# Polynomial-Time Key-Recovery Attack on the NIST Specification of PROV

River Moreira Ferreira, Ludovic Perret

Sorbonne Université, CNRS, LIP6, F-75005, Paris, France

river.moreira-ferreira@lip6.fr, ludovic.perret@lip6.fr

**Abstract.** In this paper, we present an efficient attack against PROV, a recent variant of the popular Unbalanced Oil and Vinegar (UOV) multivariate signature scheme, that has been submitted to the ongoing NIST standardization process for additional post-quantum signature schemes. A notable feature of PROV is its proof of security, namely, existential unforgeability under a chosen-message attack (EUF-CMA), assuming the hardness of solving the system formed by the public-key non-linear equations. We present a polynomial-time key-recovery attack against the first specification of PROV (v1.0). To do so, we remark that a small fraction of the PROV secret-key is leaked during the signature process. Adapting and extending previous works on basic UOV, we show that the entire secret-key can be then recovered from such a small fraction in polynomial-time. This leads to an efficient attack against PROV that we validated in practice. For all the security parameters suggested in by the authors of PROV, our attack recovers the secret-key in at most 8 seconds. We conclude the paper by discussing the apparent mismatch between such a practical attack and the theoretical security claimed by PROV designers. Our attack is not structural but exploits that the current specification of PROV differs from the required security model. A simple countermeasure makes PROV immune against the attack presented here and led the designers to update the specification of PROV (v1.1).

## 1 Introduction

In 2022, the National Institute of Standards and Technology (NIST) selected the first post-quantum cryptographic standards after five years of competition. In particular, three digital signature schemes (DSS) relying either on structured lattices (Dilithium [14] and Falcon [11]) or hash functions (SPHINCS+ [8]) have been selected for standardization. NIST also decided to start a new standardization process for additional post-quantum DSS to increase the diversity of hardness assumptions. From a practical point of view, the new call was especially targeting schemes with "*short signature*" and "*fast verification*" [1].

Such practical features are typical of multivariate schemes. As such, UOV [9] appears today as the most appealing candidate such that round-1 candidates of the new NIST standardization process [1] includes about 8 UOV-based DSS[1]. A promising candidate among these UOV-variants is the PRovable Unbalanced Oil and Vinegar (PROV) that includes a strong security argument with an EUF-CMA security proof under the hardness assumption of solving PROV public-key multivariate equations. Until now, no security weakness has been reported against PROV.

### 1.1 Organization of the Paper and Main Results

We organize the paper as follows. In Section 2, we introduce the necessary notations, mathematical objects, and the security framework used in PROV. In Section 3, we recall the PROV signature scheme as defined in the NIST-specification v1.0 [5]. Section 4 describes our attack: Section 4.1 details a polynomial-time key-recovery attack against PROV specification v1.0 (Theorem 1). To do so, we exploit the fact that a small fraction of the secret-key is leaked during the signature generation.

[1] https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures

Then, we extend to the specific characteristics PROV a result from [12] on UOV demonstrating that the entire secret-key can be then recovered from this small leakage. The attack has a polynomial-time complexity and is also very efficient in practice. In Section 4.3, we present experimental results and show that the secret key can be recovered in a few seconds for all security levels (Table 2). Section 4.4 discusses a simple tweak allowing to avoid the attack and reestablish the validity of the security model used in [5]. We emphasize that the PROV designers has updated their specification (v1.1) with a similar countermeasure.

### Acknowledgement

We thank Pierre Pébereau, who helped us with some technical details of his attack and implementation [12]. Before publishing this work, we informed the authors of PROV that confirmed the flaw and released quickly a new specification (v.1.1). We would like to thank them for the constructive discussions. The second author would like to thank Google which partially supported this work thanks to a gift dedicated to post-quantum research. Finally, we acknowledge the financial support of the French *Ministère des Armées - Agence de l'innovation de défense* on this research.

## 2 Preliminaries

### 2.1 Notations

Let $q$ be a prime or a prime power (for PROV, $q = 2^8$). We denote by bold lowercase (resp. capital) letter any column vector $\mathbf{v} \in \mathbb{F}_q^n$ of size $n$ in $\mathbb{F}_q$ or respectively any matrix $\mathbf{M} \in \mathbb{F}_q^{n\times m}$ of size $n \times m$ in $\mathbb{F}_q$. In particular, let $\mathbf{0}_n$ be the zero column vector of size $n$ in $\mathbb{F}_q$, $\mathbf{0}_{n\times m}$ be the zero matrix of size $n \times m$ in $\mathbb{F}_q$ and $\mathbf{1}_n$ be the $n$-by-$n$ identity matrix in $\mathbb{F}_q$. For a set of vector $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \in (\mathbb{F}_q^n)^m$, we denote by $\mathsf{span}(\mathbf{b})$ the linear span of $\mathbf{b}$. Also, we express the kernel of a matrix $\mathbf{M} \in \mathbb{F}_q^{n\times m}$ or a linear map $f$ respectively by $\mathsf{Ker}(\mathbf{M})$ and $\mathsf{Ker}(f)$.

