

Improved Cryptanalysis of HFEv- via Projection

Jintai Ding¹, Ray Perlner², Albrecht Petzoldt², and Daniel Smith-Tone^{2,3}

¹Department of Mathematical Sciences, University of Cincinnati,
Cincinnati, Ohio, USA

²National Institute of Standards and Technology,
Gaithersburg, Maryland, USA

³Department of Mathematics, University of Louisville,
Louisville, Kentucky, USA

jintai.ding@uc.edu, ray.perlner@nist.gov,
albrecht.petzoldt@gmail.com, daniel.smith@nist.gov✉

Abstract. The Hidden Field Equations with vinegar and minus modifiers (HFEv-) signature scheme is one of the most studied multivariate schemes and one of the major candidates for the upcoming standardization of post-quantum digital signature schemes. In this paper, we propose three new attack strategies against HFEv-, each of them using the idea of projection. Especially our third attack is very effective and is, for some parameter sets, the most efficient known attack against HFEv-. Furthermore, our attack requires much less memory than direct and rank attacks. By our work, we therefore give new insights in the security of the HFEv- signature scheme and restrictions for the parameter choice of a possible future standardized HFEv- instance.

Key words: Multivariate Cryptography, HFEv-, MinRank, Gröbner Basis, Projection

1 Introduction

Multivariate cryptography is one of the main candidates for establishing cryptosystems which resist attacks with quantum computers (so called post-quantum cryptosystems). Especially in the area of digital signatures, there exists a large number of practical multivariate schemes such as Unbalanced Oil and Vinegar (UOV) [1] and Rainbow [2].

Another well known multivariate signature scheme is the HFEv- signature scheme, which was first proposed by Patarin, Courtois and Goubin in [3]. Most notably about this scheme are its very short signatures, which are currently the shortest signatures of all existing schemes (both classical and post-quantum).

In this paper we propose three new attacks against the HFEv- signature scheme, each of them using the idea of projection. This means that each of our attacks reduces the number of variables in the system by guessing, either before or after the attack itself.

The most interesting results hereby are provided by a distinguishing based attack, which is related to the hybrid approach of the direct attack [4]. The goal of our attack is to remove the vinegar modifier. This allows the attacker to follow up with any key recovery or signature forgery attack applicable to an HFE-instance with the same degree bound and the same number of removed equations as the original HFEv- instance. The attack is very effective and outperforms, for selected parameter sets, all other attacks against HFEv-. Furthermore, the memory requirements of our attack are far less than those of direct and MinRank attacks.

The rest of the paper is organized as follows. In Section 2, we give a short overview of multivariate cryptography and introduce the HFEv- cryptosystem, while Section 3 reviews the previous cryptanalysis of this scheme. Section 4 describes our first two attacks, which combine the MinRank attack with the idea of projection. In Section 5, we present then our distinguishing based attack, whose complexity is analyzed in Section 6. Finally, Section 7 discusses ideas for future work.

2 Hidden Field Equations

2.1 Multivariate cryptography

The basic objects of multivariate cryptography are systems of multivariate quadratic polynomials over a finite field \mathbb{F} . The security of multivariate schemes is based on the *MQ Problem* of solving such a system. The MQ Problem is proven to be NP-Hard even for quadratic polynomials over the field $\text{GF}(2)$ [5] and believed to be hard on average (both for classical and quantum computers).

To build a multivariate public key cryptosystem (MPKC), one starts with an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \rightarrow \mathbb{F}^m$ (*central map*). To hide the structure of \mathcal{F} in the public key, we compose it with two invertible affine (or linear) maps $\mathcal{T} : \mathbb{F}^m \rightarrow \mathbb{F}^m$ and $\mathcal{U} : \mathbb{F}^n \rightarrow \mathbb{F}^n$. The *public key* of the scheme is therefore given by $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{U} : \mathbb{F}^n \rightarrow \mathbb{F}^m$. The relation between the easily invertible central map \mathcal{F} and the public key \mathcal{P} is referred to as a morphism of polynomials.

The *private key* consists of the three maps \mathcal{T} , \mathcal{F} and \mathcal{U} and therefore allows to invert the public key. To generate a signature for a document (hash value) $\mathbf{h} \in \mathbb{F}^m$, one computes recursively $\mathbf{x} = \mathcal{T}^{-1}(\mathbf{h}) \in \mathbb{F}^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}^n$ and $\mathbf{z} = \mathcal{U}^{-1}(\mathbf{y}) \in \mathbb{F}^n$. To check the authenticity of a signature $\mathbf{z} \in \mathbb{F}^n$, one simply computes $\mathbf{h}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$. If the result is equal to \mathbf{h} , the signature is accepted, otherwise rejected. This process is illustrated in Figure 1.

2.2 HFE Variants

The HFE encryption scheme was proposed by J. Patarin in [6]. The scheme belongs to the *BigField* family of multivariate schemes, which means that it uses

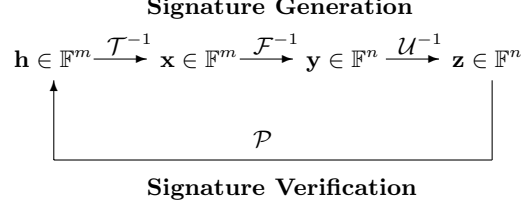


Fig. 1. Signature Generation and Verification for Multivariate Signature Schemes

a degree n extension field \mathbb{E} of \mathbb{F} as well as an isomorphism $\phi : \mathbb{F}^n \rightarrow \mathbb{E}$. The central map is a univariate polynomial map over \mathbb{E} of the form

$$\mathcal{F}(X) = \sum_{0 \leq i, j}^{q^i + q^j \leq D} \alpha_{ij} X^{q^i + q^j} + \sum_{i=0}^{q^i \leq D} \left(\beta_i X^{q^i} + \gamma \right).$$

Due to the special structure of \mathcal{F} , the map $\bar{\mathcal{F}} = \phi^{-1} \circ \mathcal{F} \circ \phi$ is a quadratic map over the vector space \mathbb{F}^n . In order to hide the structure of \mathcal{F} in the public key, $\bar{\mathcal{F}}$ is composed with two affine maps \mathcal{T} and \mathcal{U} , i.e. $\mathcal{P} = \mathcal{T} \circ \bar{\mathcal{F}} \circ \mathcal{U}$.

After the basic scheme was broken by direct [7] and rank attacks [8], several versions of HFE for digital signatures have been proposed. Basically, these schemes use two different techniques: the minus and the vinegar modification. For the HFEV- signature scheme [3], the central map \mathcal{F} has the form

$$\mathcal{F}(X, \bar{x}_V) = \sum_{0 \leq i, j}^{q^i + q^j \leq D} \alpha_{ij} X^{q^i + q^j} + \sum_{i=0}^{q^i \leq D} \left(\beta_i(x_{n+1}, \dots, x_{n+v}) X^{q^i} + \gamma(x_{n+1}, \dots, x_{n+v}) \right),$$

where β_i and γ are linear and quadratic maps in the vinegar variables $\bar{x}_V = (x_{n+1}, \dots, x_{n+v})$ respectively. Defining $\psi : \mathbb{F}^{n+v} \rightarrow \mathbb{E} \times \mathbb{F}^v$ by $\psi = \phi \times id_v$, the public key has the form

$$\mathcal{P} = \mathcal{T} \circ \phi^{-1} \circ \mathcal{F} \circ \psi \circ \mathcal{U} : \mathbb{F}^{n+v} \rightarrow \mathbb{F}^{n-a}$$

with two affine maps $\mathcal{T} : \mathbb{F}^n \rightarrow \mathbb{F}^{n-a}$ and $\mathcal{U} : \mathbb{F}^{n+v} \rightarrow \mathbb{F}^{n+v}$, and is a multivariate quadratic map with coefficients and variables over \mathbb{F} .

