard instances of this problem, with respect to the security levels identified by NIST [NIS].
The PKP-DSS scheme based on PKP compared well with the schemes listed in Section 1. We obtained positive results: smaller signature size for the same security levels. Then, this makes the signature scheme based on PKP a competitive cryptosystem.

Structure. After a short introduction to the so-called PKP problem, we give a small representation for comparable works constructed from ZK-IDS. Also, we present in the first Section 1 the main results obtained.

Section 2 contains the definition of the PKP problem and their various modeling. More precisely, we detail the well-known attacks for the so-called PKP problem over a finite field. As well, we discuss the complexity analysis of solving this problem.

Section 3 presents the ZK-IDS based on PKP. Moreover, in order to give a new signature scheme based on this problem, this Section focuses on the essential properties of a Zero-knowledge Identification Scheme (ZK-IDS). At the end of this Section, an authentication scheme based on PKP is given.

Section 4 is devoted to the main contribution of this work. The famous Fiat-Shamir (FS) method is utilized to convert this IDS given in Section 3 to a Digital Signature Scheme DSS, in purpose to introduce a competitive post-quantum scheme. At the end of this Section, our numerical results are pointed out to give various parameters used to construct the PKP-DSS.

2 The Permuted Kernel Problem

In order to introduce the signature scheme, we first present the PKP problem [Sha89]. We also present the best techniques for solving it. In [Geo92], J. GEORGIADES presents

symmetric polynomials equations which will be utilized by all the other attacks. The authors of [BCCG92] investigate also the security of PKP, where a time-memory trade off was introduced. Moreover, J. PATARIN and P. CHAUVAUD improve algorithms for the Permuted Kernel Problem[PC93]. After all, in [JJ01], the most efficient approach was proposed.

2.1 Introduction to PKP

PKP [Sha89, GJ79] is the problem on which the security of PKP-DSS is based. PKP is a linear algebra problem which asks to find a kernel vector of given matrix under a vector-entries constraint. It's a generalization of the Partition problem [GJ79, pg.224]. More precisely, it is defined as follows:

Input. A finite field $\mathbb{F}_p$, a matrix $A \in \mathcal{M}_{m\times n}(\mathbb{F}_p)$ and a n-vector $V \in {\mathbb{F}_p}^n$.
Question. Find a permutation π over $(1,\ldots,n)$ *such that* $A \times V_\pi = 0$, *where* $V_\pi = (V_{\pi(j)}), j = 1,\ldots,n$.

A reduction of the 3-Partition problem proves PKP to be NP-Complete [GJ79] in the good reasoning (*i.e.*its hardness grows exponentially with p). A fundamental design assumption of PKP-DSS is that solving random instances of PKP are hard to solve in practice (Section 2.2). In fact, the solidity of PKP comes from, on the one hand, the big number of permutations, on the other hand, from the small number of possible permutations which may suit the kernel equations. More precisely, PKP is hard because it obligates the choice of a vector, with already fixed set of entries, from the kernel of the matrix A.
Note that, to reach higher security levels, it's more desirable that the n-vector V has distinct coordinates. In the next sections, we give the best well known attacks on the PKP problem.

2.2 Best known attacks

The implementation's efficiency of the first IDS, proposed by A. SHAMIR [Sha89], based on PKP problem has led to several solving tools. In fact, there are various attacks for PKP, which are all exponential. We will describe them briefly.

We assume that the matrix $A \in \mathcal{M}_{m\times n}(\mathbb{F}_p)$ is of rank m, given in a systematic form:

$$A = (a_{ij})_{1\le i\le m, 1\le j\le n} \;=\; \left[A'|I\right],$$

where $A' = (a'_{ij})_{1\le i\le m, 1\le j\le n-m} \in \mathcal{M}_{m\times n-m}(\mathbb{F}_p)$ and I is the identity matrix of size m. By denoting $A_\pi = (a_{i\pi(j)})$, the effect of the permutation π over the columns of A, it's easy to see that $A_\pi V_\pi = AV$.

2.2.1 J. GEORGIADES attack

First of all, let's consider the exhaustive search. This test consists of examining all the possible candidates (permutations of a set on n elements) for the solution in order to determine whether a candidate satisfies the problem's conditions. Its complexity is in $n!$.

The basic idea of J. GEORGIADES attack [Geo92] is to find some new equations in order to reduce the set of suitable permutations. Since the rank of A is equal to m, then $dim\big(ker(A)\big) = n-m$. There exists in the kernel of the considered matrix $n-m$ vectors which are linearly independent. So, we can fix the first $n-m$ coordinates of each vector and the last m coordinates are constants depending on A. Consequently, it is possible to give the kernel of A the following form:

$$Ker(A) = \lambda_1 \begin{pmatrix} u_{1,1} \\ u_{1,2} \\ \vdots \\ u_{1,m} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} u_{2,1} \\ u_{2,2} \\ \vdots \\ u_{2,m} \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \lambda_{n-m} \begin{pmatrix} u_{n-m,1} \\ u_{n-m,2} \\ \vdots \\ u_{n-m,m} \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \tag{1}$$

where $u_{1,1}, \ldots, u_{n-m,m}, \ldots$ belong to $\mathbb{F}_p$, and so does the $\lambda_i s$.
Thus, we can conclude that the kernel is the set of vectors:

$$\Big(f_1, f_2, \ldots, f_m, \lambda_1, \lambda_2, \ldots, \lambda_{n-m}\Big), \tag{2}$$

