Mayonnaise Sandwich Disablement via Electric Shock: A Review of Theoretical Fault Injection attacks on Post-Quantum MAYO Encryption Scheme

Alp Berrak
Northeastern University
Boston, MA
berrak.a@northeastern.edu

Ethan Friedman
Northeastern University
Boston, MA
friedman.et@northeastern.edu

December 11, 2025

Abstract

As quantum computing poses threats to traditional cryptographic schemes, the National Institute of Standards and Technology (NIST) is evaluating post-quantum signature schemes. MAYO, as part of this ongoing evaluation, is a multivariate-based signature scheme, one of multiple standardized primitives providing an alternative to lattice-based constructions. Because Post-Quantum Cryptography (PQC) must remain secure into the future, analyzing the resilience of candidates such as MAYO to mathematical and physical attacks is essential before standardization. This paper details a theoretical attack on the MAYO signature scheme proposed by Yadi Zhong from Auburn University [1]. Zhong's paper focuses on MAYO's deterministic mode, specifically the generation of "free variables" that allow for multiple valid signatures for a single message. The attack exploits MAYO's whipped structure, where the number of free variables ranges from 2 to 20 across different parameter sets. With fault injection, an adversary would be able to modify these free variables during the signing phase, causing MAYO to output multiple valid signatures that can then be compared in order to extract information about the secret key. In this paper, we review and analyze Zhong's work as well as provide preliminary knowledge/explanation of the foundations of the MAYO signature scheme.

1 Introduction

The anticipated development of large-scale quantum computers threatens many public-key encryption algorithms that rely primarily on number-theoretic assumptions, such as integer factorization (RSA) and the discrete logarithm problem (DLP). Shor's algorithm, discovered in 1994, showed that a sufficiently powerful quantum computer could solve these problems in polynomial time, rendering widely deployed digital signature and key-exchange schemes insecure. In response, NIST initiated a multi-year Post-Quantum Cryptography (PQC) standardization effort in December 2016. The goal of NIST's process is to evaluate, test, and ultimately standardize new public-key encryption (PKE), key encapsulation (KEM), and digital schemes that remain secure even in the presence of an adversary equipped with a quantum computer. NIST's standards are published in the Federal Information Processing Standards (FIPS) series and are backed by open evaluation rounds, public comment, and extensive third-party cryptanalysis. The finalization of the first PQC standards occurred in FIPS 203 (ML-KEM, Kyber) and FIPS 204 (ML-DSA, Dilithium) in 2024, with additional signature candidates, such as Falcon and MAYO, undergoing further evaluation for future inclusion.

2 Background and Preliminary Knowledge

2.1 Unbalanced Oil and Vinegar (UOV)

The Oil and Vinegar (OV) signature scheme, introduced by Patarin in 1997 [3], is one of the most prominent families of multivariate quadratic (MQ) signature schemes. Its security is rooted in the difficulty of solving systems of quadratic polynomial equations over finite fields. This is a problem that is NP-hard for quantum computers. The most common version, Unbalanced Oil and Vinegar (UOV), modifies the original construction. This is done by selecting more vinegar variables than oil variables, improving both efficiency and resistance to known algebraic attacks. In UOV, the secret key consists of a hidden decomposition of the public multivariate map into affine transformations and a central multivariate quadratic map structured according to the oil-vinegar partition. The variable set is divided into: Vinegar variables (v): randomly chosen values that, once fixed, linearize the system. Oil variables (o): variables whose values are solved for using the now-linear system.

2.2 UOV Signing and Verification Procedure

The Signing Procedure: To sign a message, the signer first hashes the message to a target vector and then repeatedly samples random vinegar variables. For each vinegar choice, the resulting system of linear equations in the oil variables is solved. If a solution exists, its concatenation with the vinegar values yields a valid signature. If no solution exists, then the signer repeats with a new vinegar vector. This inherent randomness is essential for both correctness and security.

The Verification Procedure: The verifier applies the public multivariate quadratic map $P : F_q^{v+o} \to F_q^m$ to the candidate signature and checks whether the result matches the message hash. Because the public key hides the internal oil-vinegar structure, recovering the secret oil space from only the public polynomials is believed to be computationally infeasible.