Signature Generation: To generate a signature \mathbf{z} for a document d , one uses a hash function $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{F}^{n-a}$ to compute a hash value $\mathbf{h} = \mathcal{H}(d) \in \mathbb{F}^{n-a}$ and performs the following four steps

1. Compute a preimage $\mathbf{x} \in \mathbb{F}^n$ of \mathbf{h} under the affine map \mathcal{T} and set $X = \phi(\mathbf{x}) \in \mathbb{E}$.
2. Choose random values for the vinegar variables x_{n+1}, \dots, x_{n+v} and substitute them into the central map to obtain the parametrized map \mathcal{F}_V .
3. Solve the univariate polynomial equation $\mathcal{F}_V(Y) = X$ over the extension field \mathbb{E} by Berlekamp's algorithm.
4. Compute the signature $\mathbf{z} = \mathcal{U}^{-1}(\phi^{-1}(Y) || x_{n+1} || \dots || x_{n+v}) \in \mathbb{F}^{n+v}$.

Signature Verification: To check the authenticity of a signature $\mathbf{z} \in \mathbb{F}^{n+v}$, the verifier computes $\mathbf{h} = \mathcal{H}(d)$ and $\mathbf{h}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{h}' = \mathbf{h}$ holds, the signature is accepted, otherwise rejected.

3 Previous Cryptanalysis

3.1 Direct Algebraic Attack

The direct algebraic attack is the most straightforward way to attack a multivariate cryptosystem such as HFEv-. In this attack, one considers the public equation $\mathcal{P}(\mathbf{z}) = \mathbf{h}$ as an instance of the MQ-Problem. In the case of HFEv-, this public system is slightly underdetermined. In order to make the solution space zero dimensional, one therefore fixes $a + v$ variables in order to get a determined system before applying an algorithm like XL [9] or a Gröbner basis method such as F_4 or F_5 [10, 11]. In some cases one gets better results by guessing additional variables, even if this requires running the Gröbner basis algorithm several times (hybrid approach [4]).

The complexity of a direct attack using the hybrid approach against a system of m quadratic equations in n variables can be estimated as

$$Comp_{direct} = \min_k q^k \cdot 3 \cdot \binom{n-k+d_{\text{reg}}}{d_{\text{reg}}}^2 \cdot \binom{n-k}{2},$$

where d_{reg} is the so called *degree of regularity* of the multivariate system. Note that this formula gives only a rough estimate and lower bound of the complexity of a direct attack, since it assumes that the linear systems appearing during the attack are very sparse systems. It is not clear if this assumption holds and if the used Wiedemann algorithm can work with the assumed complexity.

Experiments have shown that the public systems of HFE and its variants can be solved significantly faster than random systems [7, 12]. This phenomenon was studied by Ding et al. in a series of papers [13–15]. In [15] it was shown that the degree of regularity of solving an HFEv- system is upper bounded by

$$d_{\text{reg, HFEv-}} \leq \begin{cases} \frac{(q-1) \cdot (r+a+v-1)}{2} + 2 & q \text{ even and } r+a \text{ odd} \\ \frac{(q-1) \cdot (r+a+v)}{2} + 2 & \text{otherwise} \end{cases}. \quad (1)$$

3.2 MinRank

The historically most effective attack on the HFE family of cryptosystems is the MinRank attack which exploits the algebraic consequence of a low degree bound D . This low degree bound leads to the fact that the central map has a low Q-rank.

Definition 1 *The Q-rank of a multivariate quadratic map $\mathcal{F} : \mathbb{F}^n \rightarrow \mathbb{F}^n$ over the finite field \mathbb{F} with q elements is the rank of the quadratic form \mathcal{Q} on $\mathbb{E}[X_1, \dots, X_n]$ defined by $\mathcal{Q}(X_1, \dots, X_n) = \phi \circ \mathcal{F} \circ \phi^{-1}(X)$, under the identification $X_1 = X, X_2 = X^q, \dots, X_n = X^{q^{n-1}}$.*

Q-rank is invariant under one-sided isomorphisms of polynomials of the form $\mathcal{G} = \mathcal{I} \circ \mathcal{F} \circ \mathcal{U}$, where \mathcal{I} is the identity transformation. Q-rank is not, however, invariant under isomorphisms of polynomials in general. The min-Q-Rank of a quadratic map \mathcal{F} is the minimum Q-rank of any quadratic map in the isomorphism class of \mathcal{F} . This quantity is invariant under isomorphisms of polynomials, and is the relevant quantity for cryptanalysis. For historical reasons, language is often abused and the term Q-rank is used in place of min-Q-rank.

As an example, consider an odd characteristic instance of HFE. We may write the homogeneous quadratic part of F as

$$\begin{pmatrix} X & X^q & \dots & X^{q^{n-1}} \end{pmatrix} \begin{bmatrix} \alpha_{1,1} & \alpha'_{1,2} & \dots & \alpha'_{1,d} & 0 & \dots & 0 \\ \alpha'_{1,2} & \alpha_{2,2} & \dots & \alpha'_{2,d} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \alpha'_{1,d} & \alpha'_{2,d} & \dots & \alpha_{d,d} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \begin{pmatrix} X \\ X^q \\ \vdots \\ X^{q^{n-1}} \end{pmatrix},$$

where $\alpha'_{i,j} = \frac{1}{2}\alpha_{i,j}$ and $d = \lceil \log_q(D) \rceil$. Clearly, this quadratic form over the ring $\mathbb{E}[X_1, \dots, X_n]$ has rank d , and thus the HFE central map has Q-rank d .

The first iteration of the MinRank attack in the *BigField* setting is the Kipnis-Shamir (KS) attack of [8]. Via polynomial interpolation, the public key can be expressed as a quadratic polynomial \mathcal{G} over the degree n extension field \mathbb{E} . By construction there is an \mathbb{F} -linear map \mathcal{T}^{-1} such that $\mathcal{T}^{-1} \circ \mathcal{G}$ has rank d , thus there is a rank d matrix that is an \mathbb{E} -linear combination of the Frobenius powers of \mathcal{G} . This turns recovery of the transformation \mathcal{T} into the solution of a MinRank problem over \mathbb{E} .

A significant improvement to this method for HFE is the key recovery attack of Bettale et al. [16]. The first significant observation made was that an \mathbb{E} -linear combination of the *public* polynomials has low rank as a quadratic form over \mathbb{E} . By constructing a formal linear combination of the public polynomials with variable coefficients, one can collect the polynomials representing $(d+1) \times (d+1)$ minors of this linear combination, which must be zero by the Q-rank bound. The advantage this technique offers is that the coefficients of the polynomial are in \mathbb{F} ; thus, the Gröbner basis calculation can be performed over \mathbb{F} , while the variety is computed over \mathbb{E} . This *minors modeling* method is significantly more efficient than the KS-attack when the number of equations is similar to the number of variables. (In contrast, for schemes such as Zhuang-zi Hidden Field Equations (ZHFE), see [17], it seems that the KS modeling is more efficient, probably due to the large number of variables in the Gröbner basis calculation, see [18].) To make the ideal zero-dimensional, we fix one variable; thus, the complexity of the KS-attack with minors modeling is asymptotically $\mathcal{O}(n^{\lceil \log_q(D) \rceil \omega})$, where $2 \leq \omega \leq 3$ is the linear algebra constant.