where $\quad f_j = \sum_{i=1}^{n-m} u_{i,j}\lambda_i, \quad j \in \{1, \ldots, m\}$.
Thus, to find the find the secret permutation π, it suffices to exactly place $n-m$ coordinates of the corresponding vector V_π, and then, the other m coordinates will be deduced from the kernel equations 1 and 2. It is equivalent to pick $(n-m)$ values out of n and correctly place them. This will decrease the number of permutations to be considered from $n!$ to:

$$\frac{n!}{\big(n-(n-m)\big)!} = \frac{n!}{m!}.$$

Moreover, this cost may be diminished if we successfully find relations between the $\lambda_i s$. As a matter of fact, $V = (v_1, \ldots, v_n)$ is known, and its permutation $V_\pi = (x_1, \ldots, x_n) \in Ker(A)$ has the form given in 2. Hence, it's feasible to get the following relations in $\mathbb{F}_p$:

$$\sum_{i=1}^{n} v_i^r \mod \text{p} \quad = \quad \sum_{i=1}^{n} x_i^r \mod \text{p} \quad = \quad \sum_{i=1}^{m} f_i^r + \sum_{i=1}^{n-m} \lambda_i^r \mod \text{p}, \qquad (G_r)$$

where r is a positive integer.

Such equations G_r are very useful and, for small values of r (for example $r = 1, 2$) , are simple to calculate. For $r = 1$ (*resp.* $r = 2$), it is easy to represent some λ_i (*resp.* $\lambda_{j\neq i}$) as a linear combination (*resp.* quadratic equation) of the other $n-m-1$ (*resp.* $n-m-2$) parameters. This will reduce, by taking $r = 2$, the possible permutations to:

$$\frac{n!}{\big(n-(n-m-2)\big)!} = \frac{n!}{(m+2)!}.$$

2.2.2 A time-memory trade-off

Another attack on PKP uses the time-memory trade-off. It was introduced by T. BARITAUD, M. CAMPANA, P. CHAUVAUD and H. GILBERT in [BCCG92]. The proposed scheme reduces the time and space required to solve the PKP problem.
Recall that, solving PKP is equivalent to find a permutation π of a vector V such that $A \times V_\pi = 0$. Thus, using the reduced form of A, we can represent PKP as:

$$\begin{pmatrix} a'_{1,1} & \cdots & a'_{1,n-m} & 1 & & \\ \vdots & & \vdots & & \ddots & \\ a'_{m,1} & \cdots & a'_{m,n-m} & & & 1 \end{pmatrix} \begin{pmatrix} V_{\pi(1)} \\ \vdots \\ V_{\pi(n)} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

Consequently, solving PKP is equivalent to solve a system $\mathcal{S}$ of m equations in n variables given by the entries of the matrix product given above. The algorithm is accomplished by considering k equations of $\mathcal{S}$, where $0 \leq k \leq m$ is a parameter of the algorithm. Due to the block form of A, one can easily see that only $n-m+k$ variables, namely $V_{\pi(1)}, \ldots, V_{\pi(n-m+k)}$, are involved in the sub-system of $\mathcal{S}$ formed by k relations. Another parameter $0 \leq k' \leq n-m+k$ is used to indicate the amount of storage to be computed in the first step of the algorithm.
This method is composed by two essential steps:

Step1: precomputation. Recall that the aim is to solve k relations of $\mathcal{S}$. Hence, for each $k'-uple$ $(V_{\pi(1)}, \ldots, V_{\pi(k')})$, the corresponding values are computed as follows:

$$b_1 = \sum_{j=1}^{k'} a'_{1,j} V_{\pi(j)}$$

$$\vdots$$

$$b_k = \sum_{j=1}^{k'} a'_{k,j} V_{\pi(j)}$$

Then, the $\frac{n!}{(n-k')!}$ possible values of the $k'-uples$ and its corresponding results $(b_1, \cdots, b_k)$ are stored. Note that, for each of the p^k possible value of the vector $(b_1, \cdots, b_k)$ the $k'-uple$ $(V_{\pi(1)}, \ldots, V_{\pi(k')})$ are quickly accessed.

This step costs $\frac{n!}{(n-k')!}$ matrix-vector product. The memory required is about $\frac{n!}{(n-k')!}$ $k'-uples$. Also, for each $(b_1,\cdots,b_k)$ value corresponds *approx.* $p^{-k}\frac{n!}{(n-k')!}$ $k'-uples$.

Step2: exhaustive trial. The exhaustive search is performed over the remaining components $(V_{\pi(k'+1)},\ldots,V_{\pi(n-m+k)})$. There is $\frac{n!}{(m+k'-k)!}$ possible value of such vector. For each tested vector, the corresponding values are computed from the k equations:

$$c_1 = \sum_{j=k'+1}^{n-m+k} a'_{1,j} V_{\pi(j)}$$

$$\vdots$$

$$c_k = \sum_{j=k+1}^{n-m+k} a'_{k,j} V_{\pi(j)}$$

Now, using the precomputation step, a list of probable $(V_{\pi(1)},\ldots,V_{\pi(k')})$ is obtained. Obviously, the k relations can be represented as:

$$b_1 + c_1 = 0$$

$$\vdots$$

$$b_k + c_k = 0.$$

Moreover, the $k'-uple$ $(V_{\pi(1)},\ldots,V_{\pi(k')})$ is certainly one of the possible $k'-uples$ for the $(-c_1,\ldots,-c_k)$ value of $(b_1,\ldots,b_k)$.
For every vector $(V_{\pi(k'+1)},\ldots,V_{\pi(n-m+k)})$ generated, there are in average $p^{-k}\frac{n!}{(n-k')!}$ $(V_{\pi(1)},\ldots,V_{\pi(k')})$ values. For each probable solution $(V_{\pi(1)},\ldots,V_{\pi(n-m+k)})$, the remaining unsolved equations from $(k'+1)$ to (m) give successively only one possible value for the last components $V_{\pi(n-m+k+1)},\ldots,V_{\pi(n)}$.
The required space of this step is negligible. In contrast, the required time is about $\sup(\frac{n!}{(m+k'-k)!}\frac{n!}{(n-k')!}p^{-k},\frac{n!}{(m+k'-k)})$ matrix vector product.
Thus, for every pair (k,k'), the total time complexity of solving PKP, using this time-memory attack is about:

$$\frac{n!}{(n-k')!} + \sup\left(\frac{n!}{(m+k'-k)!}\frac{n!}{(n-k')!}p^{-k},\frac{n!}{(m+k'-k)}\right),$$

and the total space required is about:

$$\frac{n!}{(n-k')!} \text{ k'-vectors.}$$

2.2.3 Improved algorithms for PKP

J. PATARIN and P. CHAUVAUD combine in [PC93] the two ideas presented in the previous attacks (see Sections 2.2.1, 2.2.2). The result was a reduction in the time required to attack PKP. They also present some new ideas in order to reduce this time the memory needed.
Thus, this leads to a new algorithm which is quicker and more efficient than the attacks given above [Geo92, BCCG92]. The details and the numerical results are given in the main article [PC93].

2.2.4 A new Approach of A. JOUX and E. JAULMES

In this section, we present the most efficient attack to solve PKP. In [JJ01], A. JOUX and E. JAULMES introduce a new time-memory trade-off algorithm which is an application of the algorithm described in [JL01] to the Permuted Kernel Problem. Attacking PKP with this approach is faster than any previously known method. Moreover, this technique includes the so-called 4SET problem (see [JL99, JJ01] for more details) which is defined as follows:

Input. An n-vector $P = (p_1, \ldots, p_n)$ where the p_is are primes, four sets S_i of n vectors such that $|S_i| = N_i$ for $i = 1 \ldots 4$, and n sets $D_1, \ldots, D_n$.
Question. Find $v^{(1)} \in S_1, \ldots, v^{(4)} \in S_4$, $d_1 \in D_1, \ldots, d_n \in D_n$ such that:

$$\forall i \in [1, \ldots, n], \quad v_i^{(1)} + v_i^{(2)} + v_i^{(3)} + v_i^{(4)} \equiv d_i \pmod{p_i}$$

The solving strategy of 4SET is composed of two phases: a precomputation step and a main loop consisting two enumeration steps (detailed in [JJ01]). The authors of [JJ01] specify reasonable choice of parameters for the solving technique of 4SET. Thus, the its time complexity is given by:

$$\mathcal{O}\Big((n-k)\psi N_1 N_2 N_3 N_4\Big),$$

Where $\psi = \prod_{i=1}^{k} \frac{|D_i|}{p_i}$ for suitable choices of $1 \leq k \leq n$.

As shown in [JJ01], we can reduce an instance of PKP to the 4SET problem. According to [Geo92], it is useful to add the G_r equations. The linear equation G_r, for $r = 1$, represents the fact that the sum σ of the coordinates of the vector V is independent of the secret permutation π. By considering this linear equation the kernel vector V_π verify:

$$(A'_0 | I_{m+1}) V_\pi = \begin{pmatrix} \sigma \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

as said in 2.2, $A = \left[A'|I\right]$. Thus, $A'_0 \in \mathcal{M}_{(m+1)\times(n-m-1)}(\mathbb{F}_p)$, and I_{m+1} is the identity matrix of order $(m+1)$.
Now, A'_0 is divided into four roughly equal parts, so:

$$(A'_1A'_2A'_3A'_4|I_{m+1})V_\pi = \begin{pmatrix} \sigma \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{3}$$

where A'_i is a $(m+1)\times(n_i)$ matrix and $n_1+n_2+n_3+n_4 = n-m-1$.
Recall that V is known and V_π is its permutation vector. In order to apply the 4SET problem, we need to construct the sets S_i. Since A'_i is an $(m+1)\times(n_i)$ matrix, S_i is the set of $m+1$-vectors: the product resulting of A'_i by all the possible n_i combinations of the coordinates of V. So, the size of S_i is equal to $\frac{n!}{(n-n_i)!}$.
In this case, all the primes p_is are equal to the prime number p given by the PKP instance. Now, to determine the $m+1$ sets D_i, lets decompose, similarly to the matrix A'_0, the vector $V_\pi = (V_1^{(\pi)}V_2^{(\pi)}V_3^{(\pi)}V_4^{(\pi)}V_5^{(\pi)})$ such that for $i \in [1,\ldots,4]$, $V_i^{(\pi)}$ is an n_i-vector where, as we quoted before, $n_1+n_2+n_3+n_4 = n-m-1$. So, $V_5^{(\pi)}$ is the vector formed by the last $(m+1)$ coordinates of V_π. Hence, we can reformulate the matrix-vector product 3 as follow:

$$A'_1V_1^{(\pi)} + A'_2V_2^{(\pi)} + A'_3V_3^{(\pi)} + A'_4V_4^{(\pi)} = \begin{pmatrix} \sigma \\ 0 \\ \vdots \\ 0 \end{pmatrix} - V_5^{(\pi)} = \begin{pmatrix} \sigma - v_{n-m}^{(\pi)} \\ -v_{n-m+1}^{(\pi)} \\ \vdots \\ -v_n^{(\pi)} \end{pmatrix} = \mathcal{D}$$