The function $\mathsf{Upper}$ takes as input a square matrix $\mathbf{A} = \{a_{i,j}\}_{1\leq i,j\leq n}$ and outputs an upper triangular matrix $\mathsf{Upper}(\mathbf{A}) = \{b_{i,j}\}_{1\leq i,j\leq n}$ such that $b_{i,j} = a_{i,j} + a_{j,i}$ if $i < j$, $b_{i,j} = a_{i,j}$ if $i = j$ or $b_{i,j} = 0$ otherwise. We refer by the symbol $||$ either the concatenation of two bit-strings or the horizontal concatenation of two matrices (depending on the context). Let $\emptyset$ be the empty set, i.e. the set with no element. We denote by $\mathsf{poly}(n)$ a positive polynomial in the variable $n$ (used to simplify complexity analysis).

Let $\mathbb{F}_q[x_1, \ldots, x_n]$ be the ring of multivariate polynomials in $n$ variables with coefficients over $\mathbb{F}_q$. In this work, every quadratic polynomial $p \in \mathbb{F}_q[x_1, \ldots, x_n]$ will be homogeneous, i.e. $p\big(\lambda\,(x_1, \ldots, x_n)\big) = \lambda^2\, p(x_1, \ldots, x_n)$, for all $\lambda \in \mathbb{F}_q^*$. The polar form $p^* : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ of a homogeneous quadratic polynomial $p \in \mathbb{F}_q[x_1, \ldots, x_n]$ is a bi-linear and symmetric function defined as $p^*(\mathbf{x}, \mathbf{y}) := p(\mathbf{x}, \mathbf{y}) - p(\mathbf{x}) - p(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$. Any homogeneous quadratic polynomial $p \in \mathbb{F}_q[x_1, \ldots, x_n]$ can be uniquely represented as $p(\mathbf{x}) = \mathbf{x}^\intercal \mathbf{Q}\mathbf{x}$, where $\mathbf{Q} \in \mathbb{F}_q^{n\times n}$ is an upper triangular matrix, and the corresponding polar form as $p^*(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\intercal(\mathbf{Q} + \mathbf{Q}^\intercal)\mathbf{y}$ with $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$. A multivariate quadratic map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is defined by a set of multivariate quadratic polynomials $(p_1, \ldots, p_m) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$.

### 2.2 Security framework of PROV

Here, we recall the definition of a Weak Preimage-Sampleable Function (WPSF) used in the security analysis of PROV [5].

**Definition 1 (WPSF [5]).** *A WPSF $\mathbf{T}$ consists of four probabilistic polynomial-time algorithms:*

- **Gen***: this algorithm takes as input a security parameter $1^\lambda$ and outputs a function $\mathbf{F} : \mathcal{X} \to \mathcal{Y}$ with a trapdoor $\mathbf{I}$;*
- **F***: this algorithm takes as input a value $x \in \mathcal{X}$ and deterministically outputs $\mathbf{F}(x)$;*

- $\mathbf{I} = (\mathbf{I}^1, \mathbf{I}^2)$*: the algorithm* $\mathbf{I}^1$ *takes no input and outputs a value* $z \in \mathcal{Z}$*; the algorithm* $\mathbf{I}^2$ *takes as input* $z \in \mathcal{Z}$, $y \in \mathcal{Y}$*, and outputs* $x \in \mathcal{X}$ *such that* $\mathbf{F}(x) = y$*, or outputs* $\perp$ *if it failed;*
- **SampDom***: this algorithm takes as input a function* $\mathbf{F} : \mathcal{X} \to \mathcal{Y}$ *and outputs a value* $x \in \mathcal{X}$*.*

The Preimage Sampling (PS) security of a WPSF is defined as:

**Definition 2 (PS security [5]).** *Let* $T$ *be a WPSF. The advantage of an adversary* $\mathcal{A}$ *against the PS security of* $\mathbf{T}$ *is defined as:*

$$\mathbf{Adv}_{\mathbf{T}}^{\mathsf{PS}}(\mathcal{A}) = \left|\Pr\left[\mathsf{PS}_0^{\mathcal{A}} = 1\right] - \Pr\left[\mathsf{PS}_1^{\mathcal{A}} = 1\right]\right|$$

*where* $\mathsf{PS}_0$ *and* $\mathsf{PS}_1$ *are the security games defined in Figure 1.*

```
PS_b
$(\mathbf{F}, \mathbf{I}) \leftarrow \mathbf{Gen}(1^\lambda)$
$b^* \leftarrow \mathcal{A}^{\mathsf{Sample}_b}(\mathbf{F})$
Return $b^*$
```

```
Sample_0
$z_i \leftarrow \mathbf{I}^1()$
repeat
    $y_i \xleftarrow{\$} \mathcal{Y}$
    $x_i \leftarrow \mathbf{I}^2(z_i, y_i)$
until $x_i \neq \perp$ Return $x_i$
```

```
Sample_1
$x_i \leftarrow \mathbf{SampDom}(\mathbf{F})$
Return $x_i$
```

Fig. 1: PS security games.