The MinRank approach can also be effective in attacking HFE-. The key observation in [19] is that not only does the removal of an equation increase the

Q-rank by merely one, there is also a basis in which it increases the degree only by a factor of q . Thus HFE- schemes with large base fields are vulnerable to the minors modeling method of [16], even when multiple equations are removed. The complexity of the KS-attack with minors modeling for HFE- is asymptotically $\mathcal{O}(n^{\lceil \log_q(D) \rceil + a\omega})$, where a is the number of equations removed and $2 < \omega \leq 3$ is the linear algebra constant.

4 Variants of MinRank with Projection

As first explicitly noted in [15], the Q-rank of the central map is increased by v with the introduction of v vinegar variables and therefore the min-Q-rank of HFEv- is $\lceil \log_q(D) \rceil + a + v$. We now discuss techniques for turning this observation into a key recovery attack. From this point on, let r denote $\lceil \log_q(D) \rceil$, that is, the Q-rank of the HFE component of the central map.

4.1 MinRank then Projection

The simplest way to attempt an attack utilizing the low Q-rank of the central map of HFEv- is to directly apply a MinRank attack and then try to discover the vinegar subspace by considering the solution as a quadratic form. To this end, consider the surjective \mathbb{E} -algebra representation $\Phi : \mathbb{E} \rightarrow \mathbb{A}$ defined by $\Phi(X) = (X, X^q, \dots, X^{q^{n-1}})$. We may map directly from an n -dimensional vector space over \mathbb{F} to \mathbb{A} via right multiplication by the matrix

$$\mathbf{M}_n = \begin{bmatrix} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \theta & \theta^q & \cdots & \theta^{q^{n-1}} \\ \theta^2 & \theta^{2q} & \cdots & \theta^{2q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \theta^{n-1} & \theta^{(n-1)q} & \cdots & \theta^{(n-1)q^{n-1}} \end{pmatrix} \end{bmatrix},$$

with the choice of a primitive element $\theta \in \mathbb{E}$ (i.e. $\mathbb{E} = \mathbb{F}(\theta)$). Right multiplication by \mathbf{M}_n corresponds to the linear map $\Phi \circ \phi$, where the choice of isomorphism ϕ is determined by the choice of primitive element θ .

We may incorporate the vinegar variables into the picture by simply appending them to \mathbb{A} . Specifically, define the map $\widetilde{\mathbf{M}}_n : \mathbb{F}^{n+v} \rightarrow \mathbb{A} \times \mathbb{F}^v$ by right multiplication by the matrix

$$\widetilde{\mathbf{M}}_n = \begin{bmatrix} \mathbf{M}_n & \mathbf{0}_{n \times v} \\ \mathbf{0}_{v \times n} & I_v \end{bmatrix},$$

where I_v is the identity matrix. We may then represent any HFEv- map as a single $(n+v) \times (n+v)$ matrix with coefficients in \mathbb{E} . Note specifically that any function bilinear with respect to the vinegar variable x_n and the HFE variables x_0, \dots, x_{n-1} can be encoded in row and/or column n of the quadratic form

$$\mathbf{xQx}^\top = \mathbf{x}\widetilde{\mathbf{M}}_n \mathbf{R} \widetilde{\mathbf{M}}_n^\top \mathbf{x}^\top,$$

where $\mathbf{R} \in \mathcal{M}_{(n+v) \times (n+v)}(\mathbb{E})$.

Let \mathbf{F} be defined by $\mathbf{x}\widetilde{\mathbf{M}}_n\mathbf{F}\widetilde{\mathbf{M}}_n^\top\mathbf{x}^\top = \mathcal{F}(x)$ where \mathcal{F} is the central map of HFEV-. We will say that \mathbf{F} is the matrix representation of \mathcal{F} over $\mathbb{A} \times \mathbb{F}^v$. Let \mathbf{F}^{*i} be the matrix representation of the i th Frobenius power of \mathcal{F} over $\mathbb{A} \times \mathbb{F}_v$. Then we have, for example the following shape for \mathbf{F}^{*0} :

$$\begin{bmatrix} \left(\begin{array}{ccccccc} \alpha_{0,0} & \cdots & \alpha_{0,d-1} & 0 \cdots 0 & \beta_{0,n} & \cdots & \beta_{0,n+v-1} \\ \vdots & \ddots & \vdots & \vdots \cdots \vdots & \vdots & \ddots & \vdots \\ \alpha_{0,d-1} & \cdots & \alpha_{d-1,d-1} & 0 \cdots 0 & \beta_{d-1,n} & \cdots & \beta_{d-1,n+v-1} \\ 0 & \cdots & 0 & 0 \cdots 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \cdots \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \cdots 0 & 0 & \cdots & 0 \\ \beta_{0,n} & \cdots & \beta_{d-1,n} & 0 \cdots 0 & \beta_{n,n} & \cdots & \beta_{n,n+v-1} \\ \vdots & \ddots & \vdots & \vdots \cdots \vdots & \vdots & \ddots & \vdots \\ \beta_{0,n+v-1} & \cdots & \beta_{d-1,n+v-1} & 0 \cdots 0 & \beta_{n,n+v-1} & \cdots & \beta_{n+v-1,n+v-1} \end{array} \right) \end{bmatrix}.$$

Here we see that $\text{rank}(\mathbf{F}^{*0}) = r + v$. The structure of \mathbf{F}^{*1} is similar with the upper left HFE block consisting of $\alpha_{i,j}$ shifted down and to the right and raised to the power of q , and the symmetric blocks of mixing monomials shifted down and to the right with a more complicated function applied to the $\beta_{i,j}$ coefficients to respect the Frobenius map.

Now let \mathbf{U} , \mathbf{T} and \mathbf{P}_i be the matrix representations of the affine isomorphisms \mathcal{U} and \mathcal{T} and the public quadratic forms \mathcal{P}_i , respectively. Then we derive the relation

$$(\mathbf{P}_1, \dots, \mathbf{P}_n)\mathbf{T}^{-1}\mathbf{M}_n = (\mathbf{U}\widetilde{\mathbf{M}}_n\mathbf{F}^{*0}\widetilde{\mathbf{M}}_n^\top\mathbf{U}^\top, \dots, \mathbf{U}\widetilde{\mathbf{M}}_n\mathbf{F}^{*(n-1)}\widetilde{\mathbf{M}}_n^\top\mathbf{U}^\top).$$

Thus $\mathbf{U}\widetilde{\mathbf{M}}_n\mathbf{F}^{*0}\widetilde{\mathbf{M}}_n^\top\mathbf{U}^\top$ is an \mathbb{E} -linear combination of the public quadratic forms. Since $\mathbf{U}\widetilde{\mathbf{M}}_n$ is invertible, the rank of this linear combination is the rank of \mathbf{F}^{*0} , which is $r + v$.

Following the analysis of [19, Theorem 2], we see that the effect of the minus modifier on the matrix representation of \mathcal{F} over $\mathbb{A} \times \mathbb{F}^v$ is to add to it constant multiples of itself with a cyclic shift of the rows and columns down and to the right within the HFE block. Thus for HFEV-, \mathbf{F}^{*0} has the shape given in Figure 2. The rank of this quadratic form is $r + a + v$.

The solution of the MinRank instance provides an equivalent transformation \mathcal{T}' to the output transformation \mathcal{T} (up to the choice of extension to full rank) and a matrix \mathbf{L} representing the low Q-rank quadratic form $\mathbf{U}'\widetilde{\mathbf{M}}_n\widehat{\mathbf{F}}^{*0}\widetilde{\mathbf{M}}_n^\top\mathbf{U}'^\top$ over $\mathbb{A} \times \mathbb{F}^v$, where $\mathcal{P} = \mathcal{T}' \circ \phi^{-1} \circ \widehat{\mathcal{F}} \circ \phi \circ \mathcal{U}'$ for an equivalent private key $(\mathcal{T}', \widehat{\mathcal{F}}, \mathcal{U}')$. Now that the correct output transformation is recovered, it remains to recover the vinegar subspace of the map \mathcal{L} defined by $\mathbf{L} = \mathbf{U}'\widetilde{\mathbf{M}}_n\widehat{\mathbf{F}}^{*0}\widetilde{\mathbf{M}}_n^\top\mathbf{U}'^\top$.