It's obvious that the value of $V_5^{(\pi)}$ depends on the V_i^{π}s , for $i \in [1,\ldots,4]$, so does the n-vector $\mathcal{D}$. The first component of $\mathcal{D}$ depends on $v_{n-m}^{(\pi)}$ which has n possible values. Thus, D_1 is the set of these n possible values. Since V has no double, the set D_2 is formed by $n-1$ elements, and so on. In this way, the sets $D_1,\ldots,D_{m+1}$ are built such that each one has in average:

$$\frac{n+(n-1)+,\ \ldots\ ,+(n-m)}{m+1} \quad elements.$$

Note that we are dealing with bit operations. So, in order to define the solving time complexity, we must pack 32 or even $64 = 2^6$ bit operations in one word operation. It's equivalent to divide the complexity by 2^6. In summary, we have the following time complexity(see [JJ01] for more details):

$$\mathcal{O}\Big((m+1-k)\times\psi\times\frac{n!^2}{(n-n_1-n_2)!(n-n_3-n_4)!}\times 2^{-6}\Big),$$

Where $k = \log_{\frac{|D_i|}{p}}(\psi)$.

It appears in [JJ01] that this new approach is the most efficient to solve PKP.

3 Identification scheme (IDS) based on PKP

In this section, we present the 5-pass Zero-Knowledge Identification Scheme (ZK-IDS) based on the computational hardness of PKP [Sha89, LP11], noted here PKP-IDS.

We first quote and refer to some of the general definitions given in [CHR$^+$18] : Identification scheme, Completeness, Soundness (with soundness error), Honest-verifier zero-knowledge, and also in [HNO$^+$09, Dam99] : statistically hiding commitment, computationally binding commitment. We then apply and adapt these definitions to the Identification scheme base on PKP and give and prove its own properties of performance and security. This approach will be more convenient for presenting the signature scheme in the next section.

3.1 Preliminaries

In what follows and as in [CHR$^+$18], we assume the existence of a non-interactive commitment scheme *Com* which verifies the two properties : statistically hiding and computationally binding (see [HNO$^+$09, Dam99] for details). The commitments are computed using the function *Com*. Note that, it is possible to let *Com* be $\mathcal{H}$ a one way hash and collision intractable function, behaving like a random oracle.

3.2 PKP 5-pass IDS

In this section, we present (slightly modified version of) PKP-IDS. It can be described as three probabilistic polynomial time algorithms $IDS = (\text{KEYGEN}, \mathcal{P}, \mathcal{V})$ for which we give below a literal description. The security parameter of the identification scheme is noted λ.

Generation of the public key and secret key in PKP-IDS. The users first agree on a prime number p, and a $m \times n$ matrix A with coefficients in $\mathbb{F}_p$. The public-key in PKP-IDS is given by an instance of PKP with a *preassigned* solution that will be the secret-key. Thus, each user picks a (right) kernel-vector $W \in$ Ker(A), then randomly generates a secret permutation of n elements $\mathsf{sk} = \pi$ and finishes by computing $V = W_{\pi^{-1}}$.
We summarize the public-key/secret-key generation in Algorithm 1. It takes the security parameter λ as input.

One 5-pass round of identification : Prover $\mathcal{P}$ and Verifier $\mathcal{V}$. Prover and Verifier are interactive algorithms that realize the identification protocol in 5 passes. The 5 passes consist in one commitment and two responses transmitted from the prover to the verifier and two challenges transmitted from the verifier to the prover. Random

Algorithm 1 pk/sk generation in PKP-IDS

procedure PKP-IDS.KEYGEN(1^λ)
  pk.seed $\leftarrow$ Randomly sample λ bits
  Randomly sample a matrix $A \in \mathcal{M}_{m\times n}(\mathbb{F}_p)$ using a pseudo-random generator with pk.seed
  Randomly pick a n-vector $W \in Ker(A)$
  sk.seed $\leftarrow$ Randomly sample λ bits
  Generate a random permutation $\pi \in S_n$ using a pseudo-random generator with sk.seed
  sk $\leftarrow \pi$
  Compute $V = W_{\pi^{-1}}$
  pk $\leftarrow (p,$ pk.seed$, V)$
Return (pk, sk)
end procedure

choices of prover and verifier are made using the uniform distribution. The protocol of identification is summarized in Algorithm 2.

From Shamir in [Sha89] we have the following results.

Theorem 3.1. *PKP-IDS is complete. PKP-IDS is statistically zero knowledge when the commitment scheme Com is computationally binding. PKP-IDS is sound with soundness error* $\frac{p+1}{2p}$ *when the commitment scheme Com is computationally binding.*

Definition 3.2 (N rounds of PKP-IDS)**.** *Let PKP-IDS*$= (\text{KEYGEN}, \mathcal{P}, \mathcal{V})$ *then* $PKP-IDS^N = (\text{KEYGEN}, \mathcal{P}^N, \mathcal{V}^N)$ *is the parallel composition of N rounds of PKP−IDS.*

Key sizes. The secret key is the permutation π obtained using a pseudo-random generator that takes as input a seed of λ bits. The size of the public vector V is $n\log_2(p)$ bits. The bit size of the public key (p,A,V) is:

$$\log_2(p) + \lambda + n\log_2(p)\ bits.$$

Performance of the scheme. We can now provide the communication complexity of the IDS, where its fraud's probability is $\frac{p+1}{2p}$. Consider that the commitment function *Com* used in the protocol, returns values of 2λ bits. The transfer of the n-vector $Z \in {\mathbb{F}_p}^n$ requires $n\log_2 p$ Thus, the fourth passes demand $4\lambda + (n+1)\log_2 p + 1$ bits.

