

A Forgery Attack against Frobenius-UOV

Found with OpenAI GPT-5

Augustin Bariant

ANSSI, Paris, France

Abstract. Frobenius-UOV (F-UOV) is a multivariate signature scheme by Macario-Rat over $\mathbb{F}_{p^e}$ whose public equations are quadratic over $\mathbb{F}_p$ but have high degree over $\mathbb{F}_{p^e}$. A message can be forged by solving a six-term univariate equation in $\mathbb{F}_{p^e}$, which can be reduced to a univariate equation containing three monomials with exponents $p^{a_k} + p^{b_k}$ for $k = 0, 1, 2$ and a constant term, where the constants a_k and b_k are fixed by the specification.

We show that the choice of constants a_k, b_k of F-UOV allows one to solve the univariate equation efficiently without knowledge of the secret key, leading to a forgery attack. The attack introduces a variable representing a Frobenius power of x , derives two low-degree bivariate equations, and solves them using a so-called linearized resultant. Under heuristics assumptions on the success probabilities, the average complexity is $\tilde{O}(ep^6)$ field operations. For the 128-, 192-, and 256-bit security instances, our estimates are approximately 2^{45} , 2^{52} , and 2^{53} field operations, respectively. This attack may be mitigated by changing the choice of the exponents a_k and b_k .

1 Introduction

Multivariate signatures are attractive in post-quantum cryptography because they offer short signatures and inexpensive verification. The most studied such multivariate signature scheme is Unbalanced Oil and Vinegar (UOV) [3], in which the public key is a multivariate quadratic map, and where a hidden oil space allows the signer to efficiently find a preimage of any value under that map. The main practical drawback of UOV is the size of its public key, which led several designs to add algebraic structure to the public map to obtain a more compact public key representation. Frobenius-UOV (F-UOV) is one such structured variant of UOV designed by Macario-Rat in 2026 [4]. Its public key is a polynomial map $\text{Pub} : \mathbb{F}_q^5 \rightarrow \mathbb{F}_q$, where $q = p^e$, which is of high degree in $\mathbb{F}_q$ but can be represented as e quadratic equations in $\mathbb{F}_p$.

Contributions. In this paper, we present an efficient forgery attack for the proposed F-UOV parameter sets. Its heuristic average complexity is $\tilde{O}(ep^6)$ operations in $\mathbb{F}_q$, using a resultant-based approach to eliminate a variable in a bivariate system.

2 Background

2.1 Notation

Let p and e be odd primes, and denote $q = p^e$. For a nonzero vector $U \in \mathbb{F}_q^5$, the set $\mathbb{F}_q U = \{xU : x \in \mathbb{F}_q\}$ denotes the line generated by U . For $i \in \mathbb{N}$, $x^{[i]}$ denotes x^{p^i} , and for $i \in \mathbb{Z}^-$, $x^{[i]}$ denotes $x^{p^{i+ke}}$ where k is the smallest integer such that $i + ke \geq 0$. The same notation is applied coordinatewise to vectors and matrices.

2.2 Specifications of F-UOV

Following the F-UOV specification [4], for $a, b \in \{0, \dots, e-1\}$ an (a, b) -Frobenius form is a polynomial in $\mathbb{F}_q[x_1, \dots, x_n]$ whose monomials have shape $x_i^{p^a} x_j^{p^b}$. The public map encoded by the public key is the sum of ℓ Frobenius forms: $\text{Pub} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$, where

$$\text{Pub}(X) = \sum_{k=0}^{\ell-1} \text{Pub}_k(X), \quad \text{Pub}_k(X) = X^{[a_k]} A_k (X^{[b_k]})^\top, \quad A_k \in \mathbb{F}_q^{5 \times 5}, \quad (1)$$

where $X = (x_1, \dots, x_n)$ and $a_k, b_k \in \{0, \dots, e-1\}$ are fixed parameters of an F-UOV instance, called exponent pairs.

The proposed parameters use $\ell = 5$ and

$$(p, e) \in \{(31, 61), (61, 89), (61, 127)\},$$

for the 128-, 192- and 256-bit security levels, respectively. For all proposed F-UOV instances, the exponent pairs are defined as:

$$(a_0, \dots, a_4) = (0, 2, 1, 5, 7) \quad b_k - a_k = D - k \quad D = (e-1)/2.$$

Trapdoor. The trapdoor is the knowledge of a nonzero vector $\Lambda \in \mathbb{F}_q^5$ such that $\text{Pub}_k(\Lambda) = 0$ for all $k = 0, \dots, 4$. Seen as a quadratic map of $\mathbb{F}_p$, the F-UOV public keys are structured UOV instances with $4e$ vinegar variables and e oil variables. The signer uses this trapdoor to invert Pub . Let $H : \{0, 1\}^* \rightarrow \mathbb{F}_q$ be a hash function: a signature on a message $\mathbf{m}$ is a pair $(\Sigma, \tau) \in \mathbb{F}_q^5 \times \{0, 1\}^{256}$, where τ is a salt chosen by the signer and $\text{Pub}(\Sigma) = H(\mathbf{m} \parallel \tau)$. In the analysis below, $h \in \mathbb{F}_q$ denotes the hash value for a fixed message and salt.