# 3 Description of PROV

PRovable Unbalanced Oil and Vinegar (PROV) is a new signature scheme [5] submitted to the recent NIST standardization process for additional post-quantum signature schemes [1]. As several multivariate schemes submitted to this standardization process, PROV is a variant of the Unbalanced Oil and Vinegar (UOV) multivariate signature scheme [10].

PROV uses the the recent definition of UOV introduced by W. Beullens in [4] in combination with an efficient variant of the so-called salt-UOV [13], a provably secure variant of UOV.

In [4], the traditional UOV trapdoor [10] is rephrased as the vanishing subspace of a multivariate quadratic map.

**Definition 3.** *Let* $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ *be a multivariate quadratic map and* $\boldsymbol{O} \subset \mathbb{F}_q^n$ *be a linear subspace. We shall say that* $\boldsymbol{O}$ *is a vanishing subspace of* $\mathcal{P}$ *if :*

$$\forall \mathbf{o} \in \boldsymbol{O},\ \mathcal{P}(\mathbf{o}) = \mathbf{0}_m.$$

From a high-level point of view, the public-key in PROV is given by the multivariate quadratic map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and the corresponding secret-key is a vanishing subspace $\boldsymbol{O} \subset \mathbb{F}_q^n$ of dimension $m+\delta$ with $\delta > 0$. The main specificity of PROV is related to the parameter $\delta$ that allows a more efficient reduction than salt-UOV [13]. From now on, we set $v = n - m - \delta$.

## 3.1 Key-Generation in PROV

In order to generate a PROV key pair (Definition 3) $(\mathcal{P}, \boldsymbol{O})$ with $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and a vanishing subspace $\boldsymbol{O} \subset \mathbb{F}_q^n$ of dimension $m+\delta$ with $\delta > 0$, they proceed as follows in [5]. They first generate a random basis of $\boldsymbol{O}$ in systematic form, i.e. namely a basis of the form :

$$(\mathbf{O}^\intercal\ \mathbf{1}_{m+\delta}) \in \mathbb{F}_q^{(m+\delta)\times n},\ \text{with } \mathbf{O} \in \mathbb{F}_q^{v\times(m+\delta)}. \quad (1)$$

Then, the components $p_1, \dots, p_m \in \mathbb{F}_q[x_1, \dots, x_n]$ of $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ are constructed as follows :

$$p_i(\mathbf{x}) = \mathbf{x}^\intercal \mathbf{P}_i \mathbf{x}, \quad \mathbf{P}_i = \begin{pmatrix} \mathbf{P}_i^{(1)} & \mathbf{P}_i^{(2)} \\ \mathbf{0}_{(m+\delta)\times v} & \mathbf{P}_i^{(3)} \end{pmatrix} \in \mathbb{F}_q^{n\times n},\ \forall i,\ 1 \le i \le m, \tag{2}$$

with $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^n, \mathbf{P}_i^{(1)} \in \mathbb{F}_q^{v\times v}$ be an upper triangular matrix, $\mathbf{P}_i^{(2)} \in \mathbb{F}_q^{v\times(m+\delta)}$ be a matrix and $\mathbf{P}_i^{(3)} = \mathsf{Upper}\left(-\mathbf{O}^\intercal \mathbf{P}_i^{(1)} \mathbf{O} - \mathbf{O}^\intercal \mathbf{P}_i^{(2)}\right) \in \mathbb{F}_q^{(m+\delta)\times(m+\delta)}$.

As proved in [5], the linear subspace $\boldsymbol{O}$ generated as in (1) is a vanishing subspace of the map defined by the polynomials (2).

## 3.2 Signature Verification and Generation

The PROV signature for a message $\mathsf{msg} \in \{0,1\}^*$ is given by a vector $\mathbf{s} \in \mathbb{F}_q^n$ and a fixed-length bit string $\mathsf{salt} \in \{0,1\}^{\mathrm{len}_{\mathsf{salt}}}$ such that

$$\mathcal{P}(\mathbf{s}) = \mathcal{H}(\mathsf{msg}||\mathsf{salt}),$$

where $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$ is a hash function[2].

The PROV trapdoor is based on the result below, demonstrating that the knowledge of the vanishing subspace allows one to compute a valid signature by solving a linear system.