UOV has remained attractive in the post-quantum landscape due to its mathematical simplicity, long history of cryptanalysis, and comparatively small signature sizes. Regardless, due to UOV's large public keys and weakness to specific algebraic attacks, schemes like MAYO were motivated to explore ways to reduce key sizes and strengthen internal structure while preserving the security benefits of multivariate designs.

2.3 MAYO Signature Scheme

MAYO is a recently proposed multivariate signature scheme built as a modernized variant of UOV, motivated by the need to reduce public key sizes and introduce new structural defenses while preserving post-quantum security. Like UOV, MAYO relies on the hardness of the Multivariate Quadratic (MQ) problem over finite fields. Still, it introduces several modifications designed to make the scheme more efficient and resistant to algebraic and structural attacks.

At a high level, MAYO modifies the oil-vinegar framework by reducing the size of the oil space and compensating for this reduction through a set of whipping and emulsifying transformations applied during key generation and signing. These transformations introduce additional mixing between the oil and vinegar variables, thus allowing the scheme to maintain security despite having fewer oil variables than classical UOV typically requires. MAYO also introduces an "emulsifier matrix," a public linear transformation designed to obscure relationships between internal variables while keeping verification

efficient.

Key generation proceeds by selecting dimensions for oil and vinegar variables, sampling the central UOV-like quadratic map, and then applying two affine transformations. One in the input space and one in the output space. These hide the internal structure, producing the public key as a system of quadratic polynomials. The goal is to substantially reduce public key size relative to standard UOV while maintaining or improving security at NIST Level 1 and Level 3 targets. [8]

Signing retains the classical oil-vinegar workflow of repeatedly sampling vinegar variables. The signer first hashes the message to a target vector. For each sampled vinegar vector, the quadratic system collapses into a linear system for the oil variables. Whipping modifies this linear system to produce a system with multiple valid solutions rather than a single one. When the coefficient matrix A has full row rank, all solutions are valid, and the signing algorithm uses a randomness vector r to deterministically select one particular solution from this set. If a solution exists, its concatenation with the vinegar values yields a valid signature. New vinegar values are sampled if no valid solution is found. MAYO has two modes, deterministic and random. Random mode uses a fresh random value for r each time a message is signed, thus signing the same message twice will produce different signatures (higher chance of collision depending on the security parameters). While deterministic mode sets r to 0 for each signed message, meaning each message will always output the same signature.

Verification works by evaluating the public map on the signature and checking whether the result matches the target message hash vector. Because the emulsifier matrix keeps a structure that allows for quick evaluation of the quadratic map, verification is still efficient despite the additional transformations.

Similar to the NIST PQC competition, MAYO's security parameters are made to match both classical and quantum security goals. Using a multivariate foundation instead of lattice-based assumptions, the scheme seeks to achieve security levels comparable to those targeted by Dilithium and Falcon.

The emulsifier matrix is a publicly visible linear transformation applied in the formation of the central map. Although it is public, it is designed to obscure the internal oil-vinegar decomposition and the relationships formed through whipping.

MAYO's core relies on multivariate quadratic maps. A quadratic map $P : F_q^n \to F_q^m$ can be expressed as a vector of quadratic polynomials $P_i(x) =$

$x^T M_i x$, where M_i are upper triangular matrices representing the quadratic coefficients. The polar form of a quadratic map (defined as $P_0(x, y) = P(x + y) - P(x) - P(y)$) is bilinear and linear in each argument separately. In the signing procedure, choosing vinegar variables v converts the quadratic preimage problem into a linear system over the oil variables, which can then be solved efficiently. The rank of A determines the properties of the resulting linear system $Ax = y$ when the vinegar variables are fixed. In other words, the solution space is an affine subspace of dimension $N-r$ with q^{N-r} distinct solutions if A has rank r. UOV typically yields a unique solution for the oil variables because the system is square and invertible. In MAYO, the signing system is often rectangular, with $k \cdot o > m$, where k is the number of oil blocks, and o is the oil block size. When A has full row rank, the number of free variables is $k \cdot o - m$, producing $q^{k \cdot o - m}$ possible solutions. This underpins the algebraic vulnerability exploited by fault attacks, as multiple valid oil vectors correspond to the same message hash for a given vinegar vector.