Note also that, compared to the original scheme of Shamir in [Sha89], we have reduced the complexity in communication by revealing only the seed used to generate the random elements. More precisely, instead of revealing the random permutation σ, the prover $\mathcal{P}$ only sends its seed sigma.seed.

So, the last pass needs, according to Ch_1, λ bits to reveal the permutation σ if $\text{Ch}_1 = 0$; and $\log_2(n!)$ bits to reveal the permutation $\pi\sigma$, if $\text{Ch}_1 = 1$.
In total, the weighted average bit complexity of the scheme repeated N rounds is given

Algorithm 2 One round of the 5-pass identification scheme

procedures $\mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk})$
  //Prover setup
  $\mathcal{P}$ sets $R \leftarrow$ Random vector in ${\mathbb{F}_p}^n$
  $\mathcal{P}$ sets $\sigma.\mathsf{seed} \leftarrow$ Random seed of λ bits
  $\mathcal{P}$ sets $\sigma \leftarrow$ Random permutation in S_n using a pseudo-random generator with $\sigma.\mathsf{seed}$
  //Commitment step by the Prover
  $\mathcal{P}$ sets $\mathsf{C}_0 \leftarrow Com\big(\sigma, AR\big)$
  $\mathcal{P}$ sets $\mathsf{C}_1 \leftarrow Com\big(\pi\sigma, R_\sigma\big)$
  $\mathcal{P}$ sends $(\mathsf{C}_0, \mathsf{C}_1)$ to $\mathcal{V}$
  //First challenge by the verifier
  $\mathcal{V}$ sets $\mathsf{Ch}_0 \leftarrow$ c random in $\mathbb{F}_p$
  $\mathcal{V}$ sends Ch_0 to $\mathcal{P}$
  $\mathcal{P}$ sets $Z \leftarrow R_\sigma + cV_{\pi\sigma}$ and sends Z to $\mathcal{V}$
  $\mathcal{V}$ sets $\mathsf{Ch}_1 \leftarrow$ b random bit
  $\mathcal{V}$ sends Ch_1 to $\mathcal{P}$
  if $\mathsf{Ch}_1 = 0$ **then**
    $\mathcal{P}$ reveals $\sigma.\mathsf{seed}$ to $\mathcal{V}$
    $\mathcal{V}$ accepts if $Com\big(\sigma, A_\sigma Z\big) = \mathsf{C}_0$
  else
    $\mathcal{P}$ reveals $\pi\sigma$ to $\mathcal{V}$
    $\mathcal{V}$ accepts if $Com\big(\pi\sigma, Z - cV_{\pi\sigma}\big) = \mathsf{C}_1$
  end if
end procedure

by:

$$\left(4\lambda + (n+1)\log_2 p + 1 + \frac{1}{2}(\lambda + \log_2(n!))\right) \times N.$$

4 Digital signature scheme (DSS) based on PKP

We present here the main contribution of this work which is to construct a DSS *i.e.* a digital signature scheme, based on the PKP problem, from the IDS defined in Section 3. This construction uses the well-known Fiat Shamir transformation [FS86].

So next, we introduce the basic definitions needed. Then, similarly to the MQ-based signatures and Picnic, we define our scheme, and we finish with a comparison with other cryptosystems.

4.1 Introduction

The classical method of Fiat-Shamir (FS) transforms an interactive proof of knowledge (identification scheme) into a non interactive one (signature scheme). This work is a direct application of this method to get PKP-DSS from PKP-IDS.

Fiat-Shamir transform for PKP-IDS. We recall that PKP-IDS the previously defined identification scheme achieves soundness with soundness error $\kappa = \frac{1+p}{2p}$. We select N the number of parallel rounds of PKP-IDS such that κ^N is negligible in λ. We select two cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{F}_p^N$ and $H_2 : \{0,1\}^* \to \{0,1\}^N$. By applying Construction 4.7 in [CHR+18], we get PKP-DSS$= (\text{KEYGEN}, \text{SIGN}, \text{VERIFY})$. See Algorithms 3 and 4.

A valid signature of a message m by PKP-DSS is then a tuple $(m, \sigma_0, \sigma_1, \sigma_2)$, where $\sigma_0, \sigma_1, \sigma_2$ hold the (vector of parallel) commitments and responses of the non interactive prover. The implicit values $h_1 = H_1(m, \sigma_0)$ and $h2 = H_2(m, \sigma_0, h_1, \sigma_1)$ represent the (vector of parallel) challenges of the non interactive verifier.

We get the similar result as Th. 5.1 in [CHR+18].

Theorem 4.1. *PKP-DSS is Existential-Unforgeable under Chosen Adaptive Message Attacks (EU-CMA) in the random oracle model, if*

- *the search version of the Permuted Kernel problem is intractable,*
- *the hash functions are modeled as random oracles,*
- *the commitment functions are computationally binding, computationally hiding, and the probability that their output takes a given value is negligible in the security parameter,*
- *the pseudo-random generators are modeled as random oracle, and*
- *the pseudo-random generators have outputs computationally indistinguishable from random.*

The proof is exactly the same as in [CHR+18].

