PKP-Based Signature Scheme

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

[1] 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) [1]. PKP is an NP-complete [3] 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 [1]. We derive the signature scheme PKP-DSS by using the traditional Fiat-Shamir (FS) transform [4]. 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 thorough analysis of the State-of-the-art attacks of PKP presented in **[2]**, we define several sets of parameters for different security levels. Each parameter set arises a fast scheme coming with small keys and signatures of length comparable to the other signatures derived from Zero-Knowledge identification schemes. In particular, PKP-DSS-128 gives a signature size approximately about 16 KBytes for 128 bits of classical security, while the best known signature schemes built from a ZK-IDS (such as MQDSS [7], Picnic [22],...) give similar signatures ($\approx$ 16 KB for MQDSS, $\approx$ 33 KB for Picnic,...).
Moreover, the reference implementation shows that PKP-DSS-128 is nearly 48% faster than MQDSS which in its turn is faster than Picnic, SPHINCS,...
Since there are no known quantum attacks for solving PKP significantly better than classical attacks, we believe that effects of quantum computer on our scheme will be moderate.

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 [4]. 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 [5]. 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.

In 1989, A. SHAMIR [1] introduced a scheme of a new nature, a ZK-Identification scheme, based on the Permuted Kernel Problem. It is an old-time NP-complete combinatorial problem. PKP requires simple operations which involve basic linear algebra computations. For a little long time, no new attacks on PKP were reported which makes the construction of schemes based on hard instances of this problem more applicable. Here, we study the application in cryptography of the PKP problem over a finite field. We are essentially concerned about this problem because it can be used to build a post-quantum signature scheme based on the hardness of solving random instances of PKP.

Previous work and State-of-the-art. Since quantum computers are known to be incapable to solve *NP*-Complete problems [5], the Zero-knowledge Identification schemes (ZK-IDS), based on such problems, are very interesting nowadays. The Fiat-Shamir transform [4] 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 [8,9].

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 [7] with concrete parameters and detailed implementation. It has opened the doors to consider other Identifaction schemes based on NP-Complete problems.

In [7], 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 [7] 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 [22] 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 [6]), 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 [10]. 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 [11] 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 $L1$ [6], 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 [2], 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 a 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 [6].
The signature scheme PKP-DSS based on PKP compared well (in terms of construction) with the schemes listed in Section 1 . We obtained the following results: a fast scheme concerning both signing and verification process, a small public-key, and a comparable signature size for the same security levels. Then, this makes the signature scheme based on PKP a competitive cryptosystem.

2 The Permuted Kernel Problem

In order to introduce the signature scheme, we first present the PKP problem [1]. We also briefly present the best technique for solving it. In [12], J. GEORGIADES presents symmetric polynomials equations which will be utilized by all the other attacks. The authors of [13] 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[14]. Also, in [16], a new time-memory trade-off was proposed. After all, it appears in [2] that the attack of PATARIN-CHAUVAUD [14] is the most efficient one.

2.1 Introduction to PKP

PKP [1,3] 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 [3, 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 [3] 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 ( ). 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 section, we give the best well known attack on the PKP problem.

2.2 The algorithm of PATARIN-CHAUVAUD

The implementation's efficiency of the first IDS, proposed by A. SHAMIR [1], based on PKP problem has led to several solving tools. In fact, there are various attacks for PKP, which are all exponential. We will not describe them here, instead we refer to [2] for further details .

J. PATARIN and P. CHAUVAUD combine in [14] the two ideas presented in the previous attacks [12,13]. 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 all the given attacks of PKP [12,13,17]. The details and the numerical results are given in the main article [2].

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 [1,23], noted here PKP-IDS.

We first quote and refer to some of the general definitions given in [7] : Identification scheme, Completeness, Soundness (with soundness error), Honest-verifier zero-knowledge, and also in [21,29] : statistically hiding commitment, computationally binding commitment. We then apply and adapt these definitions to the Identification scheme based 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 [7], we assume the existence of a non-interactive commitment scheme *Com* which verifies the two properties : statistically hiding and computationally

binding (see [21,29] 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 For. 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.