First, note that the kernel of \mathbf{L} as a linear map is orthogonal to the vinegar subspace, so we may simplify the analysis by projecting onto the orthogonal complement of a codimension one subspace of the kernel. Let $\widehat{\mathcal{L}}$ denote the composition of \mathcal{L} with this projection. The strategy now is to compose codimension

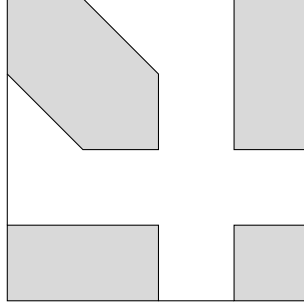


Fig. 2. The shape of the central map of HFEv- composed with the minus projection over $\mathbb{A} \times \mathbb{F}^v$. The shaded areas represent possibly nonzero entries.

one projection mappings π with the transformation $\widehat{\mathcal{L}}$ to filter out the vinegar variables. It suffices to choose projections whose kernels are orthogonal to $\ker(\widehat{\mathbf{L}})$.

If there is a nontrivial intersection between the kernel of π and the vinegar subspace, the rank of the matrix representation of $\widehat{\mathcal{L}} \circ \pi$, $\mathbf{\Pi} \widehat{\mathbf{L}} \mathbf{\Pi}^\top$, will be reduced. In contrast, if this intersection is empty, the rank of $\mathbf{\Pi} \widehat{\mathbf{L}} \mathbf{\Pi}^\top$ should remain the same. To see this, note that by an argument symmetric to that of [19, Lemma 1] we may equivalently define $\widehat{\mathcal{L}} \circ \pi$ by

$$\widehat{\mathcal{L}} \circ \pi = \mathcal{U}^{-1} \circ [(\phi \circ \pi_1 \circ \phi^{-1} \circ \mathcal{S}_1) \times \pi_2] \circ \mathcal{S}_2,$$

where $\mathcal{S}_1 : \mathbb{F}^n \rightarrow \mathbb{F}^n$ is nonsingular, $\mathcal{S}_2 : \mathbb{F}^{n+v} \rightarrow \mathbb{F}^n \times \mathbb{F}^v$ is an isomorphism, $\pi_1 : \mathbb{E} \rightarrow \mathbb{E}$ has degree at most q^{n-r-a} (since the intersection of the image of $\widehat{\mathcal{L}} \circ \pi$ and the HFE subspace is at least $(r+a)$ -dimensional) and $\pi_2 : \mathbb{F}^v \rightarrow \mathbb{F}^v$ is linear. Since the degree bound of the central HFE quadratic form is q^{r+a} , the highest monomial degree in the composition of π_2 with this map is bounded by q^{n-1} , thus the polynomials $\pi_1, \pi_1^q, \dots, \pi_1^{q^{r+a}}$ are linearly independent.

The probability that the linear form defining $\ker(\pi)$ which is orthogonal to the kernel of $\widehat{\mathbf{L}}$ lies in the vinegar subspace is $q^{-(r+a+1)}$. Once such a vector is recovered, this step is repeated on the orthogonal complement of the discovered vectors until a basis for the vinegar subspace is found. Thus the complexity of this method when fixing one variable to make the ideal zero dimensional is

$$\text{Comp}_{MP} = \mathcal{O} \left(\binom{n+r+v}{r+a+v}^2 \binom{n-a}{2} \left((r+a+v+1)^3 q^{r+a+1} \right) \right).$$

4.2 Projection then MinRank

Another approach using MinRank is a “project-then-MinRank” approach. In this strategy, one randomly projects the plaintext space onto a codimension k subspace and then applies the MinRank attack. Since the projection π cannot increase the Q-rank of the central map, the Q-rank is at most $r+a+v$.

We may choose $k = n - r - a - v$, and expect that the rank of $\mathcal{P} \circ \pi$ is still $r + a + v$, due to the fact that the HFE component is still of full rank, as noted in the previous section. If, however, there is a nontrivial intersection between the kernel of π and the vinegar subspace, the rank of this quadratic form will be less than $r + a + v$. The probability this occurs is $q^{k-n} = q^{-(r+a+v)}$.

Generalizing, we may project further in an attempt to eliminate possibly more vinegar variables and reduce the rank further. The minors system of a MinRank attack at rank r is fully determined if the square of r less than the number of variables bounds the number of public equations; thus, if the image of π is of dimension at least the sum of $\sqrt{n-a}$ and r , the minors system is still fully determined. Therefore, consider eliminating c vinegar variables. This requires k to be at least $n - a - r + c - \sqrt{n-a}$. The probability that there is a c -dimensional intersection between the kernel of π and the vinegar subspace is then $q^{c(k-n) - \binom{c}{2}} \geq q^{\binom{c+1}{2} - cr - ca - c\sqrt{n-a}}$.

Once at least one vinegar variable is found, the new basis can be utilized to filter out the remaining vinegar variables as in the previous method. The complexity of the this method with one variable fixed is

$$Comp_{PM} = \mathcal{O} \quad q^{c(r+a+\sqrt{n-a}) - \binom{c+1}{2}} \binom{n+r+v-c}{r+a+v-c}^2 \binom{n-a}{2} \binom{n-a}{2}$$

5 The Distinguishing Based attack

In this section we present our distinguishing based attack against the HFEv-signature scheme. We restrict to the case of $\mathbb{F} = \text{GF}(2)$. The idea of the attack is closely related to the direct attacks with projection (also known as the hybrid approach). We define

$$\mathcal{V} = \left\{ \sum_{i=n+1}^{n+v} \lambda_i \mathcal{U}_i \mid \lambda_i \in \{0, 1\} \right\},$$

where \mathcal{U}_i denotes the i -th component of the affine transformation $\mathcal{U} : \mathbb{F}^{n+v} \rightarrow \mathbb{F}^{n+v}$. Therefore, \mathcal{V} is the space spanned by the affine representations of the vinegar variables x_{n+1}, \dots, x_{n+v} . Our attack is based on the following two observations.

- Consider the two HFEv- public keys $\mathcal{P}_1 = \text{HFEv-}(n, D, a, v_1)$ and $\mathcal{P}_2 = \text{HFEv-}(n, D, a, v_2)$. Before applying a Gröbner basis algorithm to the systems, we fix $a+v_1$ variables in \mathcal{P}_1 and $a+v_2$ variables in \mathcal{P}_2 to get determined systems. As shown in Table 1 and Figure 3, direct attacks against these systems behave differently. In particular, we can distinguish between determined instances of the two systems \mathcal{P}_1 and \mathcal{P}_2 by looking at the step degrees of the F_4 algorithm. This remains possible even when adding (not too many) additional linear equations to the systems \mathcal{P}_1 and \mathcal{P}_2 (thus guessing some of the variables) before applying a Gröbner basis method (hybrid approach).

v	HFEv-(26, 17, 1, v)	HFEv-(33, 9, 3, v)
0	2,3,4,3,4	2,3,4,4,4
1	2,3,4,4,4	2,3,4,5,4
2	2,3,4,5,4	2,3,4,5,5
3	2,3,4,5,5	2,3,4,5,5,5,5,6
4	2,3,4,5,5,5,5	2,3,4,5,6,6
5	2,3,4,5,6	
random system	2,3,4,5,6	2,3,4,5,6,6

Table 1. Step degrees of the F_4 algorithm against determined HFEv- systems for different values of v

- Let us consider the special case where $v_2 = v_1 - 1$ holds. By adding one linear equation $\ell \in \mathcal{V}$ to \mathcal{P}_1 , we remove the influence of one of the vinegar variables from the system \mathcal{P}_1 . A direct attack against the so obtained system \mathcal{P}'_1 therefore behaves in exactly the same way as a direct attack against the system \mathcal{P}_2 (see Table 2).