**Lemma 1.** *Let $\mathbf{O} \in \mathbb{F}_q^{v\times(m+\delta)}$, $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be represented with matrices $(\mathbf{P}_1, \dots, \mathbf{P}_m) \in (\mathbb{F}_q^{n\times n})^m$ as defined in (2), $\bar{\mathbf{v}} = \begin{pmatrix} \mathbf{v} \\ \mathbf{0}_{m+\delta} \end{pmatrix}, \bar{\mathbf{o}} = \begin{pmatrix} \mathbf{O} \\ \mathbf{1}_{m+\delta} \end{pmatrix} \mathbf{o} \in \mathbb{F}_q^n$, with $\mathbf{v} \in \mathbb{F}_q^v$ and $\mathbf{o} \in \mathbb{F}_q^{(m+\delta)}$. For all $\mathbf{h} = (h_1, \dots, h_m) \in \mathbb{F}_q^m$, it holds that :*

$$\mathcal{P}(\bar{\mathbf{v}} + \bar{\mathbf{o}}) = \mathbf{h} \iff \quad \mathbf{v}^\intercal \mathbf{S}_i \mathbf{o} = h_i - \mathbf{v}^\intercal \mathbf{P}_i^{(1)} \mathbf{v},\ \forall i,\ 1 \le i \le m,$$

*with $\mathbf{S}_i = (\mathbf{P}_i^{(1)} + \mathbf{P}_i^{(1)\intercal})\mathbf{O} + \mathbf{P}_i^{(2)} \in \mathbb{F}_q^{v\times(m+\delta)}$.*

In order to generate a signature of $\mathsf{msg} \in \{0,1\}^*$, the signer generates a random pair $(\mathbf{v}, \mathsf{salt}) \in \mathbb{F}_q^v \times \{0,1\}^{\mathrm{len}_{\mathsf{salt}}}$ and solves the corresponding linear system of Lemma 1 with $\mathbf{h} = \mathcal{H}(\mathsf{msg}||\mathsf{salt}) \in \mathbb{F}_q^m$. If the linear system has no solution, then the signer samples a new $\mathsf{salt} \in \{0,1\}^{\mathrm{len}_{\mathsf{salt}}}$ and solves the new system. The process is repeated until a solution exists. Finally, he recovers the signature $\mathbf{s} = \bar{\mathbf{v}} + \bar{\mathbf{o}}$. We detail the PROV signature generation in Algorithm 1.

*Remark 1.* Note that the vector $\bar{\mathbf{o}}$ belongs in the secret vanishing subspace $\boldsymbol{O}$ of the public key.

Given a matrix $\mathbf{A} \in \mathbb{F}_q^{m\times n}$, and vector $\mathbf{b} \in \mathbb{F}_q^m$, the algorithm LinSolve outputs the set of all solutions $\mathbf{x} \in \mathbb{F}_q^n$ of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$.

## 3.3 Security of PROV

An appealing feature of PROV lies in its security proof where existential forgery under chosen message attacks (EUF-CMA) can be reduced to the problem of inverting the public-key polynomials defined as follows:

**Definition 4 (UOV$^-$ problem).** *Let $\boldsymbol{p} = (p_1, \dots, p_m) \in \mathbb{F}_q[x_1, \dots, x_n]^m$ be quadratic polynomials corresponding to a PROV public-key and $\boldsymbol{d} = (d_1, \dots, d_m) \in \mathbb{F}_q^m$. The UOV$^-$ problem asks to find a solution to the non-linear system of equations :*

$$p_1 - d_1 = 0, \dots, p_m - d_m = 0.$$

As discussed in [5], the best approaches known for solving the UOV$^-$ problem are generic techniques for solving non-linear equations and then exponential in the classical and quantum settings [2,3,7,6].

[2] Precisely, in [5], they generate $\mathbf{h}$ as $\mathcal{H}(4||\mathsf{hpk}||\mathsf{msg}||\mathsf{salt})$ where $\mathsf{hpk}$ is a hash of the public key and a secret seed. We omit this detail to simplify the presentation.

**Algorithm 1:** PROV Signing

```
Data: The secret key $\mathbf{O} \in \mathbb{F}_q^{v \times (m+\delta)}$, the public key $(\mathbf{P}_1, \ldots, \mathbf{P}_m) \in (\mathbb{F}_q^{n \times n})^m$ and a message
      $\mathsf{msg} \in \{0,1\}^*$.
Result: The signature $(\mathbf{s}, \mathsf{salt}) \in \mathbb{F}_q^n \times \{0,1\}^{\mathrm{len}_{\mathsf{salt}}}$ of message $\mathsf{msg}$.
$\mathbf{v} \xleftarrow{\$} \mathbb{F}_q^v$
$\mathcal{S} \leftarrow \emptyset$
while $\mathcal{S} = \emptyset$ do
    $\mathsf{salt} \xleftarrow{\$} \{0,1\}^{\mathrm{len}_{\mathsf{salt}}}$
    $(h_1, \ldots, h_m) \leftarrow \mathcal{H}(\mathsf{msg}||\mathsf{salt})$
    for $i$ from 1 to $m$ do
        $\mathbf{a}_i \leftarrow \mathbf{v}^\intercal((\mathbf{P}_i^{(1)} + \mathbf{P}_i^{(1)\intercal})\mathbf{O} + \mathbf{P}_i^{(2)})$
        $b_i = h_i - \mathbf{v}^\intercal \mathbf{P}_i^{(1)} \mathbf{v}$
    end
    $\mathbf{A} := (\mathbf{a}_1^\intercal || \ldots || \mathbf{a}_m^\intercal)^\intercal$
    $\mathbf{b} := (b_1 || \ldots || b_m)^\intercal$
    $\mathcal{S} \leftarrow \mathrm{LinSolve}(\mathbf{A}, \mathbf{b})$
end
$\mathbf{o} \xleftarrow{\$} \mathcal{S}$
$\mathbf{s} \leftarrow \begin{pmatrix} \mathbf{v} \\ \mathbf{0}_{m+\delta} \end{pmatrix} + \begin{pmatrix} \mathbf{O} \\ \mathbf{1}_{m+\delta} \end{pmatrix} \mathbf{o}$
Return $(\mathbf{s}, \mathsf{salt})$
```