Algorithm 1 pk/sk generation in PKP-IDS

1: **procedure** PKP-IDS.KEYGEN(n,m,p)
2: Randomly sample a matrix $A \in \mathcal{M}_{m\times n}(\mathbb{F}_p)$
3: Randomly pick a n-vector $W \in Ker(A)$
4: Generate a random permutation $\pi \in S_n$
5: $\mathsf{sk} \leftarrow \pi$
6: Compute $V = W_{\pi^{-1}}$
7: $\mathsf{pk} \leftarrow (A,V)$
8: **Return** $(\mathsf{pk},\mathsf{sk})$
9: **end procedure**

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

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

From SHAMIR in [1] we have the following results.

Theorem 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* $\kappa = \frac{p+1}{2p}$ *when the commitment scheme Com is computationally binding.*

In such ZK-IDS, it is usually possible to cheat: a cheater is generally able to predict some questions, but not all of them, so there is a possibility to fraud. The systems are constructed in a manner that answering a question reveals no secret (Zero-knowledge), when giving the answers to all the questions verifies the possession of a secret (Soundness). It is obvious that the security of a ZK-IDS relies on the difficulty by a prover $\mathcal{P}$ to prepare in advance to answer the verifier's questions. Thus, it is necessary to repeat the protocol several time in order to reduce the probability of fraud.
Thus, the cheating (fraud) probability for numerous iterations defined as follows:

Definition 1 (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* $PKP-IDS$.

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 [1], 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 $\mathsf{Ch}_1 = 0$; and $\log_2(n!)$ bits to reveal the permutation $\pi\sigma$, if $\mathsf{Ch}_1 = 1$.
In total, the weighted average bit complexity of the scheme repeated N rounds is given 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 [4].

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 $\kappa = \frac{1+p}{2p}$.

Construction. The Signature Scheme is the Fiat-Shamir Transform of N parallel rounds of the 5-pass Identification protocol. All random generations are turned into deterministic generations using Pseudo-random generators and secret seeds. We need the following:

- Three cryptographic hash functions $H_0 : \{0,1\}^* \to \{0,1\}^{2\lambda}$, $H_1 : \{0,1\}^* \to \mathbb{F}_p^N$ and $H_2 : \{0,1\}^* \to \{0,1\}^N$, where N is the number of iterations needed to attend the security level λ needed.
- A commitment function *Com* to compute the commitments, as the authentication scheme, and its outputs is of 2λ bits.
- Pseudo-random generators.

By applying the well-known Fiat Shamir transformation [4], we get PKP-DSS$=$ (KeyGen, Sign, Verify) (4).

Key generation. The *KeyGen* algorithm output a secret key sk, a public key pk defined in terms of seeds in order to minimize their size. In order to have a constant time of generation of the vector V, The matrix A is generated but its last column. The last column of A (noted LC) is computed using the relation $AV_\pi = 0$. This computation involves divisions by the last coordinate of V_π. Therefore we require that all coordinates of V are non zero, which does not decrease security, since these coordinates are public. Our key generation procedure is given in (3).

Signing. The *Sign* algorithm takes as input a message $m \in \{0,1\}^*$ and a secret key sk. Our signing process is given in (4). It is constructed from the iterations of the IDS. Denote by N the number of iterations to achieve the security level λ.

Verification. The verification function takes as input the message m, the signature σ and the public key pk.
The verification process is listed in (5).

Algorithm 3 Key generation in PKP-DSS

procedure PKP-DSS.KEYGEN(1^λ)
  $seed \leftarrow$ Randomly sample λ bits
  $(seed_\pi, seed_{A^*}, seed_V, seed_R, seed_S) \leftarrow PRG0(seed)$
  $\pi \leftarrow PRG1(seed_\pi)$
  $A^* \leftarrow PRG2(seed_{A^*})$
  $V \leftarrow PRG3(seed_V)$
    Compute LC from A^* and V_π
    $\mathsf{sk} \leftarrow seed$
    $\mathsf{pk} \leftarrow (seed_{A^*}, seed_V, LC)$
  Return $(\mathsf{pk}, \mathsf{sk})$
end procedure

A valid signature of a message m by PKP-DSS 5 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 $h_2 = H_2(m, \sigma_0, h_1, \sigma_1)$ represent the (vector of parallel) challenges of the non interactive verifier.