Additional notations. In the following, we define $M_k = A_k^{[-a_k]}$, and

$$Q_k(X) = \text{Pub}_k(X)^{[-a_k]} = X M_k (X^{[D-k]})^\top. \quad (2)$$

Then

$$\text{Pub}_k(X) = 0 \iff Q_k(X) = 0.$$

For any $a \in \mathbb{F}_q^5$, define

$$V_a = \{z \in \mathbb{F}_q^5 : a M_3(z^{[D-3]})^\top = z M_3(a^{[D-3]})^\top = 0\},$$

which is a subspace of $\mathbb{F}_q^5$ of dimension at least 3.

3 A new forgery attack against F-UOV

The first step of the attack was mentioned in the F-UOV security analysis [4, Section 4.3], although the algorithmic procedure was not specified there. It finds a publicly computable $\mathbb{F}_q$ -line on which two Frobenius blocks are zero. On this line, forging comes down to solving a four-term univariate equation in $\mathbb{F}_q$, which we show can be done with resultants.

3.1 Finding a common zero of two Frobenius blocks

Summary. We first find a nonzero $u \in \mathbb{F}_q^5$ such that $Q_3(u) = 0$, then $v \in V_u \setminus \langle u \rangle$ such that $Q_3(v) = 0$. $W = \langle u, v \rangle$ is such that for all $X, Y \in W$, $XM_3(Y^{[D-3]})^\top = 0$. We then find a nonzero $U \in W$ such that $Q_4(U) = 0$, thus $Q_3(U) = Q_4(U) = 0$.

Finding a nonzero u such that $Q_3(u) = 0$. Proceed as follows. Let $a \in \mathbb{F}_q^5$ and $b \in V_a$ be two random independent vectors. Such a vector u can be found on the line $a + xb$ for $x \in \mathbb{F}_q$ with probability $\approx 1/2$. Since $b \in V_a$,

$$Q_3(a + xb) = Q_3(a) + x^{p^{D-3}+1}Q_3(b).$$

For each proposed pair (p, e) of the F-UOV specifications, $\gcd(p^{D-3} + 1, p^e - 1) = 2$. Let $r \in \mathbb{Z}$ such that $r(p^{D-3} + 1) \equiv 2 \pmod{p^e - 1}$. If $Q_3(b) \neq 0$, the equation $Q_3(a + xb) = 0$ becomes $x^2 = (-Q_3(a)/Q_3(b))^r$, which has a solution in $\mathbb{F}_q$ with probability $\approx 1/2$. If $Q_3(b) = 0$, take $u = b$. We heuristically expect a constant average number of iterations with random a, b to recover a u such that $Q_3(u) = 0$.

Finding a nonzero $v \in V_u \setminus \langle u \rangle$ such that $Q_3(v) = 0$. Since $Q_3(u) = 0$, we have $u \in V_u$. Choose $a, b \in V_u$ such that u, a, b are linearly independent. We aim at finding v on the line $a + xb$ for $x \in \mathbb{F}_q$. We have:

$$Q_3(a + xb) = \alpha + \beta x + \gamma x^{[D-3]} + \delta x^{p^{D-3}+1},$$

with $\alpha, \beta, \gamma, \delta \in \mathbb{F}_q$ derived from a, b . The equation $Q_3(a + xb) = 0$ becomes:

$$(\gamma + \delta x)x^{[D-3]} + (\alpha + \beta x) = 0. \quad (3)$$

If $\gamma = \delta = 0$, one can solve the remaining linear equation. Otherwise, we test whether the potential solution of $\gamma + \delta x = 0$ satisfies the equation. If not, we assume $\gamma + \delta x \neq 0$ and:

$$x^{[D-3]} = -\frac{\alpha + \beta x}{\gamma + \delta x}. \quad (4)$$

One can remark that the Equation (4) can be raised at the power p^{D-3} , the left hand-side becomes $x^{[2(D-3)]}$ and the right hand-side stays homographic

(after substitution of $x^{[D-3]}$ with Equation (4)). Iterating this process e times, the left-hand side becomes $x^{[e(D-3)]} = x$ and the right-hand side is homographic and can be efficiently computed. Thus, any solution to Equation (4) is a solution to a quadratic equation in $\mathbb{F}_q$. We thus compute the candidate solutions of the quadratic equation and retain them if they are solutions to Equation (4). If no solution is found, restart the process with a new vector u . Heuristically, we expect Equation (4) to have a solution with a constant probability.

