Breaking UOV Encryption: Key Recovery Attack On Olivier

Emanuele Cornaggia

emanuele.cornaggia@gmail.com

Abstract. The Oil and Vinegar (OV)[Pat97] trapdoor is widely used in signature schemes such as UOV[KPG99] and MAYO[Beu22]. Recently, *Esposito et al.* proposed OliVier[EFR24], an encryption scheme based on this trapdoor. However, the OV trapdoor was originally designed for signatures, and adapting it to encryption introduces inherent challenges.
We identify two such challenges and analyze how OliVier addresses the first, while showing that the unresolved second challenge enables a practical key-recovery attack. We conclude that any scheme using the OV trapdoor for encryption must also solve this second problem, for which no efficient solution is currently known.

Keywords: Post-Quantum Cryptography · Multivariate Cryptography · UOV

1 Introduction

The Oil and Vinegar (OV) trapdoor, introduced by Patarin in 1997[Pat97], is used to construct systems of multivariate quadratic (MQ) polynomials that are indistinguishable from random ones, and that can be efficiently solved with knowledge of a secret structure. The scheme considers a system of m equations in n variables, yielding a central map $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ whose components $f_1, \dots, f_m$ are chosen uniformly at random of the form

$$f_i(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=i}^{n-m} \alpha_{i,j} x_i x_j$$

The variables $x_1, \dots, x_{n-m}$ are called vinegar variables, while $x_{n-m+1}, \dots, x_n$ are called oil variables. By construction, no quadratic term contains two oil variables.
To hide this structure, the central map $\mathcal{F}$ is composed with a random invertible linear transformation $\mathcal{T} \in \mathrm{GL}_n(\mathbb{F}_q)$.
The private key consists of the pair $(\mathcal{F}, \mathcal{T})$, while the public key is the composed map $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$.
In the signature scheme, to sign a message m, the signer randomly samples vinegar values, then solves the system $\mathcal{P}(\mathbf{x}) = m$, that due to the special structure of the Oil and Vinegar polynomials it has now become linear and it can be solved efficiently using Gaussian elimination.

Beullens[Beu21] simplified the description of the scheme by observing that there exists a secret linear space $\mathcal{O} \subset \mathbb{F}_q^n$ of dimension $\dim(\mathcal{O}) = m$ such that $\mathcal{P}(\mathbf{o}) = \mathbf{0}$ for all $\mathbf{o} \in \mathcal{O}$. Given a target $\mathbf{t} \in \mathbb{F}_q^m$, one can find the preimage $\mathbf{x} \in \mathbb{F}_q^n$ by solving a linear system for $\mathbf{o}$ in

$$\mathcal{P}(\mathbf{v}+\mathbf{o}) = \underbrace{\mathcal{P}(\mathbf{v})}_{\text{fixed by choice of } \mathbf{v}} + \underbrace{\mathcal{P}(\mathbf{o})}_{=0} + \underbrace{\mathcal{P}'(\mathbf{v},\mathbf{o})}_{\text{linear function of } \mathbf{o}} = \mathbf{t}$$

where $\mathbf{v} \in \mathbb{F}_q^n$ is a vinegar vector in the isomorphic system, and $\mathcal{P}'$ is the polar form of $\mathcal{P}$. From this observation, *Beullens* noted that the linear subspace $\mathcal{O}'$ on which the central map $\mathcal{F}$ vanishes on, which is given by all the vectors whose first $n-m$ entries are zero, i.e. $\mathcal{O}' = \{\mathbf{v} | v_i = 0 \text{ for all } i \leq n-m\}$, is related to $\mathcal{O}$ by the formula $\mathcal{O} = \mathcal{T}^{-1}(\mathcal{O}')$.

Beullens' formula works nicely as when $\mathbf{v}$ is fixed the system $\mathcal{P}(\mathbf{v} + \mathbf{o} = \mathbf{t})$ has a unique solution with probability roughly $1 - 1/q$. If the solution is not unique, simply sample a new $\mathbf{v}$ and try again.
However, when this mechanism is used as the basis for an encryption scheme, the situation changes fundamentally, and two issues must be addressed:

1. **Uniqueness of the preimage** In the signature scheme, we want a message (the target) to have multiple possible signatures (preimages). This is achieved by constructing a central map with more variables than equations. In an encryption scheme, however, each ciphertext (target) must correspond to a unique plaintext (preimage), so that both parties can be certain they derive the same message. This requires adjusting the construction to avoid multiple valid preimages.