# 4 Polynomial-Time Attack against PROV Specification

## 4.1 Overview

Our attack relies on the fact that the (vinegar) vector $\mathbf{v} \in \mathbb{F}_q^v$ is leaked and constant in the PROV specification v1.0 [5]. More precisely, the designers described a probabilistic signature generation, similar to Algorithm 1, only for "*ease of exposition*". In practice, they deterministically generate $\mathbf{v}$ as $\mathcal{H}(3||\mathsf{msg})$ where $\mathcal{H}$ is the hash function SHAKE256. We emphasize that the reference implementation generates the vinegar vector $\mathbf{v} \in \mathbb{F}_q^v$ similarly.

The vinegar vector (and the corresponding signature) leaks information about the secret-key, precisely it reveals one vector in the secret linear subspace $\boldsymbol{O} \subset \mathbb{F}_q^n$. Recently, P. Pébereau [12] has introduced a polynomial-time key-recovery attack on UOV that uses only one vector in the secret linear subspace. In the next part, we adapt this key-recovery attack on PROV.

## 4.2 Description of the Attack

First, we explain why the PROV specification leaks one vector in the secret linear subspace.
Let $(\mathbf{v}, \mathbf{s}) \in \mathbb{F}_q^v \times \mathbb{F}_q^n$ be a pair of a vinegar vector and a signature[3] for the message $\mathsf{msg} \in \{0,1\}^*$ and the PROV key pair $(\mathcal{P}, \boldsymbol{O})$ with $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and $\boldsymbol{O} \subset \mathbb{F}_q^n$. We recall than $\mathbf{s} = \bar{\mathbf{v}} + \bar{\mathbf{o}}$ where $\bar{\mathbf{v}}^\intercal = (\mathbf{v}^\intercal || \mathbf{0}_{m+\delta}^\intercal) \in \mathbb{F}_q^n$ and $\bar{\mathbf{o}} \in \mathbb{F}_q^n$. As discussed above, the pair $(\mathbf{v}, \mathbf{s})$ is public. Therefore, any adversary can compute a vector in the secret linear subspace $\bar{\mathbf{o}} = \mathbf{s} - \bar{\mathbf{v}} \in \boldsymbol{O}$ (see Remark 1).

In the following, we focus on the key-recovery attack assuming the knowledge of one non-zero vector in the linear subspace. Also, we assume that $n \leq 3m$ (this statement holds for concrete parameters proposed in the PROV submission, see Table 1) and that the rank of the matrices $(\mathbf{P}_1 + \mathbf{P}_1^\intercal, \ldots, \mathbf{P}_m + \mathbf{P}_m^\intercal)$ defined as in (2) is $n$. The attack from [12] uses a special property of the polar form of a PROV key pair.

**Lemma 2 ([12]).** *Let $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be a multivariate quadratic map, $\boldsymbol{O} \subset \mathbb{F}_q^n$ be a vanishing subspace of $\mathcal{P}$ and $\mathcal{P}^* : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^m$ be the polar form of $\mathcal{P}$. Then, for all $(\mathbf{x}, \mathbf{y}) \in \boldsymbol{O}^2$, we have $\mathcal{P}^*(\mathbf{x}, \mathbf{y}) = \mathcal{P}^*(\mathbf{y}, \mathbf{x}) = \mathbf{0}_m$.*

[3] The salt is irrelevant for the attack, therefore we ignore it.

Now, we present an adaption of the attack from [12] to the PROV case (see Remark 2).

**Lemma 3.** *Let $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be a* PROV *public-key, $\boldsymbol{O} \subset \mathbb{F}_q^n$ be the vanishing subspace of $\mathcal{P}$ where $\mathcal{P}$ are represented with matrices $(\mathbf{P}_1, \ldots, \mathbf{P}_m) \in (\mathbb{F}_q^{n\times n})^m$ defined as in (2). Let $\mathbf{o} \in \boldsymbol{O} \setminus \{\mathbf{0}\}$ and $J_{\mathbf{o}}(\mathbf{z}) = (\mathbf{o}^\intercal(\mathbf{P}_1 + \mathbf{P}_1^\intercal)\mathbf{z}, \ldots, \mathbf{o}^\intercal(\mathbf{P}_m + \mathbf{P}_m^\intercal)\mathbf{z})$ with $\boldsymbol{z} = (z_1, \ldots, z_n)$ a vector of variables. Then, $\mathsf{Ker}(J_{\mathbf{o}})$ is an $(n-m)$-dimensional subspace such that*