Let us define $W = \langle u, v \rangle$, such that for all $X, Y \in W$, $XM_3(Y^{[D-3]})^\top = 0$.

Finding a nonzero $U \in W$ such that $Q_4(U) = 0$. If $Q_4(v) = 0$, take $U = v$. Else, we find a U on the line $u + vx$ for $x \in \mathbb{F}_q$. The equation $Q_4(u + vx) = 0$ has the same form as Equation (3), and can be solved similarly with probability $\approx 1/2$. If no solution is found, restart with another space W . A solution gives $U \in W$ with $Q_3(U) = Q_4(U) = 0$.

Complexity and experimental verification. Solving quadratic equations and computing the coefficients of the homographic function cost $O(e \log p)$ operations in $\mathbb{F}_q$ per attempt. For each proposed F-UOV parameter set, we implemented this step in Magma, which recovers such a U in fewer than 14 attempts and a few seconds in average.

3.2 Elimination on the restricted line

Fix a nonzero vector U found as above, with $Q_3(U) = Q_4(U) = 0$. The forgery consists of finding $\Sigma \in \mathbb{F}_q^5$ such that $\text{Pub}(\Sigma) = h$ for a target $h \in \mathbb{F}_q$, which we tackle by restricting Σ to a line $\mathbb{F}_q U$.

Bivariate modeling. Since $Q_3(U) = Q_4(U) = 0$, the equation $\text{Pub}(xU) = h$ becomes

$$c_0 x^{1+p^D} + c_1 x^{p^2+p^{D+1}} + c_2 x^{p+p^{D-1}} = h, \quad (5)$$

with $c_k = \text{Pub}_k(U)$ for $k = 0, 1, 2$. Let us define $y = x^{[D-1]}$, or equivalently $y^{[D+2]} = x$ since $D = (e-1)/2$. Equation (5) and its p^{D+2} -th power give respectively:

$$F_h(x, y) = c_0 x y^{[1]} + c_1 x^{[2]} y^{[2]} + c_2 x^{[1]} y - h = 0, \quad (6)$$

$$G_h(x, y) = c_0^{[D+2]} x^{[1]} y^{[3]} + c_1^{[D+2]} x^{[2]} y^{[5]} + c_2^{[D+2]} x y^{[4]} - h^{[D+2]} = 0. \quad (7)$$

Any solution to $\text{Pub}(xU) = h$ yields a solution to $F_h(x, y) = G_h(x, y) = 0$. Therefore, if a solution to $\text{Pub}(xU) = h$ exists (which is expected with constant probability), recovering all the solutions in $\mathbb{F}_q$ of $F_h(x, y) = G_h(x, y) = 0$ and checking the x -coordinate is a root of Equation (5) allows one to recover it and to produce a valid forgery. Solving $F_h(x, y) = G_h(x, y) = 0$ can be done with a resultant approach.

Bivariate solving via linearized resultants. Instead of using the standard bivariate resultant between F_h and G_h , whose computation cost induces an overhead compared to its degree, we define a *linearized* resultant which is the determinant of a Sylvester-like matrix of smaller dimension. More precisely, write:

$$\begin{aligned} F_h(x, y) &= f_2(y)x^{[2]} + f_1(y)x^{[1]} + f_0(y)x + f_c(y), \\ G_h(x, y) &= g_2(y)x^{[2]} + g_1(y)x^{[1]} + g_0(y)x + g_c(y). \end{aligned}$$

One may remark that $F_h(x, y)^{[i]}$ and $G_h(x, y)^{[i]}$ can be expressed in the basis $(1, x^{[i]}, x^{[i+1]}, x^{[i+2]})$. Thus, the polynomials $F_h(x, y)^{[i]}$ and $G_h(x, y)^{[i]}$ for $i = 0, 1, 2$ can all be expressed in the basis $(1, x, x^{[1]}, x^{[2]}, x^{[3]}, x^{[4]})$, and one can form a so-called p -Sylvester matrix:

$$p\text{-Syl}_x = \begin{pmatrix} f_c & f_0 & f_1 & f_2 & 0 & 0 \\ f_c^{[1]} & 0 & f_0^{[1]} & f_1^{[1]} & f_2^{[1]} & 0 \\ f_c^{[2]} & 0 & 0 & f_0^{[2]} & f_1^{[2]} & f_2^{[2]} \\ g_c & g_0 & g_1 & g_2 & 0 & 0 \\ g_c^{[1]} & 0 & g_0^{[1]} & g_1^{[1]} & g_2^{[1]} & 0 \\ g_c^{[2]} & 0 & 0 & g_0^{[2]} & g_1^{[2]} & g_2^{[2]} \end{pmatrix}$$