Algorithm 4 Signing process in PKP-DSS

procedure PKP-DSS.SIGN(m, sk)
  $(seed_\pi, seed_{A^*}, seed_V, seed_R, seed_S) \leftarrow PRG0(\mathsf{sk})$
  $\pi \leftarrow PRG1(seed_\pi)$
  $A^* \leftarrow PRG2(seed_{A^*})$
  $V \leftarrow PRG3(seed_V)$
  Compute LC (last column of A) from A^* and V
  $A \leftarrow A^* || LC$
  $\mathsf{R} \leftarrow \mathcal{H}_0\big(\mathsf{sk} \,||\, m\big)$, 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
  $\mathsf{R}^{(1)}, \ldots, \mathsf{R}^{(\mathsf{N})} \leftarrow PRG4\big(seed_R \,||\, D\big)$
  $\mathsf{seed}_{\sigma^{(1)}}, \ldots, \mathsf{seed}_{\sigma^{(\mathsf{N})}} \leftarrow PRG5\big(seed_S \,||\, D\big)$
  for j *from* 1 *to* N **do**
    $\sigma^{(j)} \leftarrow PRG1(seed_{\sigma^{(j)}})$
    $\mathsf{C}_0^{(j)} = Com\big(\sigma^{(j)}, AR^{(j)}\big)$,
    $\mathsf{C}_1^{(j)} = Com\big(\pi\sigma^{(j)}, R^{(j)}_{\sigma^{(j)}}\big)$.
    $\mathsf{COM}^{(\mathsf{j})} := \big(\mathsf{C}_0^{(j)}, \mathsf{C}_1^{(j)}\big)$
  end for
  $\mathcal{S}_0 \leftarrow \mathcal{H}_0\big(\mathsf{COM}^{(1)} || \ \ldots \ || \mathsf{COM}^{(N)}\big)$.
  $\mathsf{Ch}_0 \leftarrow \mathcal{H}_1\big(\mathsf{D},\ \mathcal{S}_0\big)$
  $(c^{(1)}, \ldots, c^{(N)}) \leftarrow PRG6(\mathsf{Ch0}),\ c^{(j)} \in \mathbb{F}_p$
  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**
      $\mathsf{resp}_1^{(j)} \leftarrow seed_{\sigma^{(j)}}$.
    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 5 Verification process in PKP-DSS

procedure PKP-DSS.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
  $\mathsf{Ch}_0 \leftarrow \mathcal{H}_1\big(\quad(\mathsf{D},\quad \mathcal{S}_0)\quad\big)$
  $(c^{(1)}, \ldots, c^{(N)}) \leftarrow PRG6(\mathsf{Ch0})$, $c^{(j)} \in \mathbb{F}_p$
  $\mathsf{Ch}_1 \leftarrow \mathcal{H}_2\big(\mathsf{D},\ \mathcal{S}_0,\ \mathsf{Ch}_0,\ \mathcal{S}_1\big)$
  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**
      $\sigma^{(j)} := PRG1(\mathsf{resp}_1^{(j)})$,
      $\mathsf{C}_0^{(j)} := Com\big(\sigma^{(j)}, A_{\sigma^{(j)}} Z^{(j)}\big)$
    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

We get the similar result as Th. 5.1 in [7].

Theorem 2. *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 [7].

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 similar and even smaller than the other signature schemes, precisely MQDSS [7] and Picnic [22].

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:

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

- **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 weighted average signature size:

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

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 [1] was designed to suit small devices. Thus, A. SHAMIR proposed $p = 251$. 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 [1], A. SHAMIR estimated n to be between 32 and 64. Later on, several attacks [13,14] 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 $n \approx \mathcal{O}(\sqrt{p})$.