$SampleSolution(A, y, r)$, a randomized subroutine used in MAYO's signing algorithm, uses a randomness vector r to choose a solution from the affine space defined by the linear system. In order to make solving $A(x-r) = y-Ar$ equal to solving $Ax = y$, the process initializes x to r, substitutes y with $y - Ar$, and then uses Gaussian elimination to confirm full rank. Non-pivot coordinates stay equal to the randomness, but pivot coordinates undergo back-substitution. The algorithm is efficient, with Gaussian elimination on an $m \times k * o$ matrix costing $O(m^2 k * o)$ field operations. Distinct choices of r on free variables produce distinct valid solutions.

An important probabilistic aspect is the likelihood that a randomly chosen vinegar vector produces a coefficient matrix of full row rank. For the parameter values recommended in the literature, this probability is high, indicating that restart rates are extremely low, often on the order of $2^{-12} to 2^{-20}$. When full rank is achieved, the expected number of valid solutions for a given vinegar vector is $q^{k \cdot o - m}$, which can be extremely large even for moderate parameters.

Knowing even a single valid oil vector can allow an attacker to reconstruct the whole secret oil space in polynomial time. These attacks exploit linear algebra and the structural properties of the central map: a single preimage constrains the candidate space for trapdoor matrices and can lead to the recovery of a basis for the secret oil space. This makes fault-induced multiple signatures extremely dangerous, as each distinct valid signature may reveal a different oil vector, and even a single oil vector can be sufficient to compromise the private key fully. [4]

3 Overview

3.1 Contributions

Zhong's paper presents several contributions to the analysis of the MAYO signature scheme.

1. Identification of a vulnerability during the signing process of MAYO, specifically when using the deterministic mode of MAYO, where there is no added randomness per-signature.

2. A novel theoretical fault injection attack. Zhong introduces a theoretical attack that targets the generation of valid signatures, showing that an attacker can obtain multiple valid signatures that, when compared, leak information about the private oil space. This allows the use of previous work by Pierre [4] to uncover the entire oil space in polynomial time.

3. Mitigations. Zhong provides two mitigation recommendations: First, do not use deterministic mode. Instead, use the randomized mode, which introduces new randomness for every signature, preventing this attack. Second, if using deterministic mode, minimize the number of free variables (though impossible to remove them entirely, given they are a core part of how MAYO reduces the public key size of UOV)

4. Shows that for any previous schemes, no matter the security parameters, as long as MAYO is used in deterministic mode with some number of free variables, it is vulnerable to this attack.

3.2 Technical Approach

Zhong's paper examines the non-unique signature problem in MAYO. In schemes like classic UOV, a fixed choice of the vinegar variable typically yields a single valid solution for the oil variables. In MAYO, however, the signing algorithm routinely produces many valid signatures for the same message and the same vinegar vector.

The core mathematical step is as follows: for a fixed vinegar vector, if the matrix A has full row rank m, then the linear system has $k \cdot o - m$ free variables. Every assignment of these free variables yields a different valid oil vector solution. In the paper, Lemma 1 proves that the number of free variables is precisely $k \cdot o - m$, and Lemma 2 shows that the number of distinct

valid signatures for the same message-vinegar pair is $q^{k \cdot o - m}$. This non-uniqueness directly enables fault attacks because the signer's deterministic algorithm selects one particular solution using a randomness vector r. If an attacker alters this randomness, the resulting signature changes while the vinegar vector remains fixed.