Algorithm 3 Signing process in PKP-DSS

procedure PKP-DSS.SIGN(m, sk)
  $\mathsf{R} \leftarrow \mathcal{H}_0\big(\mathsf{sk} \,||\, m\big)$, R is a message-dependent random value
  $\mathsf{D} \leftarrow \mathcal{H}_0\big(\mathsf{pk} \,||\, \mathsf{R} \,||\, m\big)$, D is the randomized message digest
  $\mathsf{R}^{(1)}, \ldots, \mathsf{R}^{(\mathsf{N})} \leftarrow \mathsf{RG}_0\big(\mathsf{R.seed},\ \mathsf{D}\big)$
  $\sigma^{(1)}, \ldots, \sigma^{(\mathsf{N})} \leftarrow \mathsf{RG}_1\big(\sigma\mathsf{.seed},\ \mathsf{D}\big)$
  for j *from* 1 *to* N **do**
    $\mathsf{C}_0^{(j)} = Com\big(\sigma^{(j)}, AR^{(j)}\big)$,
    $\mathsf{C}_1^{(j)} = Com\big(\pi\sigma^{(j)}, R^{(j)}_{\sigma^{(j)}}\big)$.
    $\mathsf{COM}^{(\mathsf{j})} := \big(\mathsf{C}_0^{(j)},\ \mathsf{C}_1^{(j)}\big)$
  end for
  $\mathcal{S}_0 \leftarrow \mathcal{H}_0\big(\mathsf{COM}^{(1)} || \ \ldots\ || \mathsf{COM}^{(N)}\big)$.
  $\mathsf{Ch}_0 \leftarrow \mathcal{H}_1\big(\mathsf{D},\ \mathcal{S}_0\big)$
  Parse Ch_0 *as* $\mathsf{Ch}_0 := (c^{(1)}, \ldots, c^{(N)}),\ c^{(j)} \in \mathbb{F}_p$
  for j *from* 1 *to* N **do**
    $Z^{(j)} \leftarrow R^{(j)}_{\sigma^{(j)}} + c^{(j)} V_{\pi\sigma^{(j)}}$,
    $\mathsf{resp}_0^{(j)} := Z^{(j)}$.
  end for
  $\mathcal{S}_1 \leftarrow \big(\mathsf{resp}_0^{(1)} || \ \ldots\ || \mathsf{resp}_0^{(N)}\big) = \big(Z^{(1)} || \ \ldots\ ||, Z^{(N)}\big)$.
  $\mathsf{Ch}_1 \leftarrow \mathcal{H}_2\big(\mathsf{D},\ \mathcal{S}_0,\ \mathsf{Ch}_0,\ \mathcal{S}_1\big)$
  Parse Ch_1 *as* $\mathsf{Ch}_1 := (b^{(1)}, \ldots, b^{(N)}),\ b^{(j)} \in \{0,1\}$
  for j *in* $(1 \ldots N)$ **do**
    if $b^{(j)} = 0$ **then**
      $\mathsf{resp}_1^{(j)} \leftarrow \sigma^{(j)}$.
    else
      $\mathsf{resp}_1^{(j)} \leftarrow \pi\sigma^{(j)}$.
    end if
  end for
  $\mathcal{S}_2 \leftarrow \big(\mathsf{resp}_1^{(1)} || \ \ldots\ || \mathsf{resp}_1^{(N)} || \mathsf{C}^{(1)}_{1-b^{(1)}} || \ \ldots\ || \mathsf{C}^{(N)}_{1-b^{(N)}}\big)$.
Return $\big(\mathsf{R},\ \mathcal{S}_0,\ \mathcal{S}_1,\ \mathcal{S}_2\big)$.
end procedure

Algorithm 4 Verification process in PKP-DSS

procedure PKP-DSS.$\textsc{Verify}\big(m, \mathsf{pk}, \mathcal{S} = (\mathsf{R}, \mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2)\big)$
  $\mathsf{D} \leftarrow \mathcal{H}_0\big(\mathsf{pk} \,||\, \mathsf{R} \,||\, m\big)$, D is the randomized message digest
  $\mathsf{Ch}_0 \leftarrow \mathcal{H}_1\big(\quad(\mathsf{D}, \quad \mathcal{S}_0)\quad\big)$
  Parse Ch_0 *as* $\mathsf{Ch}_0 := (c^{(1)}, \ldots, c^{(N)})$, $c^{(j)} \in \mathbb{F}_p$
  $\mathsf{Ch}_1 \leftarrow \mathcal{H}_2\big(\mathsf{D},\ \mathcal{S}_0,\ \mathsf{Ch}_0,\ \mathcal{S}_1\big)$
  Parse Ch_1 *as* $\mathsf{Ch}_1 := (b^{(1)}, \ldots, b^{(N)})$, $b^{(j)} \in \{0,1\}$
  Parse $\mathcal{S}_1$ *as* $\mathcal{S}_1 := \big(\mathsf{resp}_0^{(1)} || \ \ldots \ || \mathsf{resp}_0^{(N)}\big)$
  Parse $\mathcal{S}_2$ *as* $\mathcal{S}_2 := \big(\mathsf{resp}_1^{(1)} || \ \ldots \ || \mathsf{resp}_1^{(N)} || \mathsf{C}^{(1)}_{1-b^{(1)}} || \ \ldots \ || \mathsf{C}^{(N)}_{1-b^{(N)}}\big)$.
  for j *in* $(1 \ldots N)$ **do**
    $Z^{(j)} := \mathsf{resp}_0^{(j)}$,
    if $b^{(j)} = 0$ **then**
      $\sigma^{(j)} := \mathsf{resp}_1^{(j)}$,
      $\mathsf{C}_0^{(j)} := Com\big(\sigma^{(j)}, A_{\sigma^{(j)}} Z^{(j)}\big)$
    else
      $\pi\sigma^{(j)=\mathsf{resp}_1^{(j)}}$
      $\mathsf{C}_1^{(j)} = Com\big(\pi\sigma^{(j)}, Z^{(j)} - c^{(j)} V_{\pi\sigma^{(j)}}\big)$
    end if
    $\mathsf{COM}^{(\mathrm{j})} := \big(\mathsf{C}_0^{(j)},\ \mathsf{C}_1^{(j)}\big)$
  end for
  $\mathcal{S}_0' \leftarrow \mathcal{H}_0\big(\mathsf{COM}^{(1)} || \ \ldots \ || \mathsf{COM}^{(N)}\big)$.
  return $\mathcal{S}_0' \ = \ \mathcal{S}_0$.
end procedure

4.2 Performance of the scheme

Our main goal is to find the best parameters which can ensure the minimal size of a signature. We show, in the next sections, that the PKP-based signature scheme provides a signature's size less than the other signature schemes, precisely MQDSS [CHR+18] and Picnic [CDG+17].