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:

$$p \approx \mathcal{O}(n^2),\ n! \approx p^m.$$

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 [28], 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 [28] 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 [2], mainly, against the most efficient attack: the algorithm of PATARIN-CHAUVAUD [14].

Parameters Set	Security parameter λ	p	n	m	Iterations number N	Best classical attack
PKP-DSS-128	128	251	69	41	129	2^{130} $op.$
PKP-DSS-192	192	509	94	54	193	2^{193} $op.$
PKP-DSS-256	256	4093	106	47	257	2^{257} $op.$

Table 1. PKP-DSS Parameters sets

Next, we compare PKP-DSS to MQDSS [7] and Picnic [22]. We consider the public/secret $(\mathsf{pk}/\mathsf{sk})$ 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	73	16.37
	MQDSS-31-48	16	46	16.15
	Picnic-L1-FS	16	32	33.2
192	PKP-DSS-192	24	109	37.06
	MQDSS-31-64	24	64	33.23
	Picnic-L3-FS	24	48	74.9
256	PKP-DSS-256	32	135	68.97
	MQDSS-31-88	32	87	60.28
	Picnic-L5-FS	32	64	129.7

Table 2. Comparison of different schemes

4.3 Detailed performance analysis

In order to determine the implementation's efficiency of our scheme, we evaluate the performance over 1000 executions of keygen, sign, verf on 32 bytes messages. In particular, our analysis is performed on a machine using Win7, Visual Studio 2017 Community, Release build, Intel Core i5-6300U 2.4GHz. keygen, sign, verif are expressed on the number of cycles. The values given below are the medians of 1000 executions.

	PKP-DSS-128	PKP-DSS-192	PKP-DSS-256
keygen	663254	1501931	2722117
sign	17947216	44624112	78550809
verif	14578081	37842135	67173820

Table 3. Performances in cycles

By considering the parameters of 128 (*resp.* 192) bits of security, it appears that the reference implementation of PKP-DSS is approximately (in terms of signing process) 48% (*resp.* 90%) faster than than MQDSS which in its turn is faster than Picnic, SPHINCS,...
The signature verification process is also quicker than the other schemes.
One can conclude that the signature scheme based on PKP constitutes one of the most efficient schemes.

5 Conclusion

The main thing that we have essentially looked at is the construction of a post-quantum secure cryptosystem. In [1], a Zero-knowledge identification scheme (ZK-IDS) was

introduced. A well-known method, namely FIAT-SHAMIR technique [4], is used to turn an IDS into a digital signature scheme (DSS).

The authors of [7], 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 [22] 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 [1] 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 which provides 128, 192 and 256 bits of classical security. Mainly, we conclude that the DSS based on PKP gives an efficient and fast scheme in terms of signing and verification processes. It also has small keys and a signature size comparable to the ones in MQDSS and smaller than the ones given by Picnic. Consequently, this is what makes from this PKP-DSS a competitive scheme to the other related cryptosystems.

References

1. Shamir, A. (1989, August). An efficient identification scheme based on permuted kernels. In Conference on the Theory and Application of Cryptology (pp. 606-609). Springer, New York, NY.
2. Authors of this article. (2019) In preparation.
3. Gary, M., Johnson, D. (1979). Computers and Intractability: A Guide to NP-Completeness. New York: W H.
4. Fiat, A., Shamir, A. (1986, August). How to prove yourself: Practical solutions to identification and signature problems. In Advances in Cryptology—CRYPTO'86 (pp. 186-194). Springer, Berlin, Heidelberg.
5. Bennett, C. H., Bernstein, E., Brassard, G., Vazirani, U. (1997). Strengths and weaknesses of quantum computing. SIAM journal on Computing, 26(5), 1510-1523.
6. NIST categories: Security strength categories. `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf`,
7. Chen, M. S., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P. MQDSS specifications. (2018).
8. Nachef, V., Patarin, J., Volte, E. (2012, October). Zero-knowledge for multivariate polynomials. In International Conference on Cryptology and Information Security in Latin America (pp. 194-213). Springer, Berlin, Heidelberg.
9. Sakumoto, K., Shirai, T., Hiwatari, H. (2011, August). Public-key identification schemes based on multivariate quadratic polynomials. In Annual Cryptology Conference (pp. 706-723). Springer, Berlin, Heidelberg.
10. Lyubashevsky, V. (2009, December). Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 598-616). Springer, Berlin, Heidelberg.

11. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D. (2018). CRYSTALS-Dilithium: a lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018(1), 238-268.
12. Georgiades, J. (1992). Some remarks on the security of the identification scheme based on permuted kernels. Journal of Cryptology, 5(2), 133-137.
13. Baritaud, T., Campana, M., Chauvaud, P., Gilbert, H. On the security of the permuted kernel identification scheme. In Annual International Cryptology Conference (pp. 305-311). (1992, August), Springer, Berlin, Heidelberg.
14. Patarin, J., Chauvaud, P. Improved algorithms for the permuted kernel problem. In Annual International Cryptology Conference (pp. 391-402). (1993, August) Springer, Berlin, Heidelberg.
15. Chauvaud, P., Patarin, J. Improved algorithms for the permuted kernem problem. (1994) CRYPTO93, 773, 391-402.
16. Jaulmes, É., Joux, A. Cryptanalysis of pkp: a new approach. In International Workshop on Public Key Cryptography (pp. 165-172). (2001, February) Springer, Berlin, Heidelberg.
17. Joux, A., Lercier, R. "Chinese and Match", an alternative to Atkin's "Match and Sort" method used in the SEA algorithm. Mathematics of computation, 70(234), 827-836.
18. Unruh, D. Post-quantum security of Fiat-Shamir. In International Conference on the Theory and Application of Cryptology and Information Security. pp. 65-95. (2017, December) Springer, Cham.
19. Joux, A., Lercier, R. Chinese and Match, an alternative to Atkin's Match and Sort method used in the SEA algorithm. (1999). Preprint.
20. Kipnis, A., Patarin, J., Goubin, L. Unbalanced oil and vinegar signature schemes. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 206-222). (1999, May) Springer, Berlin, Heidelberg.
21. Haitner, I., Nguyen, M. H., Ong, S. J., Reingold, O., Vadhan, S. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. SIAM Journal on Computing, 39(3), pp. 1153-1218.
22. Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., ... Zaverucha, G. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1825-1842) (2017, October). ACM.
23. Lampe, R., Patarin, J. Analysis of Some Natural Variants of the PKP Algorithm. IACR Cryptology ePrint Archive, 2011, 686.
24. Goldwasser, S., Micali, S., Rackoff, C. The knowledge complexity of interactive proof systems. SIAM Journal on computing, 18(1), 186-208.
25. Halevi, S., Micali, S. Practical and provably-secure commitment schemes from collision-free hashing. In Annual International Cryptology Conference (pp. 201-215). (1996, August) Springer, Berlin, Heidelberg.
26. Stern, J. A new identification scheme based on syndrome decoding. In Annual International Cryptology Conference (pp. 13-21). (1993, August) Springer, Berlin, Heidelberg.
27. Pointcheval, D. A new identification scheme based on the perceptrons problem. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 319-328). (1995, May). Springer, Berlin, Heidelberg.
28. Girault, M., Stern, J. On the length of cryptographic hash-values used in identification schemes. In Annual International Cryptology Conference (pp. 202-215). (1994, August).Springer, Berlin, Heidelberg.
29. Damgård, I. Commitment schemes and zero-knowledge protocols. In School organized by the European Educational Forum (pp. 63-86). (1998, June). Springer, Berlin, Heidelberg.

30. Pointcheval, D., Stern, J. Security proofs for signature schemes. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 387-398). (1996, May). Springer, Berlin, Heidelberg.
31. Maurer, U. (Ed.). (2003). Advances in Cryptology—EUROCRYPT'96: International Conference on the Theory and Application of Cryptographic Techniques Saragossa, Spain, May 12–16, 1996 Proceedings (Vol. 1070). Springer.
32. Unruh, D. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 755-784). (2015, April). Springer, Berlin, Heidelberg.