The attack's methodology focuses on altering the randomness vector during the signing phase. The process sets $x \leftarrow r$ at the beginning of the *SampleSolution* process and uses r to compute both the residual vector $y - Ar$ and the initial values for non-pivot coordinates. Fault injection is deployed either when r is assigned and decoded from the SHAKE stream or inside the *SampleSolution* algorithm, specifically during the steps that initialize x from r or compute Ar. By injecting a fault into any of these operations, the attacker forces the signer to produce a signature corresponding to a different set of free variables while keeping the vinegar vector unchanged.

Once multiple valid signatures for the same vinegar vector are obtained, the attacker can compare them to extract an oil vector. Because the difference between two signatures with the same vinegar values lies entirely in the oil component, subtracting one signature from another yields a vector lying in the secret oil space. The paper notes that recent algebraic work [4] shows that a single valid oil vector is enough to recover the entire secret oil space in polynomial time. Thus, as soon as a fault-induced signature yields even one non-zero oil component, the attacker has enough information to recover the entire trapdoor.

The complete attack proceeds as follows: use a fault injection technique during the signing phase to alter the randomness vector r, receive a valid but distinct signature, compute the difference between this signature and the one generated without faults, extract an oil vector from this difference, apply the previously known polynomial-time key recovery method to reconstruct the entire secret key. Because MAYO almost always produces a full-rank matrix A, and the signing algorithm only needs to restart between 2^{-12} and 2^{-20} of the time, the fault injection succeeds with essentially the same probability as a normal signature generation.

3.3 Results

The core theoretical result is labeled Theorem 1 in the paper. It describes a security game in which an adversary A has access to a signing oracle and can change the values of free variables via fault injection. The probability of A successfully recovering the oil-space is lower-bounded by the probability that a randomly sampled vinegar vector v produces a matrix with full row rank.

As mentioned before in the preliminary knowledge section: If rank(matrix) = m, then the number of free variables is equal to k * o - m. This basically states that the attack only fails if the signing stage fails and MAYO restarts. As mentioned previously, the restart rate is incredibly low, which means this attack will work virtually every time. This is under the heavy assumption that whatever fault injection technique is being used also works every time.

Given that free variables are a core component of MAYO, and $k \cdot o - m > 0$, any security parameters will still cause MAYO to be susceptible to this attack. Zhong calculates the number of free variables for each set of security parameters. None of these security parameters determines the attack's success. Still, the more free variables/valid solutions in a given scheme, the easier it may be for an attacker to determine how to conduct fault injection, as there is a slightly larger attack surface (more outcomes to check).

3.4 Strengths

1. Attack focuses on the core design choice of MAYO (whipping/free variables). As a result, it cannot be patched without completely changing the scheme. This also makes any past and future schemes vulnerable to the attack, regardless of implementation and/or security parameters.

2. Very high rate of success. Unlike many fault injection attacks that require many attempts, certain conditions, or luck in general, this attack will succeed as long as a message is successfully signed. The only chance of failure is if the signing fails, which is incredibly small (around 1 in 2^{-12} to 2^{-20} chance).

3. Efficient and leads to complete key recovery. Truly breaks the signature scheme. Full attack chain (including Pébereau [4] recovery) showcases complete secret key recovery in polynomial time.

4. Full coverage of MAYO. Any implementation of MAYO using deterministic mode is vulnerable to this attack. Any security parameters or versions of MAYO.

3.5 Weaknesses

1. Very strong assumptions about a potential adversary/attacker. To conduct this attack, three key things must be true:

   (a) Attacker has physical access to the device

(b) The attacker has determined when/how to implement the fault injection on a given SoC or CPU. This takes time and requires specialized, potentially expensive equipment, as the precision required for such an attack is likely very high.

(c) Must be using the deterministic mode of MAYO.

2. Purely theoretical. While the paper provides a detailed theoretical overview of an attack, it does not include examples of potential attacks, PoCs, etc. The theoretical nature leaves much of the heavy lifting to future work, where the attack is actually implemented.

3. Limited impact. Only applies to a mode of MAYO that would likely not see much use (deterministic). Any real-world implementations or use of MAYO would likely default to random mode. This limits the full potential of this attack as it may be hard to find valid candidates for it.