Signature size: We said that our signing scheme is constructed from the iterations of the IDS (given in 2). Now, to have the total cost, it is important to define the number of rounds N needed to achieve **EU-CMA** for λ bits of security. By considering the scheme where the fraud's probability is $P_f = \frac{p+1}{2p}$. We require that

$$P_f^N \leq 2^{-\lambda},$$

as an attacker could perform a preimage search to control the challenges. Hence, we get that $N \geq \lambda / \log_2(\frac{p+1}{2p})$.

We begin to present how to compute the complexity in bits. Recall that the signature is composed of R the message-dependent random value, $\mathcal{S}_0$, $\mathcal{S}_1$ and $\mathcal{S}_2$, where $\mathcal{S}_0$ is the hashed value of the commitments of all rounds, $\mathcal{S}_1$ is formed by the first responses, and $\mathcal{S}_2$ is the concatenation of the some commitments and the second responses to the challenges.
For $\mathcal{S}_0$ which is a hashed value, it costs 2λ bits. $\mathcal{S}_1$ depends on the size of Z, so it is in $N \times n \log_2 p$. For $\mathcal{S}_2$, we present next each case:

- **b=0:** The signer reveals one seed $\mathsf{sigma.seed}$ (similarly to 2) as a response. It costs the seed size which is presented by λ bits. In addition to the size of the commitment C_1, we have in average:

$$A \;=\; \frac{1}{2}\big(\mathit{Size}(\mathsf{C}_1) + \mathit{Size}(\mathsf{resp}_1)\big) \;=\; \frac{3}{2}\lambda.$$

- **b=1:** The signer reveals the permutation $\pi\sigma^{(j)}$ as a response resp_1 to the challenge $b^{(j)}$. By adding also the commitment C_0 of size 2λ bits, we have in total:

$$B \;=\; \frac{1}{2}\big(2\lambda + \log_2(n!)\big).$$

We have thus the following signature size:

$$\overbrace{2\lambda}^{\textit{size of } \mathsf{R}} \;+\; \overbrace{2\lambda}^{\textit{size of } \mathcal{S}_0} \;+\underbrace{N\big(n\log_2(p) + A + B\big)}_{\textit{size of } \mathcal{S}_1 \textit{ and } \mathcal{S}_2}.$$

4.2.1 How parameters affect performance

As we said previously, the DSS is mainly affected by the following set of parameters: (p,n,m). We now explicitly detail the choice of parameters. Recall that firstly the IDS [Sha89] was designed to suit small devices. Thus, A. SHAMIR proposed $p = 256$. Nowadays, with the $64-$bit computer architecture, the computations modulo a prime number of 32 or 64 bits are feasible. Thus, we consider that p is of $8,\ 16,\ 32$, or 64 bits.

A solution of a random instance of PKP is to find a kernel n-vector (V_π) with distinct coordinates in $\mathbb{F}_p$. Hence, the probability to find such vector shouldn't be too small. Also in [Sha89], A. SHAMIR estimated n to be between 32 and 64. Later on, several attacks [BCCG92, PC93] shows that the choice $n = 32$ is not recommended for strong security requirements. So, to find an n-vector with no double in $\mathbb{F}_p$, and by considering the Birthday Paradox, we keep the choice of n around 64, in addition to $n \approx \mathcal{O}\left(\sqrt{p}\right)$.

On the other hand, the probability of an arbitrary vector to be in the kernel of the matrix $A \in \mathcal{M}_{m\times n}$ whose rank is equal to m, is p^{-m}. Moreover, if the n-vector V has no double, the cardinal of its orbit under the possible permutations π is $n!$. Thus, in order to get one solution, we have the following constraint: $n! \approx p^m$.

Hence, following these criteria, we have in total:

$$\begin{gathered} p \approx \mathcal{O}\left(n^2\right), \\ n! \approx n^n \approx p^m. \end{gathered}$$

This leads to take $m \approx n\log(n)/log(p) \approx n/2$.

How to choose the security parameter λ. Recall that, the security parameter λ controls the number of iterations $N = \lambda/\log_2(\frac{p+1}{2p})$ performed to achieve a security level needed. It also defines the output of the hash and commitments functions which is in 2λ , in addition to the seeds length.

In general, the hash and commitment functions require collision resistance, preimage resistance, and/or second preimage resistance. Thus, in this article, to reach for example a security of 128 bits, we initiate λ to be exactly of 128 bits. As well for the others security levels (192 and 256).

However, as shown in [GS94], it is always possible to reduce this choice of 256-bit hash values while keeping a security level of 128 bits. Yet, to compare PKP-DSS to the other schemes (as MQDSS) we keep this doubling. Note that, the optimization of [GS94] can be applied to PKP-DSS as well to the other schemes (MQDSS, Picnic,...).

In the following table we present several parameters sets for different levels of security. We define these parameters by considering the formulas given in Section 4.2 and the criteria defined above. Furthermore, our parameters raise a secure scheme against all the attacks described in Section 2.2, mainly, against the most efficient attack: the approach of A. JOUX and E. JAULMES [JJ01].