2. **Choice of vinegar** In the signature scheme, the holder of the secret key is the one who fixes the vinegar variables. In the encryption scheme, by contrast, the vinegar values are indirectly chosen by the party generating the ciphertext, simply by selecting the message to encrypt. This happens implicitly, since the sender does not know how the secret key splits a vector $\mathbf{x}$ into vinegar and oil components $\mathbf{x} = \mathbf{v} + \mathbf{o}$. Therefore, the holder of the private key must be able to efficiently determine the correct vinegar values in order to solve the system and recover the original message.

2 Olivier

OliVier[EFR24] is a public-key multivariate encryption scheme proposed by *Esposito et al.* that exploits the OV trapdoor to recover the message from a ciphertext.

The public key is the polynomial map defined as:

$$\mathcal{P} = \begin{cases} \mathcal{F} \circ \mathcal{T} + \Lambda\mathcal{Q} \\ \mathcal{Q} \end{cases}$$

where:

- $\mathcal{F} \circ \mathcal{T}$ is an OV map
- $\mathcal{Q}$ is a map consisting of random polynomials
- Λ is a coefficient map

The secret key is $(\mathcal{F}, \mathcal{T}, \Lambda)$.

The authors address the issue of preimage uniqueness by choosing the OV map $\mathcal{F} \circ \mathcal{T}$ to be a system of n equations in n variables, ensuring that each output admits only a small number of possible preimages within the OV system. To further eliminate these remaining candidates and achieve uniqueness with high probability (so that, for appropriate security parameters, collisions become negligible), the scheme also incorporates the evaluation of the vector on the random polynomials $\mathcal{Q}$.
Because the dimension of the system changes, the numbers of vinegar and oil variables also change. We therefore denote them by v and o, with the relation $n = v + o$.

To recover the correct vinegar values, the authors propose a brute-force search over all possible vinegar assignments. As a consequence, for the system to remain efficient, the value of v must be kept relatively small. This makes the OV map $\mathcal{F} \circ \mathcal{T}$ inherently weak, but the authors argue that the addition of the term $\Lambda\mathcal{Q}$ results in a secure overall system.

To encrypt a message $\mathbf{x}$, one simply evaluates the polynomial map $\mathcal{P}(\mathbf{x})$.

To decrypt a ciphertext, one first splits it into two parts: the first n coordinates contain the evaluation

$$(\mathcal{F} \circ \mathcal{T})(\mathbf{x}) + \Lambda\, \mathcal{Q}(\mathbf{x}),$$

while the last u coordinates contain the values $\mathcal{Q}(\mathbf{x})$. The second part is then multiplied by the coefficient matrix Λ and subtracted from the first part, thereby isolating the evaluation of the OV system (i.e. $\mathcal{F} \circ \mathcal{T})(\mathbf{x})$).

Next, all possible vinegar values are enumerated. For each choice, the resulting linear system is solved. The candidate solution $\mathbf{x}'$ to the OV system is then evaluated under $\mathcal{Q}$, and if $\mathcal{Q}(\mathbf{x}') = \mathcal{Q}(\mathbf{x})$, the candidate $\mathbf{x}'$ is accepted as the correct preimage.

To accelerate the search over vinegar values, the authors propose providing multiple ciphertexts as input to the decryption procedure. This increases the probability that the correct vinegar assignment is found early for at least one of the ciphertexts, thereby reducing the overall decryption time. If t ciphertexts are provided, the expected number of vinegar iterations decreases from $\frac{q^v}{2}$ to $\frac{q^v}{h+1}$, where h denotes the number of ciphertexts whose corresponding plaintexts require distinct vinegar values (for practical parameter choices, $k \approx t$).