$$\boldsymbol{O} \subset \mathsf{Ker}(J_{\mathbf{o}}).$$

*Proof.* Let $\mathcal{P}^* : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^m$ be the polar form of $\mathcal{P}$ with components $p_1^*, \ldots, p_m^*$ where $p_i^*(\mathbf{y}, \mathbf{z}) = \mathbf{y}^\intercal(\mathbf{P}_i + \mathbf{P}_i^\intercal)\mathbf{z}$ for all $1 \leq i \leq m$. By Lemma 2, for all $\mathbf{x} \in \boldsymbol{O}$.

$$p_i^*(\mathbf{o}, \mathbf{x}) = 0, \forall 1 \leq i \leq m.$$

This implies that the kernel of the linear application $p_{i,\mathbf{o}}^*(\mathbf{z}) = \mathbf{o}^\intercal(\mathbf{P}_i + \mathbf{P}_i^\intercal)\mathbf{z}$ contains $\boldsymbol{O}$. By hypothesis, all the matrices $(\mathbf{P}_1 + \mathbf{P}_1^\intercal, \ldots, \mathbf{P}_m + \mathbf{P}_m^\intercal)$ are of rank $n$ and $\mathbf{o} \neq 0_n$, therefore $p_{i,\mathbf{o}}^*(\mathbf{z})$ is non-zero. Since the linear map is non-zero, its kernel is a hyperplane. We have shown that $\boldsymbol{O} \subset \mathsf{Ker}(p_{i,\mathbf{o}}^*)$, for all $1 \leq i \leq m$. Therefore, we obtain :

$$\boldsymbol{O} \subset \bigcap_{1\leq i \leq m} \mathsf{Ker}(p_{i,\mathbf{o}}^*) = \mathsf{Ker}(J_{\mathbf{o}})$$

Also, the hyperplanes are non-parallel, because we have $\boldsymbol{O} \subset \mathsf{Ker}(p_{i,\mathbf{o}}^*)$ for all $1 \leq i \leq m$. Therefore, the intersection of the $m$ hyperplanes has dimension $n - m$. This concludes the proof.

*Remark 2.* In [12], the authors consider finite fields of odd characteristics and exploit the bijective relation $p = 2^{-1}p^*$ between any quadratic homogeneous polynomial $p \in \mathbb{F}_q[x_1, \ldots, x_n]$ and its polar form $p^* : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$. Therefore, any homogeneous quadratic polynomial $p \in \mathbb{F}_q[x_1, \ldots, x_n]$ can be represented with a symmetric matrix $\mathbf{M} \in \mathbb{F}_q^{n\times n}$ such that $p(\mathbf{x}) = \mathbf{x}^\intercal\mathbf{M}\mathbf{x}$. However, a polynomial $p(\mathbf{x}) = \mathbf{x}^\intercal\mathbf{M}\mathbf{x}$ with a symmetric matrix $\mathbf{M}$ is either linear or zero in a finite field of characteristic two (as with PROV). In other words, the bijective relation does not hold for finite fields of characteristic two (as discussed in [12,15]). Therefore, we need to slightly adapt the approach from [12] by considering the polar form of the public-key and not by exploiting a symmetric representation of the public-key as in [12].

The next step is then essentially similar to [12, Theorem 7], namely we restrict the public-key polynomials on $\mathsf{Ker}(J_{\mathbf{o}})$ and obtain new polynomial with fewer variables and the same secret vanishing subspace $\boldsymbol{O}$. The secret-key can be recovered in polynomial-time from these new polynomials.

**Theorem 1.** *Let $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be a* PROV *public-key, $\boldsymbol{O} \subset \mathbb{F}_q^n$ be the vanishing subspace of $\mathcal{P}$ with $\mathsf{dim}(\boldsymbol{O}) = m+\delta$ where $\delta > 0$ and $\mathcal{P}$ be represented by matrices $(\mathbf{P}_1, \ldots, \mathbf{P}_m) \in (\mathbb{F}_q^{n\times n})^m$ defined as in (2). Then, there exists an adversary $\mathcal{A}$ taking as input $\big((\mathbf{P}_1, \ldots, \mathbf{P}_m), \mathbf{o}\big) \in (\mathbb{F}_q^{n\times n})^m \times \boldsymbol{O} \setminus \{\mathbf{0}\}$ that outputs a basis of $\boldsymbol{O}$ in polynomial-time.*

The proof is similar to [12] but provided for the sake of correctness and completeness.