We call $p\text{-Res}_x(F_h, G_h) = \det(p\text{-Syl}_x) \in \mathbb{F}_q[y]$. For $y_0 \in \mathbb{F}_q$, if $F_h(x, y_0)$ and $G_h(x, y_0)$ have a common root ν , then the vector $(1, \nu, \nu^{[1]}, \nu^{[2]}, \nu^{[3]}, \nu^{[4]})$ belongs to the kernel of $p\text{-Syl}_x(F_h, G_h)$, so $p\text{-Res}_x(F_h, G_h)(y_0) = 0$, which implies that y_0 is a root of $p\text{-Res}_x(F_h, G_h)$. Therefore, the y -coordinate of any solution to $F_h(x, y) = G_h(x, y) = 0$ is a solution to $p\text{-Res}_x(F_h, G_h)(y) = 0$.

One can inspect the p -Sylvester matrix and see that the largest and smallest degrees of $p\text{-Res}_x(F_h, G_h)$ are respectively $m_+ = p^7 + p^6 + p^5 + p^2 + p$ and $m_- = 2p^4 + 2p^3 + p$, as long as h, c_0, c_1 are nonzero. If h, c_0 , or c_1 is zero, the attacker resamples the salt or U .

Every nonzero determinant term contains one entry from the first column and five entries whose degrees are powers of p . Its degree is therefore congruent to 5 modulo $p - 1$. It follows that

$$p\text{-Res}_x(F_h, G_h)(y) = y^{m_-} S_h(y^{p-1}), \quad (8)$$

where

$$d := \deg(S_h) = (m_+ - m_-)/(p - 1) = p^6 + 2p^5 + 3p^4 + p^3 - p^2.$$

The determinant expansion contains at most 56 nonzero terms. Thus, S_h can be constructed directly with a constant number of coefficient operations, once the required Frobenius conjugates have been computed.

Let r denote the number of roots of S_h in $\mathbb{F}_q$. They are obtained by computing $\gcd(S_h, Z^q - Z)$ and factoring the resulting product of r linear factors. With fast

Level	(p, e)	C
I	(31, 61)	2^{45}
III	(61, 89)	2^{52}
V	(61, 127)	2^{53}

Table 1. Complexity of our attack for the different F-UOV parameter sets.

polynomial multiplication, modular exponentiation, half-gcd, and equal-degree factorization, this requires

$$O(d \log d \log \log d (\log q + \log d) + r(\log r)^2 \log \log r (\log q + \log r))$$

operations in $\mathbb{F}_q$ [2, Corollary 14.16]. Since $r \leq d$ and $\log q = e \log p$, this is $\tilde{O}(ed) = \tilde{O}(ep^6)$. For each root z , the condition $z^{(q-1)/(p-1)} = 1$ determines whether $z = y^{p-1}$ for some $y \in \mathbb{F}_q^*$. One could then compute all nonzero solutions of $y^{p-1} = z$ (which all differ by a factor in $\mathbb{F}_p^*$), and test whether $y^{[D+2]}U$ leads to a valid solution of Equation (5). However this adds a term $(p-1)r = O(p^7)$ in the complexity. Instead, one can compute a solution y_1 of $y^{p-1} = z$ and compute $x_1 = y_1^{[D+2]}$. The other solutions of this equation are of the form $y_\lambda = \lambda y_1$ (with a corresponding $x_\lambda = \lambda x_1$) with $\lambda \in \mathbb{F}_p$. Since $\text{Pub}(x_\lambda U) = \lambda^2 \text{Pub}(x_1 U)$, one can recover λ by solving a quadratic equation $\lambda^2 \text{Pub}(x_1 U) = h$ in $\mathbb{F}_p$. With this procedure, any solution to Equation (5) in $\mathbb{F}_q$ is recovered. Heuristically, we expect to find a solution with constant probability, which is consistent with our reduced experiments.