3 Key Recovery Attack

We begin by noting that the oil subspace contains q^o vectors out of a total of q^n vectors. Hence, a uniformly random vector lies in the oil subspace with probability $q^{\,o-n} = q^{-v}$. Consequently, one expects to find a vector in the oil subspace after about q^v random samples on average. For the proposed parameters over $\mathbb{F}_2$ with $v = 24$, this yields a complexity factor of 2^{24}, which is insufficient to ensure security.

However, when the attacker computes $\mathcal{P}(\mathbf{o})$, the result is not 0 but rather $\Lambda\mathcal{Q}(\mathbf{o})$ due to the contribution of the random polynomials. As a result, the attacker cannot determine whether a given vector lies in the oil space or not.

Nevertheless, we can observe that if an attacker manages to obtain several oil vectors, then Λ can be recovered. Once $\Lambda\mathcal{Q}$ is removed from $\mathcal{P}$, the weak OV map $\mathcal{F} \circ \mathcal{T}$ is revealed, and it can then be broken in polynomial time using the Kipnis-Shamir attack[KS98].

Each MQ polynomial $\mathcal{P}^{(i)}$ can be expressed in the form

$$x_{1,1}a_{1,1} + x_{1,2}a_{1,2} + \cdots + x_{n,n}a_{n,n} + \lambda_1^{(i)}\mathcal{Q}_1(\mathbf{x}^{(k)}) + \cdots + \lambda_u^{(i)}\mathcal{Q}_u(\mathbf{x}^{(k)}) = \mathcal{P}^{(i)}(\mathbf{x}^{(k)}),$$

where $x_{r,s} = x_r x_s$, with x_r and x_s being the r-th and s-th coordinates of the vector $\mathbf{x}^{(k)}$. The coefficient of the monomial $x_{r,s}$ is denoted by $a_{r,s}$, and λ_j is the coefficient of $\mathcal{Q}_j$ in the j-th row of Λ.

Evaluating this polynomial at z distinct vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(z)}$ yields the following linear

system:

$$\begin{bmatrix} x_{1,1}^{(1)} & x_{1,2}^{(1)} & \cdots & x_{n,n}^{(1)} & \mathcal{Q}_1(\mathbf{x}^{(1)}) & \cdots & \mathcal{Q}_u(\mathbf{x}^{(1)}) \\ x_{1,1}^{(2)} & x_{1,2}^{(2)} & \cdots & x_{n,n}^{(2)} & \mathcal{Q}_1(\mathbf{x}^{(2)}) & \cdots & \mathcal{Q}_u(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{1,1}^{(z)} & x_{1,2}^{(z)} & \cdots & x_{n,n}^{(z)} & \mathcal{Q}_1(\mathbf{x}^{(z)}) & \cdots & \mathcal{Q}_u(\mathbf{x}^{(z)}) \end{bmatrix} \begin{bmatrix} a_{1,1}^{(i)} \\ a_{1,2}^{(i)} \\ \vdots \\ a_{n,n}^{(i)} \\ \lambda_1^{(i)} \\ \vdots \\ \lambda_u^{(i)} \end{bmatrix} = \begin{bmatrix} \mathcal{P}^{(i)}(\mathbf{x}^{(1)}) \\ \mathcal{P}^{(i)}(\mathbf{x}^{(2)}) \\ \vdots \\ \mathcal{P}^{(i)}(\mathbf{x}^{(z)}) \end{bmatrix}. \tag{1}$$

If the sampled vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(z)}$ are chosen such that the associated vectors of linearized monomials $\mathbf{y}^{(k)} = [x_{1,1}^{(k)}, \ldots, x_{n,n}^{(k)}]$ are linearly independent, then we can have at most $z = \frac{n(n+1)}{2}$ such vectors, corresponding to the full dimension of the space of quadratic monomials.
In this case, the linear system contains z equations in $z + u$ unknowns, and therefore it has $q^{(z+u)-z} = q^u$ solutions.