5.1 The Distinguisher

Based on the two above observations, we can now construct a distinguisher as follows. We start with an HFEv- public key $\mathcal{P} = \text{HFEv-}(n, D, a, v)$. \mathcal{P} consists of $n - a$ quadratic equations in $n + v$ variables over the field $\text{GF}(2)$. After adding the field equations $\{x_i^2 - x_i : i = 1, \dots, n + v\}$, we append k randomly chosen linear equations ℓ_1, \dots, ℓ_k to the system. Therefore, our new system \mathcal{P}' consists of

- the $n - a$ quadratic HFEv- equations from \mathcal{P}
- $n + v$ field equations $x_i^2 - x_i = 0$ ($i = 1, \dots, n + v$)
- the k linear equations ℓ_1, \dots, ℓ_k .

Altogether, the system \mathcal{P}' consists of $2n - a + v + k$ equations in $n + v$ variables.

After having constructed the system \mathcal{P}' , we solve it via a Gröbner basis algorithm. Due to Observation 2, the behaviour of this algorithm should depend on the fact whether one of the linear equations ℓ_i added to the system (or a linear combination of the ℓ_i) is an element of the vinegar space \mathcal{V} . In fact, we can observe a difference in the step degrees of the algorithm (see Example 1 below).

Formally written, we can use our technique to distinguish between the two cases

$$\left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\} \cap \mathcal{V} = \emptyset \text{ and} \\ \left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\} \cap \mathcal{V} \neq \emptyset. \quad (2)$$

However, in most cases that $\left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\} \cap \mathcal{V} \neq \emptyset$, the intersection contains only a single equation $\tilde{\ell}$.

Remark: We have to note here that the number k of linear equations added to the system \mathcal{P} is upper bounded by a value $\bar{k}(n, D, a, v)$. When adding more than \bar{k} linear equations to the system, a distinction between the two cases of (2) is no longer possible.

Example 1: We consider HFEV- systems with $(n, D, a) = (33, 9, 3)$ and varying values of $v \in \{0, \dots, 4\}$. The resulting HFEV- public keys are systems of $n - a = 30$ quadratic equations in $n + v$ variables. After appending the field equations $\{x_i^2 - x_i = 0\}$ to the systems, we added randomly chosen linear equations to reduce the effective number of variables in our systems. Figure 3 shows the degree of regularity of a direct attack using F_4 against the (projected) systems. For comparison, the figure also contains data for a random system of the same size.

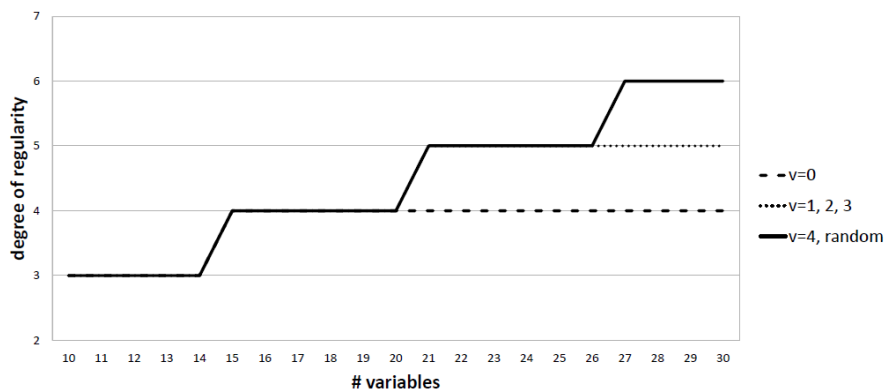


Fig. 3. Direct attack against (projected) HFEV- systems with $(n, D, a) = (33, 9, 3)$ and varying values of v

As Figure 3 shows, there exists, for every parameter set (n, D, a, v) a number \bar{k} such that

- 1) When adding less than \bar{k} linear equations to the system, the degree of regularity of a direct attack against the projected system is the same as that of a direct attack against the unprojected system.
- 2) When adding $k \geq \bar{k}$ linear equations, the system behaves exactly like a random system of the same size.

Let us now look at our distinguisher. For this, we skip the parameter set $(n, D, a, v) = (33, 9, 3, 0)$ since, in this case, $\mathcal{V} = \emptyset$ holds. However, as Table 2

shows, we can, for each of the values $v \in \{1, \dots, 4\}$, distinguish between the two cases of (2).

v	\bar{k}	$n - \bar{k}$	step degrees of F_4	
			for $\mathcal{L} \cap \mathcal{V} = \emptyset$	for $\mathcal{L} \cap \mathcal{V} = \{\tilde{\ell}\}$
4	3	27	1,2,3,4,5,6	1,2,3,4,5,5,5
3	4	26	1,2,3,4,5,5,5	1,2,3,4,5,5
2	4	26	1,2,3,4,5,5	1,2,3,4,5,4
1	9	21	1,2,3,4,5	1,2,3,4,4,4

Table 2. Distinguisher Experiments on HFEv-(33, 9, 3, v) systems for different values of v

For abbreviation, we use in the table $\mathcal{L} := \left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\}$. Note that the evolution of the step degrees for HFEv-(33,9,3,4) is the same as for a random system of the same size.

5.2 The Attack

Based on the distinguisher presented in the previous section, we can construct an attack against HFEv- as follows. By performing the distinguishing experiment with a large number of systems \mathcal{P}' (containing different linear equations), we can find a set of k linear equations ℓ_1, \dots, ℓ_k such that $\left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\} \cap \mathcal{V} = \{\tilde{\ell}_1\}$. Using this, we can determine the exact form of $\tilde{\ell}_1$ as follows. Note that there exist coefficients $\alpha_i \in \{0, 1\}$ ($i = 1, \dots, k$) such that

$$\tilde{\ell}_1 = \sum_{i=1}^k \alpha_i \cdot \ell_i.$$

In order to determine the exact form of this linear combination, we remove one of the linear equations (say ℓ_1) from the system \mathcal{P}' and add another randomly chosen linear equation. If we still can observe a difference in the behaviour of a direct attack compared to a random choice of linear equations, we know that the coefficient α_1 must be 0. Otherwise, the coefficient α_1 must be 1, and we have to add ℓ_1 back to the system.

We repeat this step for $i = 2, \dots, k$ to determine the values of all the coefficients α_i ($i = 1, \dots, k$). This will give us the exact form of the linear equation $\tilde{\ell}_1 \in \mathcal{V}$. We denote this technique as “remove-and-add” strategy.

Having found $\tilde{\ell}_1$, we add it to the original HFEv- (n, D, a, v) system. The resulting system will behave exactly like an HFEv- $(n, D, a, v - 1)$ system, and we can again use our distinguisher and repeat the above procedure to find a second linear equation $\tilde{\ell}_2 \in \mathcal{V}$. Note that this will be much easier than finding $\tilde{\ell}_1$ (see next section).