Parameters Set	Security parameter λ	p	n	m	Iterations number N	Best classical attack
PKP-DSS-128	128	379	51	16	129	2^{129} $op.$
PKP-DSS-192	192	521	70	23	193	2^{192} $op.$
PKP-DSS-256	256	661	90	33	257	2^{257} $op.$

Table 1: PKP-DSS Parameters sets

Next, we compare PKP-DSS to MQDSS [CHR+18] and Picnic [CDG+17]. We consider the public/secret (pk/sk) keys size and the signature size, for different security levels.

Security level	Parameters Sets	Secret key size (Bytes)	Public key size (Bytes)	Signature size (KBytes)
128	PKP-DSS-128	16	71.7	14.72
	MQDSS-31-48	16	46	16.15
	Picnic-L1-FS	16	32	33.2
192	PKP-DSS-192	24	104.1	32.46
	MQDSS-31-64	24	64	33.23
	Picnic-L3-FS	24	48	74.9
256	PKP-DSS-256	32	138.6	57.87
	MQDSS-31-88	32	87	60.28
	Picnic-L5-FS	32	64	129.7

Table 2: Comparison of different schemes

One can conclude that the IDS based on PKP constitutes one of the most efficient schemes.

5 Conclusion

We have seen the most well-known attacks of PKP. We presented briefly each one. Also, we discussed the Approach of A. JOUX and E. JAULMES introduced in [JJ01], which is the most efficient one.

The main thing that we have essentially looked at is the construction of a post-quantum secure cryptosystem. In [Sha89], a Zero-knowledge identification scheme (ZK-IDS) was introduced. A well-known method, namely FIAT-SHAMIR technique [FS86], is used to turn an IDS into a digital signature scheme (DSS).

The authors of [CHR+18], presents a DSS, named MQDSS. It was built from an IDS based on the MQ problem (Multivariate quadratic equations solving problem). Thus, they give several sets of parameters which provide post quantum security.

As well, Picnic [CDG+17] is designed to be secure against classical and quantum attacks. It was also constructed from a Zero-knowledge identification scheme to match different security levels.

Hence, similarly to the technique used to build these schemes, we have constructed a DSS based on the PKP problem. We utilized the ZK-authentication scheme presented in [Sha89] to deduce the signature scheme. In order to compare this latter to the other schemes, we have tested the most known techniques to solve PKP.

We finally conclude several sets of parameters given in 4.2.1 which provides 128, 192 and 256 bits of classical security. Mainly, we conclude that the DSS based on PKP gives signatures with a size smaller than the ones given by the MQDSS and Picnic. Consequently, this is what makes from this PKP-DSS a competitive scheme to the other related cryptosystems.

Since we are comparing PKP-DSS to other post-quantum schemes, it is important to define the security of our scheme against quantum attacks. This can be done by investigating quantum algorithms for solving PKP. Moreover, due to the post quantum security of the FIAT-SHAMIR transform which has been studied in [Unr17], we need to study our results carefully to conclude whether our sets of parameters are post-quantum secure.

References

[BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[BCCG92] Thierry Baritaud, Mireille Campana, Pascal Chauvaud, and Henri Gilbert. On the security of the permuted kernel identification scheme. In *Annual International Cryptology Conference*, pages 305–311. Springer, 1992.

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1825–1842. ACM, 2017.

[CHR+18] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. Mqdss specifications, 2018.

[Dam99] Ivan Damgård. *Commitment Schemes and Zero-Knowledge Protocols*, pages 63–86. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer, 1986.

[Geo92] Jean Georgiades. Some remarks on the security of the identification scheme based on permuted kernels. *Journal of Cryptology*, 5(2):133–137, 1992.

[GJ79] Michael R Garey and David S Johnson. Computers and intractability: a guide to np-completeness, 1979.

[HNO+09] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM Journal on Computing*, 39(3):1153–1218, 2009.

[JJ01] Éliane Jaulmes and Antoine Joux. Cryptanalysis of pkp: a new approach. In *International Workshop on Public Key Cryptography*, pages 165–172. Springer, 2001.

[JL99] Antoine Joux and Reynald Lercier. Chinese & match, an alternative to atkin's match and sort method used in the sea algorithm. *Preprint*, 1999.

[JL01] Antoine Joux and Reynald Lercier. "chinese & match", an alternative to atkin's "match and sort" method used in the sea algorithm. *Mathematics of computation*, 70(234):827–836, 2001.

[LP11] Rodolphe Lampe and Jacques Patarin. Analysis of some natural variants of the pkp algorithm. *IACR Cryptology ePrint Archive*, 2011:686, 2011.

[Lyu09] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 598–616. Springer, 2009.

[NIS] Security strength categories. `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf`.

[NPV12] Valérie Nachef, Jacques Patarin, and Emmanuel Volte. Zero-knowledge for multivariate polynomials. In *LATINCRYPT*, pages 194–213. Springer, 2012.

[PC93] Jaques Patarin and Pascal Chauvaud. Improved algorithms for the permuted kernel problem. In *Annual International Cryptology Conference*, pages 391–402. Springer, 1993.

[Sha89] Adi Shamir. An efficient identification scheme based on permuted kernels. In *Conference on the Theory and Application of Cryptology*, pages 606–609. Springer, 1989.

[SSH11] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-key identification schemes based on multivariate quadratic polynomials. In *CRYPTO*, volume 6841, pages 706–723. Springer, 2011.

[Unr17] Dominique Unruh. Post-quantum security of fiat-shamir. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 65–95. Springer, 2017.