If, instead of evaluating at arbitrary vectors, we evaluate at vectors $\mathbf{o} \in \mathcal{O}$, we may exploit the property

$$\sum x_{r,s} a_{r,s} = 0.$$

In this case, each polynomial reduces to

$$\lambda_1^{(i)} \mathcal{Q}_1(\mathbf{o}) + \cdots + \lambda_u^{(i)} \mathcal{Q}_u(\mathbf{o}) = \mathcal{P}^{(i)}(\mathbf{o}),$$

and the corresponding linear system becomes

$$\begin{bmatrix} \mathcal{Q}_1(\mathbf{o}^{(1)}) & \cdots & \mathcal{Q}_u(\mathbf{o}^{(1)}) \\ \mathcal{Q}_1(\mathbf{o}^{(2)}) & \cdots & \mathcal{Q}_u(\mathbf{o}^{(2)}) \\ \vdots & & \\ \mathcal{Q}_1(\mathbf{o}^{(z)}) & \cdots & \mathcal{Q}_u(\mathbf{o}^{(z)}) \end{bmatrix} \begin{bmatrix} \lambda_1^{(i)} \\ \vdots \\ \lambda_u^{(i)} \end{bmatrix} = \begin{bmatrix} \mathcal{P}^{(i)}(\mathbf{o}^{(1)}) \\ \mathcal{P}^{(i)}(\mathbf{o}^{(2)}) \\ \vdots \\ \mathcal{P}^{(i)}(\mathbf{o}^{(z)}) \end{bmatrix}. \tag{2}$$

For the case $z = u$, provided that the vectors $\mathbf{y}^{(k)}$ associated with the samples $\mathbf{o}^{(k)}$ are linearly independent, the system becomes a square system of u equations in u unknowns, thus admitting a unique solution.
Since at most $\frac{o(o+1)}{2}$ such vectors can be linearly independent, we require $\frac{o(o+1)}{2} \geq u$.

A multivariate quadratic system becomes overdetermined and vulnerable to linearization attacks (like the XL algorithm[CKPS00]) if the number of equations, m, approaches the square of the number of variables, n (i.e. $m \sim n^2$).

In OliVier, where $o \sim n$ and the total equations are defined as $m = n + u$, we must constrain u. If u is increased such that it exceeds the number of quadratic terms $\frac{o(o+1)}{2}$, the system provides enough equations to treat each quadratic monomial as a unique linear variable.

Therefore, to prevent the system from becoming trivially solvable via linearization, the condition:

$$\frac{o(o+1)}{2} \geq u$$

must be satisfied.

3.1 How to find vectors of the Oil Space

When the secret key holder attempts to decrypt a ciphertext obtained by encrypting a vector from the oil space, the removal of the $\Lambda\mathcal{Q}(\mathbf{o})$ contribution yields the zero vector (since, by construction, $\mathcal{P}(\mathbf{o}) = \mathbf{0} + \Lambda\mathcal{Q}(\mathbf{o})$). Thus, the only information revealed is that the target vector lies in the oil space and the value of its evaluation under the random polynomials $\mathcal{Q}$. Consequently, recovering the correct vector requires iterating over all possible $\mathbf{o}$ and checking, for each candidate, whether $\mathcal{Q}(\mathbf{o}) = \mathcal{Q}(\mathbf{x})$.

In practice, this exhaustive search is infeasible: the scheme already requires a small value of v, and choosing a small oil dimension o would render the system vulnerable to direct attacks. For example, under the proposed parameters $v = 24$ and $q = 2$, the corresponding oil dimension is $o = 296$, which would require 2^{296} operations, clearly infeasible. Therefore, when the decryption algorithm receives a ciphertext corresponding to a vector in the oil space, it must abort.

If the system is implemented such that decryption failure is easily detectable (for instance, after transforming the scheme into a KEM, where both parties must derive a shared key and decryption failure prevents further symmetric communication), then the decryption routine effectively becomes an oracle: it returns a valid output when the ciphertext corresponds to a vector outside the oil space and triggers an abort when the ciphertext corresponds to a vector inside the oil space.

Since we need u vectors from the oil space to recover Λ, and a random vector lies in the oil space with probability q^{-v}, the expected number of oracle queries required is uq^v.

Because OliVier allows (and is more efficient with) multiple ciphertexts provided simultaneously to the decryption function, one might imagine modifying the decryption routine so that it requires a batch of ciphertexts in order to proceed, thereby making the abort probability negligible. However, an attacker can exploit this setting with the same complexity.