*Proof.* Let $J_{\mathbf{o}}(\mathbf{z}) = (\mathbf{o}^\intercal(\mathbf{P}_1 + \mathbf{P}_1^\intercal)\mathbf{z}, \ldots, \mathbf{o}^\intercal(\mathbf{P}_m + \mathbf{P}_m^\intercal)\mathbf{z})$ with $\boldsymbol{z} = (z_1, \ldots, z_n)$ a vector of variables. By Lemma 3, $\boldsymbol{O} \subset \mathsf{Ker}(J_{\mathbf{o}})$ for which a basis $\mathbf{B} \in \mathbb{F}_q^{n\times(n-m)}$ can be computed in $O(n^\omega)$, with $2 \leq \omega \leq 3$ the matrix multiplication exponent. Then, we restrict the public-key polynomials to $\mathsf{Ker}(J_{\mathbf{o}})$. This yields :

$$\mathbf{P}_{i,\mathsf{Ker}(J_{\mathbf{o}})} = \mathbf{B}^\intercal\mathbf{P}_i\mathbf{B},\ \forall i,\ 1 \leq i \leq m.$$

The restricted public-key $\mathcal{P}_{\mathsf{Ker}(J_{\mathbf{o}})} : \mathbb{F}_q^{(n-m)} \to \mathbb{F}_q^m$ can be computed in polynomial-time $O(mn^\omega)$ and be represented with matrices $\mathbf{P}_{1,\mathsf{Ker}(J_{\mathbf{o}})}, \ldots, \mathbf{P}_{m,\mathsf{Ker}(J_{\mathbf{o}})} \in \mathbb{F}_q^{(n-m)\times(n-m)}$ is a PROV public key with parameters $(q, n-m, m, \delta)$ because $\boldsymbol{O} \subset \mathsf{Ker}(J_{\mathbf{o}})$. Let $\bar{\boldsymbol{O}} \subset \mathbb{F}_q^{n-m}$ be the vanishing subspace of $\mathcal{P}_{\mathsf{Ker}(J_{\mathbf{o}})}$ with $\dim(\bar{\boldsymbol{O}}) = m+\delta$. With our assumption $n \leq 3m$, we obtain $n-m-2m-2\delta \leq 0$. As explained in the specification of PROV [5], the so-called Kipnis-Shamir attack on PROV for parameters

$(q, n, m, \delta)$ has complexity $\mathsf{poly}(n)\, q^{n-2m-2\delta}$. This attack recovers a basis $\mathbf{C} \in \mathbb{F}_q^{(n-m)\times(m+\delta)}$ of the secret subspace $\bar{\boldsymbol{O}}$ in time $\mathsf{poly}(n)$. Then, for all $i$ with $1 \leq i \leq m$, we have

$$(\mathbf{BC})^\intercal \mathbf{P}_i \mathbf{BC} = \mathbf{C}^\intercal (\mathbf{B}^\intercal \mathbf{P}_i \mathbf{B}) \mathbf{C} = \mathbf{C}^\intercal \mathbf{P}_{i,\mathsf{Ker}(J_{\mathbf{o}})} \mathbf{C} = \mathbf{0}_{(m+\delta)\times(m+\delta)}.$$

Namely, the matrix $\mathbf{BC} \in \mathbb{F}_q^{n\times(m+\delta)}$ is a basis of $\boldsymbol{O}$. Multiplying these matrices takes time $O(n^\omega)$ and concludes the proof that the secret-key can be recovered in $O(m\,\mathsf{poly}(n))$.

## 4.3 Experimental results

In this part, we show that our attack is not only efficient from a theoretical point of view but also very practical. To do so, we implemented the attack (Theorem 1) in Sagemath[4] ([16] taking as reference the code used in [12]) with the parameters of PROV suggested in [5] (Table 1). The non-zero vector of the vanishing subspace of the public key is generated with an oracle since such vector is leaked in PROV specification (Sub-Section 4.2).

| Variant | $\lambda$ | $q$ | $n$ | $m$ | $\delta$ | $v$ |
|---|---|---|---|---|---|---|
| PROV-I | 128 | 256 | 136 | 46 | 8 | 82 |
| PROV-III | 192 | 256 | 200 | 70 | 8 | 122 |
| PROV-V | 256 | 256 | 264 | 96 | 8 | 160 |

Table 1: Parameter sets of the PROV signature scheme.

As discussed in [12], the Kipnis-Shamir attack is replaced by the "kernel approach" (see [12] for details). We estimate the performance of the implementation on a single thread of a laptop with an Intel CPU i7-1365U at 5.2GHz and with 32GB of RAM. In Table 2, we report the experimental results obtained. To summarize, we recover the secret-key of PROV in a few seconds for every security level.

| Variant | PROV-I | PROV-III | PROV-V |
|---|---|---|---|
| Time | $1.78.s$ | $4.72.s$ | $7.93s$ |

Table 2: Key-recovery attack of PROV.

