MAYO: Practical Signatures from Oil-and-Vinegar Maps

Ward Beullens   Fabio Campos   Sofía Celi   Basil Hess   Matthias J. Kannwischer

Background of Oil and Vinegar (OV) schemes

Since 1985, various authors have proposed building public key schemes where the public key is a set of **multivariate quadratic equations over a small finite field** K. The general problem of solving such a set of equations is NP-hard and considered a good basis for post-quantum cryptography. The *Oil and Vinegar* scheme (sometimes referred to as *unbalanced Oil and Vinegar*) [5, 6] is one of the earliest signature schemes in this framework.

In the *Oil and Vinegar* scheme, the public key represents a trapdoored homogeneous multivariate map $\mathcal{P}(\mathbf{x}) = (p_1, \dots, p_m) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ which consists of a sequence of m multivariate quadratic polynomials $p_1(\mathbf{x}), \cdots, p_m(\mathbf{x})$ in n variables $\mathbf{x} = (x_1, \cdots, x_n)$. The trapdoor information is a secret subspace $O \subset \mathbb{F}_q^n$ of dimension m, on which $\mathcal{P}(\mathbf{x})$ evaluates to zero. Given a salted hash digest $\mathbf{t} \in \mathbb{F}_q^m$ of a message M, the trapdoor information allows sampling a signature $\mathbf{s}$ such that $\mathcal{P}(\mathbf{s}) = \mathbf{t}$.

To do this, the signer first picks a random vector $\mathbf{v} \in \mathbb{F}_q^n$, and then solves for a vector $\mathbf{o}$ in the oil space O such that $\mathcal{P}(\mathbf{v}+\mathbf{o}) = \mathbf{t}$. In general, for a quadratic maps $\mathcal{P}$ we can define its differential $\mathcal{P}'$ as $\mathcal{P}'(\mathbf{x},\mathbf{y}) := \mathcal{P}(\mathbf{x}+\mathbf{y}) - \mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{y})$, which is a bilinear map. Using $\mathcal{P}'$, it becomes apparent that solving for $\mathbf{o}$ is easy, because

$$\mathcal{P}(\mathbf{v}+\mathbf{o}) = \underbrace{\mathcal{P}'(\mathbf{v},\mathbf{o})}_{\text{Linear in } \mathbf{o}} + \underbrace{\cancel{\mathcal{P}(\mathbf{o})}}_{=0} + \underbrace{\mathcal{P}(\mathbf{v})}_{\text{fixed}} = \mathbf{t}$$

is a system of m linear equations in m variables (since O has dimension m). The signer outputs the signature $\mathbf{s} = \mathbf{v}+\mathbf{o}$. To verify a signature, the verifier simply recomputes $\mathcal{P}(\mathbf{s})$ and the hash digest $\mathbf{t}$, and verifies that they are equal.

A practical drawback is that the public map $\mathcal{P}$ consists of approximately $mn^2/2$ coefficients. We can sample $\mathcal{P}$ such that approximately $m(n^2 - m^2)/2$ of the coefficients can be expanded publicly from a short seed, but the remaining $m^3/2$ coefficient still make for a relatively large public key size. (e. g., 66 KB for 128 bits of security). This problem is solved by our scheme: **MAYO** [1, 2].

A practical scheme: MAYO

MAYO is a variant of the *Oil and Vinegar* scheme whose public keys are smaller. A **MAYO** public key $\mathcal{P}$ has the same structure as an *Oil and Vinegar* public key, except that the dimension of the space O on which $\mathcal{P}$ evaluates to zero is "too small", i.e., $\dim(O) = o$, with o less than m. We explore the scheme below.

MAYO

In **MAYO**, The dimension of the space O is "too small", which makes the problem of recovering O from $\mathcal{P}$ becomes much harder, which allows for smaller parameters. However, since O is "too small", the algorithm to sample a signature $\mathbf{s}$ such that $\mathcal{P}(\mathbf{s}) = \mathbf{t}$ breaks down: the system $\mathcal{P}(\mathbf{v}+\mathbf{o}) = \mathbf{t}$ is now a system of m linear equations in only o variables, so it is very unlikely to have any solutions. We need a new way to produce and verify signatures.