Suppose an adversary encrypts t random vectors and submits their ciphertexts. The decryption function returns one plaintext, which the adversary now knows is not in the oil space. The adversary can then sample another random vector, compute its ciphertext, and submit this new ciphertext together with the previous ones, excluding the one whose value has already been decrypted. Since the decryption function never returns a vector in the oil space, the attacker can eventually replace all t vectors with vectors from the oil space; once this happens, the decryption function will fail.

We now compute the expected complexity of this attack.
The adversary samples t random vectors independently, each lying in the oil space with probability q^{-v}. Thus, the expected number of vectors not in the oil space is

$$t - \frac{t}{q^v}.$$

When replacing one such vector with a freshly sampled one, the probability that the new vector lies in the oil space is q^{-v}, so on average q^v queries are required per vector. Hence, the expected number of queries needed to obtain t oil vectors is

$$\left(t - \frac{t}{q^v}\right) q^v = t(q^v - 1).$$

After this point the adversary possesses t oil vectors.
Since Λ recovery requires u such vectors, if $t < u$ the adversary simply continues replacing plaintexts and keeps any additional oil vectors found. This requires an additional $(u-t)q^v$

queries. Summing the contributions, the total expected number of queries is

$$t(q^v - 1) + (u - t)q^v = uq^v - t.$$

This quantity is clearly dominated by q^v, so the complexity of recovering Λ is

$$O(q^v).$$

If we increase the security parameters to achieve resistance against this attack, we necessarily increase the complexity of the decryption procedure. Requiring the attack to need 2^{128} queries implies that the decryption function must effectively iterate over 2^{128} possible values (although the time complexity of the decryption step is $\frac{q^v}{h+1}$, we disregard the term h since it is linear and cannot be made large without rendering the ciphertext length unacceptably large). We thus claim that the attack breaks OliVier.

In order to address this issue, one would need to find a more efficient method to recover the vinegar vector from the ciphertext, rather than attempting all possible combinations. Further research is required to determine whether there exist algebraic methods that enable this.

4 Conclusion

We have identified two fundamental challenges that arise when using the Oil and Vinegar (OV) trapdoor for encryption, which was originally designed for signatures. From a construction perspective, OliVier successfully addresses the first challenge; however, the second remains unresolved, enabling a practical key-recovery attack.

While OliVier serves as a representative example, this issue is inherent to any scheme that employs the OV trapdoor for encryption and attempts to iterate over all possible vinegar vectors to recover the correct one during decryption. The main obstacle is that the probability of a random vector lying in the oil space, q^{-v}, governs the complexity of the decryption algorithm, and we have demonstrated a Key Recovery Attack that links these two complexities. Consequently, increasing q^v to achieve higher security also renders the scheme impractical.

We therefore conclude that any encryption scheme leveraging the OV trapdoor must address this second challenge, for which no efficient solution is currently known, in order to ensure both security and practicality.

References

[Beu21] 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, Cham, October 2021.

[Beu22] Ward Beullens. MAYO: Practical post-quantum signatures from oil-and-vinegar maps. In Riham AlTawy and Andreas Hülsing, editors, *SAC 2021*, volume 13203 of *LNCS*, pages 355–376. Springer, Cham, September / October 2022.

[CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 392–407. Springer, Berlin, Heidelberg, May 2000.

[EFR24] Antonio Corbo Esposito, Rosa Fera, and Francesco Romeo. Olivier: an oil and vinegar based cryptosystem. arXiv preprint arXiv:2405.08375, 2024. arXiv:2405.08375 [math.AC], submitted 14 May 2024.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In Jacques Stern, editor, *EUROCRYPT 1999, Lecture Notes in Computer Science, Vol. 1592*, pages 206–222. Springer, Heidelberg, 1999.

[KS98] Aviad Kipnis and Adi Shamir. Cryptanalysis of the Oil & Vinegar signature scheme. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 257–266. Springer, Berlin, Heidelberg, August 1998.

[Pat97] Jacques Patarin. The oil and vinegar signature scheme. In *Dagstuhl Workshop on Cryptography*, September 1997. September, 1997.