After having found v linear independent equations $\tilde{\ell}_1, \dots, \tilde{\ell}_v \in \mathcal{V}$ and adding them to the HFEV- system, the resulting system will behave exactly like an HFE-(n,D,a) system (i.e. we have no vinegar variables any more). We can then use any attack against HFE- (e.g. the key recovery attack of Vates et al. [19] or a direct attack) to break the scheme. We analyze the complexity of our distinguisher and this attack in the next section.

Let us briefly return to Example 1. When we start with the system $\mathcal{P} = \text{HFEV-}(33,9,3,4)$, we can use our distinguisher to find a set $\{\ell_1, \dots, \ell_k\}$ of linear equations such that $\left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\} \cap \mathcal{V} = \{\tilde{\ell}_1\}$. After having recovered the exact form of $\tilde{\ell}$, we can append it to the system \mathcal{P} , which will then behave exactly like an HFEV-(33,9,3,3) system. Let us denote this new system by $\mathcal{P}^{(1)}$. We can then use the distinguisher on $\mathcal{P}^{(1)}$ to obtain a second linear equation $\tilde{\ell}_2 \in \mathcal{V}$. Adding $\tilde{\ell}_2$ to the system $\mathcal{P}^{(1)}$ leads to a system $\mathcal{P}^{(2)}$ behaving exactly like a HFEV-(33,9,3,2) system. By continuing this process, we finally obtain the system $\mathcal{P}^{(4)}$ corresponding to an HFEV-(33,9,3,0) system. We can then break this scheme by using any attack on HFE-.

Algorithm 1 Our distinguishing based attack

Input: HFEV-(n, D, a, v) public key \mathcal{P}

Output: equivalent HFE-(n, D, a) public key $\tilde{\mathcal{P}}$

- 1: Append \bar{k} randomly chosen linear equations $\ell_1, \dots, \ell_{\bar{k}}$ in the variables x_1, \dots, x_{n+v} (as well as the field equations $x_i^2 - x_i = 0$) to the system \mathcal{P} and solve it by F_4 .
 - 2: Repeat this step until the F_4 -step degrees differ from the standard case. This means that we have found a set of linear equations ℓ_1, \dots, ℓ_k such that $\left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\} \cap \mathcal{V} = \{\tilde{\ell}_1\}$
 - 3: Determine the exact form of $\tilde{\ell}$ by the above described “remove-and-add” strategy.
 - 4: Append the linear equation $\tilde{\ell}$ to the system \mathcal{P} . The resulting system \mathcal{P}' will behave exactly like an HFEV-($n, D, a, v-1$) public key.
 - 5: Repeat the above steps until having found v linear independent equations $\tilde{\ell}_1, \dots, \tilde{\ell}_k \in \mathcal{V}$.
 - 6: **return** $\tilde{\mathcal{P}} = (\mathcal{P}, \tilde{\ell}_1, \dots, \tilde{\ell}_v)$
-

6 Complexity Analysis

In the first step of our attack, we have to find one linear equation $\tilde{\ell} \in \mathcal{V}$ by using our distinguisher and a following application of the “remove-and-add” strategy described in the previous section. Therefore, the complexity of this first step of our attack is determined by three factors:

1. The number of times we have to run the distinguisher in order to find a set of linear equations ℓ_1, \dots, ℓ_k such that $\left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\} \cap \mathcal{V} = \{\tilde{\ell}\}$,
2. The cost of one run of the distinguisher and

3. The cost of recovering the exact form of $\tilde{\ell}$.

The first number is determined by

- The probability that a randomly chosen linear equation in $n + v$ variables is contained in the space \mathcal{V} spanned by the linear representation of the vinegar variables $\mathcal{U}_{n+1}, \dots, \mathcal{U}_{n+v}$. A randomly chosen linear equation $\bar{\ell}$ in $n + v$ variables can be seen as a linear combination of the components of \mathcal{U} , i.e.

$$\bar{\ell} = \sum_{i=1}^{n+v} \lambda_i \cdot \mathcal{U}_i. \quad (3)$$

The reason for this is that \mathcal{U} is an invertible map from \mathbb{F}^{n+v} to itself, which means that the components of \mathcal{U} form a basis of this space. There are 2^{n+v} choices for the parameters λ_i ($i = 1, \dots, n + v$). On the other hand, every element $\bar{\ell}$ of the space \mathcal{V} spanned by the linear transformations of the vinegar variables v_1, \dots, v_v can be written in the form

$$\tilde{\ell} = \sum_{i=n+1}^{n+v} \lambda_i \cdot \mathcal{U}_i.$$

The probability that a randomly chosen linear equation $\bar{\ell}$ lies in \mathcal{V} is therefore given by

$$\text{prob}(\bar{\ell} \in \mathcal{V}) = 2^{-n}. \quad (4)$$

The reason for this is that all the coefficients λ_i ($i = 1, \dots, n$) in the representation (3) of $\bar{\ell}$ must be zero.

- The number of linear equations (and linear combinations thereof) added to the public key. When adding k linear equations ℓ_1, \dots, ℓ_k to the public key, we do not have to consider only the k equations ℓ_1, \dots, ℓ_k itself, but also all linear combinations of the form

$$\ell = \sum_{i=1}^k \lambda_i \cdot \ell_i.$$

The total number of linear equations we have to consider is therefore not k , but 2^k .

Therefore, when adding k linear equations ℓ_1, \dots, ℓ_k to the public key, the probability of finding one linear equation $\tilde{\ell} \in \mathcal{V}$, is given by

$$\text{prob} = 1 - (1 - 2^{-n})^{2^k} \approx 2^{k-n}.$$

In order to find one linear equation $\tilde{\ell} \in \mathcal{V}$, we therefore have to run our distinguisher about 2^{n-k} times.

A single run of our distinguisher corresponds to one run of the F_4 algorithm. The cost of this can be estimated as

$$\text{Comp}_{F_4} = 3 \cdot \binom{n'}{\ell_{\text{reg}}}^2 \cdot \binom{n'}{2}.$$

where n' is the number of variables in the quadratic system and d_{reg} is the so called degree of regularity.

Note that this formula assumes that the linear systems appearing during the attack are solved using a sparse Wiedemann solver. Furthermore we use the fact that the system is defined over the field $\text{GF}(2)$, which reduces the number of terms in the high degree polynomials.

With regard to the number n' of variables we find that the linear equations added to the public key are “absorbed” at a very early step of the F_4 algorithm, i.e. they are used to reduce the number of variables in the system. This fact is illustrated in Table 3. In the table, we consider two random systems, both containing 25 quadratic equations. However, while the equations of system A are polynomials in 25 variables, the polynomials of system B contain 35 variables. On the other hand, the system B additionally contains 10 linear equations.

	25 equations, 25 variables			25 quadr. + 10 lin. equations, 35 variables		
step	degree	matrix size	time (s)	degree	matrix size	time (s)
				1	10×36	0.0
				1	20×36	0.0
1	2	25×326	0.0	2	330×631	0.0
2	3	652×2626	0.02	3	650×2626	0.02
3	4	$7894 \times 14\,498$	1.27	4	$7864 \times 15\,568$	1.34
4	5	$52\,488 \times 52\,956$	79.86	5	$52\,197 \times 52\,665$	80.26
5	6	$248\,705 \times 245\,506$	179.34	6	$248\,273 \times 108\,524$	182.24

Table 3. Experiments with random systems

As the table shows, both systems behave very similarly. Starting at step 2 (degree 3), there is no significant difference between the matrix sizes or the running times of the single steps between the two systems.

We can therefore conclude that the quadratic systems we consider in our distinguishing based attack ($n - a$ quadratic equations + k linear equations in $n + v$ variables) behave just like systems of $n - a$ quadratic equations in $n + v - k$ variables.