## 4.4 Countermeasure

Before presenting the countermeasure, we briefly recall the security model used PROV. One idea of the proof is to model the PROV signature scheme as a weak preimage-sampleable function (WPSF) (Definition 1), denoted $\mathbf{T}_{\texttt{PROV}}$, such as :

- The algorithm $(\mathsf{pk}, \mathsf{sk}) \leftarrow$ **Gen** is the PROV key generation;
- The algorithm **F** evaluates the PROV public-key;
- The algorithm **SampDom** uniformly generates a value in $\mathbb{F}_q^n$;
- The pair of algorithms $\mathbf{I} = (\mathbf{I}^1, \mathbf{I}^2)$ are defined as follows:
  - The algorithm $\mathbf{I}^1$ outputs a uniformly distributed vector $\mathbf{v} \in \mathbb{F}_q^v$;
  - The algorithm $\mathbf{I}^2$ takes as input $(\mathsf{pk}, \mathsf{sk}, \mathbf{v}, \mathbf{y})$ with $\mathbf{v} \in \mathbb{F}_q^v$ and $\mathbf{y} \in \mathbb{F}_q^m$, performs one iteration of the while loop in the signature generation (see Algorithm 1) for the given vector $\mathbf{v}$ and outputs a PROV signature $\mathbf{s} \in \mathbb{F}_q^n$ for $\mathbf{h} = \mathbf{y}$ or $\perp$ if the iteration failed.

One can remark that the model assumes, in particular, that the vinegar vector should be uniform and kept secret to the adversary in PS security of $\mathbf{T}_{\texttt{PROV}}$ (see Definition 2). Precisely, in the $\mathsf{PS}_0$ game, the adversary has access to the oracle $\mathsf{Sample}_0$ (both described in Figure 1). The oracle

[4] Our implementation is available at https://github.com/River-Moreira-Ferreira/prov-attack

$\mathsf{Sample}_0$ keeps secret the value $z_i \leftarrow \mathbf{I}^1$ used for $\mathbf{I}^2$ from the adversary $\mathcal{A}$. Also, the algorithm $\mathbf{I}^1$ uniformly generates the vector $z_i \in \mathbb{F}_q^v$ for $\mathbf{T}_{\texttt{PROV}}$.

The specification of PROV v.1.0 differs from this model as the vinegar vector, which corresponds to a value $z_i \in \mathbb{F}_q^v$, is leaked during signature generation and constant.

The countermeasure appears evident when knowing this flaw in the security model: the vinegar vector should be uniformly generated and kept secret. This tweak will prevent an adversary from recovering easily a vector in the secret linear subspace with the previous strategy and makes PROV immune against our polynomial-time key-recovery attack.

For example, we can suggest generating the vector $\mathbf{v}$ as $\mathcal{H}(3||\mathsf{s}_{\mathsf{sk}}||\mathsf{msg})$ where $\mathsf{s}_{\mathsf{sk}}$ is the secret seed uniformly generated during the key generation (this was the strategy followed by others UOV candidates to the ongoing NIST standardization process). We will obtain a deterministic signature generation, as desired in the PROV specification.

Finally, we have reported this vulnerability to the designer of PROV and they updated the specification (v1.1) with such countermeasure.

## References

1. NIST. Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process.
2. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *J. Math. Cryptol.*, 3(3):177–197, 2009.
3. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In Joris van der Hoeven and Mark van Hoeij, editors, *International Symposium on Symbolic and Algebraic Computation, ISSAC'12, Grenoble, France - July 22 - 25, 2012*, pages 67–74. ACM, 2012.
4. Ward Beullens. Improved cryptanalysis of UOV and Rainbow. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 348–373. Springer, Heidelberg, October 2021.
5. Benoit Cogliati, Jean-Charles Faugère, Pierre-Alain Fouque, Louis Goubin, Robin Larrieu, Gilles Macario-Rat, Brice Minaud, and Jacques Patarin. PROV: Provable unbalanced Oil and Vinegar specification v1.0 – 06/01/2023.
6. Andre Esser, Javier A. Verbel, Floyd Zweydinger, and Emanuele Bellini. ttcryptographicestimators: a software library for cryptographic hardness estimation. *IACR Cryptol. ePrint Arch.*, page 589, 2023.
7. Jean-Charles Faugère, Kelsey Horan, Delaram Kahrobaei, Marc Kaplan, Elham Kashefi, and Ludovic Perret. Fast quantum algorithm for solving multivariate quadratic equations. Cryptology ePrint Archive, Paper 2017/1236, 2017. https://eprint.iacr.org/2017/1236.
8. Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Jean-Philippe Aumasson, Bas Westerbaan, and Ward Beullens. SPHINCS$^+$. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.
9. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 206–222. Springer, Heidelberg, May 1999.
10. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 206–222. Springer, 1999.
11. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.
12. Pierre Pébereau. One vector to rule them all: Key recovery from one vector in uov schemes. Cryptology ePrint Archive, Paper 2023/1131, 2023. https://eprint.iacr.org/2023/1131.

13. Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. On provable security of UOV and HFE signature schemes against chosen-message attack. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*, volume 7071 of *Lecture Notes in Computer Science*, pages 68–82. Springer, 2011.
14. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.
15. Jean-Pierre Serre. *A Course in Arithmetic*, volume 7 of *Graduate Texts in Mathematics*. Springer, New York, NY, 1973.
16. The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.5)*, 2022. https://www.sagemath.org.