The solution is to publicly "whip up" the oil and vinegar map $\mathcal{P}(\mathbf{x}) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ into a k-fold larger map $\mathcal{P}^*(\mathbf{x}_1, \dots, \mathbf{x}_k) : \mathbb{F}_q^{kn} \to \mathbb{F}_q^m$, where k is a parameter of the scheme. The whipped map $\mathcal{P}^*$ is constructed in such a way that it evaluates to zero on the subspace $O^k = \{(\mathbf{o}_1, \dots, \mathbf{o}_k) \mid \forall i : \mathbf{o}_i \in O\}$ which has dimension ko. Concretely, we define:

$$\mathcal{P}^*(\mathbf{x}_1, \dots, \mathbf{x}_k) := \sum_{i=1}^{k} \mathbf{E}_{ii}\mathcal{P}(\mathbf{x}_i) + \sum_{i=1}^{k} \sum_{j=i+1}^{k} \mathbf{E}_{ij}\mathcal{P}'(\mathbf{x}_i, \mathbf{x}_j)$$

where the $\mathbf{E}_{ij} \in \mathbb{F}_q^{m \times m}$ are fixed public matrices (referred to as $\mathbf{E}$-matrices), and $\mathcal{P}'(\mathbf{x},\mathbf{y})$, the differential of $\mathcal{P}$, is defined as $\mathcal{P}'(\mathbf{x},\mathbf{y}) := \mathcal{P}(\mathbf{x}+\mathbf{y}) - \mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{y})$. We choose parameters such that $ko > m$ to make sure that the space O^k is large enough so that the signer can sample signatures $\mathbf{s} = (\mathbf{s}_1, \cdots, \mathbf{s}_k)$ such that $\mathcal{P}^*(\mathbf{s}) = \mathbf{t}$ with the usual *Oil and Vinegar* approach. The signer first samples $(\mathbf{v}_1, \dots, \mathbf{v}_k) \in \mathbb{F}_q^{kn}$ at random, and then solves for $(\mathbf{o}_1, \dots \mathbf{o}_k) \in O^k$ such that

$$\mathcal{P}^*(\mathbf{v}_1 + \mathbf{o}_1, \dots, \mathbf{v}_k + \mathbf{o}_k) = \mathbf{t}$$

which is a system of m linear equations in ko variables.

Parameter sets of MAYO

We chose 4 parameter sets in accordance to security levels 1, 3, and 5, which seem to work pretty good in many network protocols.

Parameter set of scheme	MAYO$_1$	MAYO$_2$	MAYO$_3$	MAYO$_5$
Security level of scheme	1	1	3	5
n	66	78	99	133
m	64	64	96	128
o	8	18	10	12
k	9	4	11	12
q	16	16	16	16
salt_bytes	24	24	32	40
digest_bytes	32	32	48	64
pk_seed_bytes	16	16	16	16
$f(z)$	$f_{64}(z)$	$f_{64}(z)$	$f_{96}(z)$	$f_{128}(z)$
Secret key size	24 B	24 B	32 B	40 B
Public key size	1168 B	5488 B	2656 B	5008 B
Signature size	321 B	180 B	577 B	838 B
Expanded **sk** size	69 KB	92 KB	230 KB	553 KB
Expanded **pk** size	70 KB	97 KB	233 KB	557 KB

Table 1. Parameter sets for **MAYO**. All sizes are reported in bytes (B) or kilobytes (KB).

Performance (AVX2)

Following the work of [3], we present the following results on Intel Skylake and Icelake using a nibble-sliced implementation with the Method of the 4 Russians (M4R).

Nibble Representation (M4R)

	Scheme	`KeyGen`	`ExpandSK`	`ExpandPK`	`ExpandSK + Sign`	`ExpandPK + Verify`
Skylake	MAYO$_1$	73 668	82 820	43 970	283 126	83 846
	MAYO$_2$	144 508	154 002	59 178	324 402	84 974
	MAYO$_3$	295 606	358 416	147 758	920 944	344 994
	MAYO$_5$	642 690	889 100	355 238	1 737 426	706 316
Ice Lake	MAYO$_1$	43 550	53 710	22 432	218 300	53 660
	MAYO$_2$	86 014	98 402	30 244	239 852	47 360
	MAYO$_3$	169 258	237 450	74 992	718 586	205 938
	MAYO$_5$	369 898	517 660	180 568	1 244 038	401 310

Table 2. Performance of **MAYO** in CPU cycles on Intel Xeon E3-1245 v5 (Skylake) and Xeon Gold 6338 (Ice Lake) using the nibble representation.