The cost of recovering the exact form of $\tilde{\ell}$ is negligible in comparison to finding linear equations ℓ_1, \dots, ℓ_k such that $\left\{ \sum_{i=1}^k \lambda_i \ell_i \mid \lambda_i \in \{0, 1\} \right\} \cap \mathcal{V} = \{\tilde{\ell}\}$. Remember that $\tilde{\ell}$ can be written as a linear combination of ℓ_1, \dots, ℓ_k , i.e. $\tilde{\ell} = \sum_{i=1}^k \lambda_i \cdot \ell_i$.

As described in the previous section, we remove for this one linear equation ℓ_i from the system \mathcal{P}' . By adding a randomly chosen linear equation, we obtain a system \mathcal{P}'' of the same dimensions. We apply the F_4 algorithm against the two systems \mathcal{P}' and \mathcal{P}'' . If we observe a difference in the behavior of the algorithm, we know that the coefficient λ_i in the above linear combination is 1. Otherwise we have $\lambda_i = 0$. By running this test for all $i \in \{1, \dots, k\}$, we can determine

all the coefficients λ_i and therefore recover $\tilde{\ell}$. In order to recover $\tilde{\ell}$, we therefore need $2 \cdot k$ runs of the F_4 algorithm, which is far less than the 2^{n-k} F_4 -runs above. Therefore, we do not have to consider this step in our complexity analysis.

Altogether, we can estimate the complexity of this first step of our attack by

$$Comp_{\text{Dist}; \text{ classical}} = 2^{n-k} \cdot 3 \cdot \binom{n+v-k}{d_{\text{reg}}}^2 \cdot \binom{n+v-k}{2} \cdot \left(\right) \quad (5)$$

In the presence of quantum computers, we can speed up the searching step of this attack using Grover's algorithm. Thus we get

$$Comp_{\text{Dist}; \text{ quantum}} = 2^{(n-k)/2} \cdot 3 \cdot \binom{n+v-k}{d_{\text{reg}}}^2 \cdot \binom{n+v-k}{2} \cdot \left(\right)$$

Note that this assumption of the complexity is very optimistic, since it assumes a perfect "square-root" speed up by Grover's algorithm. Since quantum algorithms must be reversible, it is not clear if this is possible.

As equation (5) shows, the complexity decreases when we increase the number k of linear equations added to the public key. However, as already mentioned in the previous section, our distinguisher fails when k is too large. We denote the maximal value of k for which our distinguisher works by $\bar{k}(n, D, a, v)$.

In order to remove all the vinegar variables from the system \mathcal{P} , we have to repeat the above process v times. However, with decreasing v we find (see Table 2)

- 1) the number \bar{k} of linear equations that we can add to the public system increases, reducing the number of F_4 -runs.
- 2) the degree of regularity of the systems generated by our distinguisher decreases, reducing the complexity of a single F_4 -run.

Therefore, the following steps of our attack will be much faster than the first step. This means, that we can estimate the complexity of the whole attack as in formula (5).

However, in order to estimate the complexity of our attack against an HFEv- (n, D, a, v) scheme in practice, we still have to answer the following two questions.

- What is the maximal number \bar{k} of linear equations we can add to the public key such that our distinguisher works?
- What is the degree of regularity of the systems generated by our distinguisher?

In order to answer these questions, we once more consider Example 1 (see previous section).

First, let us consider the second question. As a comparison of Table 2 and Figure 3 shows, the degree of regularity of solving the systems generated by our distinguisher corresponds exactly to the degree of regularity of an unprojected HFEv- system with parameters (n, D, a, v) . As stated in [20], we can estimate this value as

$$d_{\text{reg}} = \left\lfloor \frac{r+a+v+7}{3} \right\rfloor \left(\right) \quad (6)$$

where $r = \lfloor \log_q(D - 1) \rfloor + 1$.

To answer the first question, let us take a closer look at the behavior of the hybrid approach against random systems (see Figure 3). We start with a random system of 30 quadratic equations in 30 variables over GF(2). After appending the field equations $x_i^2 - x_i = 0$ ($i = 1, \dots, 30$), we add $k \in \{0, \dots, 20\}$ linear equations to the system. Table 4 shows for which values of k we reach given values of regularity.

d_{reg}	# k of added linear equations
3	for $k \geq 16$
4	for $10 \leq k \leq 15$
5	for $4 \leq k \leq 9$
6	for $k \leq 3$

Table 4. Degree of regularity of projected random systems with 30 equations

Let us define $\hat{k}(d)$ to be the maximal number of linear equations we can add to the random system, such that the degree of regularity of a direct attack against the system is greater or equal to d , i.e. $\hat{k}(6) = 3$, $\hat{k}(5) = 9$ and $\hat{k}(4) = 15$.

By comparing these numbers with the values of \bar{k} listed in Table 2, we find

$$\hat{k}(d^*) \leq \bar{k} \leq \hat{k}(d^*) + 1,$$

where d^* is the degree of regularity of a direct attack against an HFEV- (n, D, a, v) scheme (see equation (6)).

In order to estimate the complexity of our attack against an HFEV- (n, D, a, v) scheme, we therefore proceed as follows.

1. We compute the degree of regularity of the unprojected HFEV- (n, D, a, v) system (see equation (6)). Denote the result by d^* .
2. We estimate the maximal number \bar{k} of linear equations we can add to the public HFEV- system by $\hat{k}(d^*)$. This value can be obtained as follows.

The degree of regularity of a random system of $m = n - a$ quadratic equations in n' variables over GF(2) can be estimated as the smallest index d for which the coefficient of X^d in

$$\frac{1}{1 - X} \cdot \left(\frac{1 - X^2}{1 - X} \right)^{n'} \cdot \left(\frac{1 - X^2}{1 - X^4} \right)^m$$

is non-positive [21].

We can use this equation to determine the values of $\hat{k}(d^*)$.

By substituting the so obtained values of \bar{k} and d^* into formula (5), we therefore get a close estimation of the complexity of our distinguishing based attack against an HFEV- (n, D, a, v) system.

Remark: The above procedure allows us to get an estimation of the complexity of our distinguishing based attack against a given HFEV- scheme. However, it seems to be a very hard task to find a closed formula for this complexity.

Example 2: Consider an HFEV- system over $\text{GF}(2)$ with $(n, D, a, v) = (91, 5, 3, 2)$. We obtain $r = \lfloor \log_2(D-1) \rfloor + 1 = 3$. The degree of regularity of a direct attack against the HFEV- system (with field equations) is given by

$$d_{\text{reg}} = \left\lfloor \frac{3+3+2+7}{3} \right\rfloor = 5.$$

Therefore, we get

$$\text{Comp}_{\text{direct}} = 3 \cdot \binom{88}{5}^2 \cdot \binom{88}{2} \approx 2^{63.9}.$$

After adding $k = 68$ randomly chosen linear equations to the system, the step degrees of the $F4$ algorithm look like 1; 1, 2, 3, 4. When one of the linear equations was chosen from the vinegar space \mathcal{V} , we obtain 1; 1, 2, 3, 3.

Therefore, we can estimate the complexity of our distinguisher by

$$\text{Comp}_{\text{Distinguisher}} = 2^{23} \cdot \binom{25}{4}^2 \cdot \binom{25}{2} \approx 2^{60.1},$$

which is nearly 16 times faster than a direct attack.

The “MinRank-then-project” approach has a complexity estimated by

$$\text{Comp}_{\text{MP}} = 3 \cdot \binom{96}{8}^2 \cdot \binom{88}{2} \approx 2^{87.4},$$

while the complexity of the “project-then-MinRank” approach has complexity

$$\text{Comp}_{\text{PM}} = 2^{14} \cdot 3 \cdot \binom{95}{7}^2 \cdot \binom{88}{2} \approx 2^{92.6}.$$

Therefore, for the above parameter set, the distinguishing based attack is the most efficient classical attack against HFEV-.