4. Relies on Pébereau [4]. This paper is not the full attack chain and relies on previous work for the final key extraction. If there are limitations or potential ways to counter Pébereau's work, this attack becomes useless.

4 Related Work and Follow-up Research

Since Zhong's paper is so recent, there hasn't been any papers specifically following up on her work. However, comparable fault-based vulnerabilities have been found in previous research on this multivariate scheme and published around the same time. This implies that Zhong's results are consistent with a wider pattern.

Jendral and Dubrova's MAYO Key Recovery: Fixing Vinegar Seeds (2024) [5] is one notable paper. The authors of this study apply three first-order single-execution fault injections to an actual MAYO implementation. An ARM Cortex-M4 processor is used for this. To illustrate useful key recovery, Jendral and Dubrova assessed three different fault models against MAYO. The first model modifies the randomness used in the signing process by applying bit-flip faults to the vinegar seed during initialization. The second model modifies intermediate computations in the oil-vector linear system by introducing timing faults during modular multiplication. In order to generate multiple legitimate signatures for the same message, the

third model uses voltage glitches that momentarily corrupt memory or register values. Across these models, the researchers extracted oil vectors from faulty signatures. They ultimately recovered the full secret key, highlighting the practical relevance of the theoretical vulnerabilities identified in Zhong's paper. Their experiments achieve success rates of 36%, 82% and 99% for the three fault models.

Another important work is Single-trace side-channel attacks on MAYO [6], exploiting leaky modular multiplication by the same authors. Rather than using fault injection, this paper shows that side-channel leakage (specifically, power consumption leakage during modular multiplication) can reveal enough information to leak an oil-vector from MAYO. With only a single signature trace and the help of deep-learning-assisted power analysis, they recover a secret key with success probabilities of 99.9% and 91.6%, depending on the attack variant. This considerably expands the attack surface, showing that physical security risks for MAYO are not limited to active fault attacks but also include passive side-channel vulnerabilities. The reliance on leaky modular multiplication underlines the importance of secure finite-field implementations.

More broadly, the community has begun to step back and take a comprehensive view of multivariate schemes under physical attacks. The survey paper SoK: On the Physical Security of UOV-based Signature Schemes [7] offers exactly that. It systematically collects existing side-channel and fault attacks against UOV-based schemes (including MAYO) and maps them to current implementations. The authors argue that because many of these schemes (UOV, MAYO, SNOVA, QR-UOV) share structural design features, vulnerabilities observed in one are often relevant to others. They also discuss how implementation decisions, such as key compression, randomization choices, and fault detection/masking mechanisms, can significantly affect physical security. Moreover, they provide hardened reference implementations (for ARM Cortex-M4) using first-order masking and fault protections, and they benchmark the overhead of these countermeasures. These studies collectively highlight that both active fault injections and passive side-channel leaks pose significant threats to MAYO, emphasizing the need for robust, implementation-level countermeasures.

References

[1] Y. Zhong, Variables for Free: Fault Injection Attack on MAYO via Valid Solutions, 2025.

[2] W. Beullens et al., MAYO, NIST Round 2 Submission, 2024.

[3] A. Kipnis, J. Patarin, and L. Goubin, Unbalanced Oil and Vinegar Signature Schemes, in *EUROCRYPT*, 1999.

[4] P. Pébereau, One Vector to Rule Them All: Key Recovery from One Vector in UOV Schemes, in *PQCrypto*, 2024.

[5] S. Jendral and E. Dubrova, MAYO Key Recovery by Fixing Vinegar Seeds, *Cryptography in Context*, 2024.

[6] S. Jendral and E. Dubrova, Single-trace Side-channel Attacks on MAYO Exploiting Leaky Modular Multiplication, IACR ePrint Archive, 2024.

[7] T. Aulbach, F. Campos, and J. Krämer, SoK: On the Physical Security of UOV-based Signature Schemes, in *PQCrypto*, 2025.

[8] Marin, NIST PQC Security Strength Categories (1–5) Explained, PostQuantum - Quantum Computing, Quantum Security, PQC, April, 2025. https://postquantum.com/post-quantum/nist-pqc-security-categories/