Comparison with other schemes (AVX2)

Type	Sec. Lvl.	Key Gen.	Sign	Verify
MAYO [2] (default/pre-expanded)				
MAYO$_1$	1	44*k*/44*k*	218*k*/165*k*	54*k*/31*k*
MAYO$_2$	1	86*k*/86*k*	240*k*/142*k*	47*k*/17*k*
MAYO$_3$	3	169*k*/169*k*	719*k*/481*k*	206*k*/131*k*
MAYO$_5$	5	370*k*/370*k*	1 244*k*/726*k*	401*k*/221*k*
Oil and Vinegar [4] (pkc+skc/classic)				
`ovIp`	1	2 316*k*/2 341*k*	1 548*k*/79*k*	168*k*/58*k*
`ovIs`	1	3 715*k*/3 734*k*	2 063*k*/83*k*	203*k*/46*k*
`ovIII`	3	13 168*k*/12 832*k*	8 293*k*/243*k*	679*k*/197*k*
`ovV`	5	34 989*k*/35 792*k*	18 802*k*/462*k*	1 514*k*/364*k*
Dilithium				
dilithium2	2	81*k*	219*k*	79*k*
dilithium3	3	137*k*	355*k*	129*k*
dilithium5	5	212*k*	420*k*	204*k*

Table 3. **MAYO** performance in CPU cycles using AVX2 optimizations in comparison with other post-quantum signature schemes running on Intel Ice Lake (Xeon Gold 6330). Dilithium, Falcon and SPHINCS+ benchmarks use libOQS v0.9.0-rc1 with AVX2 optimized code.

Performance (Arm Cortex-M4)

Type	Sec. Level	Key Gen.	Sign	Open
MAYO				
MAYO$_1$	1	4 410*k*	8 270*k*	4 808*k*
MAYO$_1$-pre	1	4 410*k*	3 888*k*	1 709*k*
MAYO$_2$	1	8 847*k*	9 916*k*	5 102*k*
MAYO$_2$-pre	1	8 847*k*	2 761*k*	952*k*
MAYO$_3$	3	15 972*k*	27 401*k*	15 573*k*
MAYO$_3$-pre	3	15 972*k*	10 204*k*	5 102*k*
Oil and Vinegar				
`ovIp` (classic)	1	138 833k	2 482*k*	995*k*
`ovIp` (pkc+skc)	1	175 021*k*	88 757*k*	11 551*k*
`ovIs` (classic)	1	195 744*k*	2 374*k*	616*k*
`ovIs` (pkc+skc)	1	296 161*k*	113 446*k*	16 045*k*
Dilithium				
dilithium2	2	1 598*k*	4 093*k*	1 572*k*
dilithium3	3	2 827*k*	6 623*k*	2 692*k*
Falcon				
falcon-512	1	163 994k	39 014k	473k
SPHINCS+				
sha256-128f-simple	1	15 388*k*	382 534*k*	21 151*k*
sha256-128s-simple	1	985 367*k*	7 495 604*k*	7 166*k*

Table 4. **MAYO** performance on Cortex-M4 in comparison to other post-quantum signature schemes. **MAYO** *pre* variants refer to pre-expanded public and secret keys in a similar fashion as *classic* OV.

Advantages

- **Small key and signature sizes.** **MAYO** offers some of the smallest sizes of all current candidates.
- **Computational efficiency.** **MAYO** performance is competitive with Dilithium on big CPUs.
- **Flexibility.** **MAYO** parameter sets are easily adjusted to reach a specific security level.
- **Wide security margin.** Known attacks against **MAYO** are well-understood and easy to analyze.

References

[1] Ward Beullens. MAYO: Practical post-quantum signatures from oil-and-vinegar maps. pages 355–376, 2022.

[2] Ward Beullens, Fabio Campos, Sofía Celi, Basil Hess, and Matthias Kannwischer. MAYO. MAYO specification, 2023. `https://pqmayo.org/assets/specs/mayo.pdf`.

[3] Ward Beullens, Fabio Campos, Sofía Celi, Basil Hess, and Matthias J. Kannwischer. Nibbling MAYO: Optimized implementations for AVX2 and Cortex-M4. Cryptology ePrint Archive, Paper 2023/1683, 2023. `https://eprint.iacr.org/2023/1683`.

[4] Ward Beullens, Ming-Shing Chen, Shih-Hao Hung, Matthias J. Kannwischer, Bo-Yuan Peng, Cheng-Jhih Shih, and Bo-Yin Yang. Oil and vinegar: Modern parameters and implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(3):321–365, Jun. 2023.

[5] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. pages 206–222, 1999.

[6] Jacques Patarin. The Oil and Vinegar signature scheme. In *Dagstuhl Workshop on Cryptography September, 1997*, 1997.

https://pqmayo.org/assets/specs/mayo.pdf   NIST Workshop 2024   contact@pqmayo.org