MysteryTwister

THE CRYPTO CHALLENGE CONTEST

Vinaigrette signature scheme

Gabriel Sá Diogo, Jonas Schmitt

August 2025

Introduction

In this challenge, we deal with a signature scheme called *Vinaigrette*. It is based on the Unbalanced Oil and Vinegar signature scheme [1] and is inspired by the MAYO signature scheme [2] published 2021-2023.

UOV in a Nutshell

As with UOV, we start with a multivariate quadratic (MQ) mapping $\mathcal{P}\colon \mathbb{F}_q{}^n \to \mathbb{F}_q{}^m$. That is, $\mathcal{P}(x_1, \dots, x_n) = (p_1(x_1, \dots, x_n), \dots, p_m(x_1, \dots, x_n))$, where $x_i \in \mathbb{F}_q$ and p_j are homogeneous polynomials [3] of degree 2 over $\mathbb{F}_q$ in n variables.

For this mapping, there is a linear subspace $\mathcal{O} \subseteq \mathbb{F}_q{}^n$, called the *Oil Space*, on which it vanishes. This means $\mathcal{P}(o) = (0, \dots, 0)$ for all $o \in \mathcal{O}$.

In the signature procedure, $\mathcal{P}$ is the public key and $\mathcal{O}$ is the secret key. The signature for a message λ is created using the *full-domain hash* approach. The signature s is a preimage of $H(\lambda)$ under $\mathcal{P}$, where $H\colon \{0,1\}^* \to \mathbb{F}_q{}^m$ is a suitable hash function. A valid signature is therefore $\mathcal{P}(s) = H(\lambda)$.

To find such an s, first choose an arbitrary $v \in \mathbb{F}_q{}^n$, called the *Vinegar Vector*, and look for an $o \in \mathcal{O}$ with $\mathcal{P}(v + o) = H(\lambda)$. To do this, consider the differential of $\mathcal{P}$, which is bilinear:

$$\mathcal{P}'(x, y) := \mathcal{P}(x + y) - \mathcal{P}(x) - \mathcal{P}(y)$$

Now the following holds:

$$H(\lambda) \overset{!}{=} \mathcal{P}(v + o) = \underbrace{\mathcal{P}(v)}_{\text{constant}} + \underbrace{\mathcal{P}(o)}_{=0} + \underbrace{\mathcal{P}'(v, o)}_{\text{linear in } o}$$

So all you have to do is solve the linear system of equations $\mathcal{P}'(v, o) = H(\lambda) - \mathcal{P}(v)$ for $o \in \mathcal{O}$. The signature for λ is then $v + o$.

Structure of Vinaigrette

The above system of linear equations consists of m equations in $o := \dim(\mathcal{O})$ variables. To guarantee a solution, $o \geq m$ must hold.

However, it becomes easier to determine $\mathcal{O}$ using brute force the larger this subspace is. This is why you want to keep the parameter o as small as possible. So that signatures can still be calculated, Vinaigrette performs a *Whipped Up MQ mapping* $\mathcal{P}^*\colon \mathbb{F}_q{}^{kn} \to \mathbb{F}_q{}^m$:

$$\mathcal{P}^*(\mathrm{x}_1, \dots, \mathrm{x}_k) := \mathcal{P}(\mathrm{x}_1) + \dots + \mathcal{P}(\mathrm{x}_k)$$

where $x_i \in \mathbb{F}_q{}^n$ for all $1 \leq i \leq k$, and k is chosen such that $ko \geq m$. Sums of MQ mappings are again MQ mappings and if $\mathcal{P}$ vanishes on $\mathcal{O}$, then $\mathcal{P}^*$ also vanishes on $\mathcal{O}^*$. Thus, a signature s with $\mathcal{P}^*(s) = H(\lambda)$ can be found in the same way as described above if $\mathcal{O}$ is known. Nevertheless, the description of $\mathcal{P}$ is sufficient. When signing and verifying, the mapping is then “opened” locally.

It should therefore be possible to achieve the same level of security with smaller public keys and smaller signatures.

Hash function used

The hash function used is the extendable-output function (XOF) `SHAKE256`. For Vinaigrette, we take the first $\lceil(\lceil \log_2(q)\rceil * m/8)\rceil$ bytes of output from the function. The output is then broken down into $\lceil \log_2(q)\rceil = 5$-bit pieces. So we have m values $t_i \in \mathbb{F}_q$, which are our $H(\lambda)$.

Format of the public key

The homogeneous polynomials of degree 2 can also be written compactly in matrix-vector notation:

$$p_k(x_1, \dots, x_n) = \sum_{1 \leq i \leq j \leq n} (P_k)_{ij} x_i x_j = x^T P_k x$$

where $x := (x_1, \dots, x_n) \in \mathbb{F}_q{}^n$ and $P_k \in \mathbb{F}_q{}^{n \times n}$. Note that the matrix P_k is an upper triangular matrix, i.e., all entries below the diagonal are 0.

Thus, the MQ mapping $\mathcal{P}$ can be represented by m many $n \times n$ upper triangular matrices $P_1, \dots, P_m$.

In this challenge, the entries of the matrices are specified row by row, starting with the diagonal. The different matrices are separated by a blank line. For example, we save the matrices

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{pmatrix} \quad \begin{pmatrix} 9 & 8 & 7 \\ 0 & 6 & 5 \\ 0 & 0 & 4 \end{pmatrix}$$

in a text file as follows:

```
1 2 3
4 5
6

9 8 7
6 5
4
```

There are, of course, more efficient methods of storing the public key. For the sake of simplicity, however, we will leave it at that. We refer interested readers to the current MAYO specification.

Where did the name come from?

The MAYO signature method uses the same idea of whipping up. However, it uses a more complex construction with so-called *emulsifier matrices*.

Hence the name: mayonnaise is an emulsion of oil and vinegar. If you leave out the emulsifier, you get a vinaigrette instead.

The challenge

Given a public key P and a message λ, the challenge is to forge a valid signature s using the hash function or the XOF `SHAKE256`. The following parameters are used:

$$(q, n, m, o, k) = (31, 62, 60, 6, 10)$$

The public key is located in `public-key.txt`, the message is: `Mayonnaise is not an instrument`

Delivery format of the signature

The signature $s = (s_1, \dots, s_k) \in (\mathbb{F}_q{}^n)^k$ is to be submitted as a simple list $(s_1, \dots, s_k)$. For example, the signature $(s_1, s_2) = ((1, 2, 3), (4, 5, 6))$ is submitted as follows:

```
1 2 3 4 5 6
```

Additional files

- `public-key.txt` - The public key used

References

[1] Wikipedia: Unbalanced oil and vinegar scheme
[2] pqmayo: MAYO
[3] Wikipedia: Homogeneous polynomials