With regard to the memory consumption, we get

$$\text{Memory}_{\text{direct}} = \binom{88}{5}^2 \approx 2^{50.4},$$

$$\text{Memory}_{\text{MP}} = \binom{96}{8}^2 \approx 2^{73.9},$$

$$\text{Memory}_{\text{PM}} = \binom{95}{7}^2 \approx 2^{66.7},$$

$$\text{Memory}_{\text{Distinguisher}} = \binom{25}{4}^2 \approx 2^{27.3}.$$

As these data show, the distinguishing based attack requires much less memory than the direct and the MinRank attack. Since attacks against large instances of multivariate schemes often fail due to memory restrictions, the small memory consumption is a huge benefit of this attack.

Remark: The comparably low complexity of our attack in Example 2 is caused by the small number of vinegar variables in the system. Due to this, the distinguisher works also for relatively small numbers of variables, which enables us to add a large number of linear equations to the system. This again reduces the number of distinguisher runs and therefore the complexity of the attack. (In the case of the example, we found that the distinguisher works for only 25 variables in the system, due to which we had to run our distinguisher only 2^{23} times.)

When the number v of vinegar variables increases, we can not distinguish between the two cases at 25 variables any more. We have to reduce the number of linear equations added to the system and therefore have to run the distinguisher much more often (and for larger systems). Therefore, for larger values of v , the complexity of our attack increases.

For the parameter sets usually used in HFEV- like schemes (and suggested for the National Institute of Standards and Technology (NIST) call for proposals), the direct attack is usually more efficient than our attack. However, in terms of memory consumption, our attack is still much better.

7 Possible Future Work

In this section we shortly describe a strategy to reduce the complexity of our attack. However, since we have neither enough space nor time to present our idea completely, we leave a detailed analysis as future work.

In the distinguishing step of our attack, we solve a large number of multivariate systems using a direct attack. These systems are obtained by adding k linear equations to a multivariate quadratic system \mathcal{P} of m equations in $n+v$ variables (or equivalently projecting the system to a $n+v-k$ dimensional subspace). In Section 5, these projections were chosen at random.

The main idea to reduce the complexity of this step is now to select the projection in a slightly nonrandom fashion. In particular, we consider a projection in two steps. We apply a projection $\tilde{\pi}$ of corank $k+1$ to the system \mathcal{P} and derive from this a set of corank k projections $\{\pi_i\}$. In this case, we can treat the image of $\tilde{\pi}$ in the plaintext space as being generated by the variables $x_1, \dots, x_{n+v-k-1}$, while the image of each of the projections π_i is generated by the same variables plus one additional variable x_{n+v-k} , which defines a 1-dimensional subspace of the kernel of $\tilde{\pi}$, which will vary depending on the choice of π_i .

During the computation of a Gröbner basis of $\mathcal{P}(\pi_i) = (f_1(\pi_i), \dots, f_m(\pi_i))$, the F_4 algorithm looks for polynomials p_j of degree $d-2$ such that $\sum p_j \cdot f_j(\pi_i) = q$, where q is a polynomial of degree at most $d-1$.

Our strategy will be to first solve for all p_j in the variables $x_1, \dots, x_{n+v-k-1}$, such that

$$\sum p_j f_j(\pi_i) = q \pmod{x_{n+v-k}}.$$

As the above equation is equivalent to $\sum p_j f_i(\tilde{\pi}) = q$, this computation can be reused for multiple different choices of π_i . By doing so, we therefore can reduce the effort of computing the Gröbner basis needed during the application of our distinguisher.

However, in order to find the exact amount of saving, much more work is required. We therefore leave an exact analysis of the above mentioned idea as future work.

Another topic for future work is a precise complexity analysis of our attack. The complexity analysis presented in Section 6 is based much on heuristics and experiments. In particular, formula (5) contains the parameters \bar{k} and d_{reg}^* , which (so far) could only be determined experimentally. It therefore would be desirable to develop a formula which computes the complexity of our attack for given HFEv- parameters n , D , a and v .

Acknowledgements

We thank the anonymous reviewers of PQCrypto for their valuable comments which helped to improve this paper. In particular we want to thank the shepherd for her help in creating the final version of this paper.

References

1. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. EUROCRYPT 1999, LNCS **1592** (1999) 206–222
2. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. ACNS 2005, LNCS **3531** (2005) 164–175
3. Patarin, J., Courtois, N., Goubin, L.: Quartz, 128-bit long digital signatures. In Naccache, D., ed.: CT-RSA. Volume 2020 of Lecture Notes in Computer Science., Springer (2001) 282–297
4. Bettale, L., Faugère, J.C., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology **3** (2009) 177–197
5. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1979)
6. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: EUROCRYPT. (1996) 33–48
7. Faugere, J.C.: Algebraic cryptanalysis of hidden field equations (HFE) using grobner bases. CRYPTO 2003, LNCS **2729** (2003) 44–60
8. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. Advances in Cryptology - CRYPTO 1999, Springer **1666** (1999) 788

9. Courtois, N., Klimov, A., Patarin, J., A.Shamir: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. EUROCRYPT 2000, LNCS **1807** (2000) 392–407
10. Faugere, J.C.: A new efficient algorithm for computing grobner bases (f4). Journal of Pure and Applied Algebra **139** (1999) 61–88
11. Faugere, J.C.: A new efficient algorithm for computing grobner bases without reduction to zero (f5). ISSAC 2002, ACM Press (2002) 75–83
12. Mohamed, M.S.E., Ding, J., Buchmann, J.: Towards algebraic cryptanalysis of hfe challenge 2. In: ISA. Volume 200 of Communications in Computer and Information Science., Springer (2011) 123–131
13. Ding, J., Hodges, T.J.: Inverting HFE systems is quasi-polynomial for all fields. In Rogaway, P., ed.: Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Volume 6841 of Lecture Notes in Computer Science., Springer (2011) 724–742
14. Ding, J., Kleinjung, T.: Degree of regularity for HFE-. IACR Cryptology ePrint Archive **2011** (2011) 570
15. Ding, J., Yang, B.Y.: Degree of regularity for hfev and hfev-. In Gaborit, P., ed.: PQCrypto. Volume 7932 of Lecture Notes in Computer Science., Springer (2013) 52–66
16. Bettale, L., Faugère, J., Perret, L.: Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. Des. Codes Cryptography **69** (2013) 1–52
17. Porras, J., Baena, J., Ding, J.: ZHFE, A new multivariate public key encryption scheme. In Mosca, M., ed.: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. Volume 8772 of Lecture Notes in Computer Science., Springer (2014) 229–245
18. Cabarcas, D., Smith-Tone, D., Verbel, J.A.: Key recovery attack for ZHFE. [22] 289–308
19. Vates, J., Smith-Tone, D.: Key recovery attack for all parameters of HFE-. [22] 272–288
20. Petzoldt, A.: On the complexity of the hybrid approach on hfev-. Cryptology ePrint Archive, Report 2017/1135 (2017) <https://eprint.iacr.org/2017/1135>.
21. Yang, B., Chen, J.: Theoretical analysis of XL over small fields. In Wang, H., Pieprzyk, J., Varadharajan, V., eds.: Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-15, 2004. Proceedings. Volume 3108 of Lecture Notes in Computer Science., Springer (2004) 277–288
22. Lange, T., Takagi, T., eds.: Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings. Volume 10346 of Lecture Notes in Computer Science., Springer (2017)