3.3 Complexity and experiments

The dominant step of the attack is root extraction for the degree- d polynomial S_h . Its complexity is

$$O(d \log d \log \log d (\log d + e \log p)) = \tilde{O}(ep^6)$$

operations in $\mathbb{F}_q$ [2, Corollary 14.16]. Testing all roots and recovering the corresponding candidates is also $\tilde{O}(ed)$, and is therefore covered by the same bound. For the concrete estimates, we evaluate the explicit expression

$$C = d \log_2 d \log_2 \log_2 d (\log_2 d + e \log_2 p).$$

Here C estimates the number of $\mathbb{F}_q$ operations used by root extraction when r is bounded and the hidden constant in the asymptotic bound is set to one. Table 1 reports C , rounded to a power of two. This estimate is not a prediction of CPU time and does not include the lower-order determinant construction and candidate-recovery steps.

Table 1 summarizes the complexity of our attack against the proposed parameter sets of F-UOV.

(p, e)	d	Line p-resultant		Roots	Full attack
(3, 61)	1 476	< 1	< 1	6	6
(5, 61)	23 850	< 1	5	873	878

Table 2. Complete reduced-characteristic attacks. Timings are in seconds.

Experimental validation The attack is implemented in Magma with $e = 61$, $p = 3$ and $p = 5$. Each target $h \neq 0$ is sampled independently. If no candidate verifies on a line U which cancels two public blocks, the experiment samples a fresh target h , and retries the bivariate solving on the same line U . For both characteristics, the second target produces a verified forgery. The costs of the attack are presented in Table 2, together with the public computation of U . CPU times were measured with Magma V2.29-8 on an AMD EPYC 7543 machine.

3.4 Countermeasures

The attack relies on the structure of the specified exponent pairs (a_k, b_k) : the first three public exponents are close to $(0, (e - 1)/2)$, making this bivariate representation particularly effective. Changing the pairs (a_k, b_k) , by choosing different regions of $[0, \dots, e - 1]$ for the values of a_k may mitigate this bivariate attack. It is possible, however, to derive a similar modeling with more than two variables, therefore a careful analysis still needs to be performed.

4 Conclusion

For the exponent pairs specified by F-UOV, we found an efficient forgery attack with heuristic average complexity $\tilde{O}(ep^6)$ operations in $\mathbb{F}_q$. The attack was verified experimentally by complete reduced-characteristic attacks with $p = 3$ and $p = 5$.

Acknowledgements. The author would like to thank Gilles Macario-Rat for the insightful discussions around the linearized resultant and the possible countermeasures.

On the use of LLM. This attack was found by OpenAI GPT-5 after a few hours of work. A few more hours were necessary to find some optimizations, notably the linearized resultant technique. The author tried to have the LLM write the entire report, but the presentation of the paper was unsatisfactory (use of unusual technical words, blog-post-like formulation of the ideas, struggle to explain the ideas proportionally to their difficulty...). Therefore, the author re-wrote almost the entire report. The small-scale implementation of the attack was performed by the LLM and verified by the author.

Prompt (in French). Tu es un chercheur en cryptographie, spécialisé dans la cryptographie multivariée. Tu t'intéresses à une nouvelle construction, Frobenius UOV, qui donne l'impression d'avoir de très bonnes caractéristiques en termes de taille de clé publique, signature et clé privée. La spécification est disponible en pdf dans 2026-968.pdf.

Ton objectif est de trouver une attaque polynomiale sur ce cryptosystème, qui casse donc les paramètres de sécurité du schéma. Aussi, il serait intéressant d'avoir une attaque qui améliore asymptotiquement la complexité des attaques par rapport aux concepteurs de F-UOV.

Tu ne t'arrêtes pas avant d'avoir trouvé une attaque. Modifie ton /goal en conséquence. Tu ne sors pas de ce dossier. Tu n'accèdes pas à internet.

Si tu trouves une attaque, produis un rapport en latex résumant l'attaque, que tu compiles. Essaie d'être le plus clair possible.

References

1. Cantor, D.G., Zassenhaus, H.: A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation* **36**(154), 587–592 (1981). <https://doi.org/10.1090/S0025-5718-1981-0603967-0>
2. von zur Gathen, J., Gerhard, J.: *Modern Computer Algebra* (3. ed.). Cambridge University Press (2013)
3. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: *Advances in Cryptology – EUROCRYPT '99*. Lecture Notes in Computer Science, vol. 1592, pp. 206–222. Springer (1999). https://doi.org/10.1007/3-540-48910-X_15
4. Macario-Rat, G.: Frobenius-UOV: A very efficient multivariate public key signature scheme. *Cryptology ePrint Archive*, Report 2026/968 (2026), <https://eprint.iacr.org/2026/968>