Practical Attack on All Parameters of the HPPC Signature Scheme

Pierre Briaud[1], Maxime Bros[2], Ray Perlner[3], and Daniel Smith-Tone[3,4]

[1] Simula UiB, Bergen, Norway
pierre@simula.no
[2] Associate, National Institute of Standards and Technology (NIST), Maryland, USA
maxime.bros@nist.gov
[3] National Institute of Standards and Technology (NIST), Maryland, USA
daniel.smith@nist.gov
[4] University of Louisville, Louisville KY, USA

Abstract. HPPC is a multivariate signature scheme submitted to the NIST PQC standardization process in response to the recent call for additional signature schemes. We show that, despite some non-standard notational choices in the submission document, HPPC can be viewed as a special case of the well-studied, but broken for all practical parameters, HFE signature scheme. We further show that the HPPC construction introduces additional structure that further weakens the scheme. For instance, the central map has Q-rank 2 independently from the degree D of the central polynomial that is used.
Using these observations, we show that HPPC is weaker against the direct attack than claimed in the submission document and more crucially that all parameter sets can be practically broken using MinRank techniques. For instance, with a very naive implementation, we have been able to recover an equivalent key in approximately 8 minutes for security level 2, an hour and a half for security level 4, and slightly more than 7 hours for security level 5.

1 Introduction

After selecting the first post-quantum standards for encryption and signature schemes, see [1], the National Institute of Standards and Technology (NIST) announced an expansion to their post-quantum cryptography standardization project and released a call for additional signature proposals [19].

The HPPC scheme [25,26] is one of the 10 multivariate submissions to this new call. It is in fact the only one that can be classified as a multivariate "big-field" scheme (other candidates such as SNOVA [29] use field extensions but in different ways or for different purposes). Its signature size is barely the smallest among all candidates, even though the public key is quite large compared to most of the other multivariate proposals. From a security point of view, HPPC was claimed to be as resistant as an HFE scheme [23] with a central polynomial of maximal degree. In the HFE case, it has been known for a long time that both

the direct attack [10] and rank attacks [17] have a complexity which is essentially an increasing function of the degree of this central polynomial. This relationship, ultimately because the degree of an HFE map gives an upper bound on the Q-rank of its central map, has remained consistent, although rank attacks have progressed significantly at the end of NIST's previous call for encryption and signature schemes [28,2]. Finally, it must be noted that the description of HPPC uses the same tensor representation of quadratic forms as the one employed in the other candidate 3WISE [24], recently broken in [27].

Contributions. Although it is compared to an HFE scheme over $\mathbb{F}_2$ of maximal degree in the submission document, the HPPC scheme is actually constructed from a central polynomial of degree as small as $2^{10} + 1 = 1025$, in all parameter sets. Despite the unusual tensor representation, we show that all these parameter sets can be interpreted to have the structure of HFE over $\mathbb{F}_2$ with the same degree bound 1025. We further show that HPPC keys form an especially structured subclass of HFE keys, having Q-rank equal to 2 rather than 10, as would be expected for generic HFE keys with degree bound 1025.

As a result of this structure, we show that the polynomial system used in the direct attack exhibits degree fall polynomials of degree 3 and cannot be analyzed as a Boolean semi-regular system, as initially claimed. This analysis may be viewed as analogous to the results of [21,20,22] elucidating connections between properties of the extension field map and degree falls in the direct attack. Even though the study of degree falls in higher degrees seems more complicated, our tests suggest that the HPPC direct system is even easier to solve than a generic quadratic system with the same number of degree falls of degree 3 added.

More importantly, we show that all parameter sets can be practically broken with a key recovery attack based on a rank-2 MinRank problem. Our instance has matrices over $\mathbb{F}_2$ but solutions over $\mathbb{F}_{2^n}$, as in the rank attacks on HFE subsequent to [17], i.e., [6,28]. To solve this instance, we follow the strategy of [2], where an overdefined quadratic system in $n - 1$ variables is obtained from the Support-Minors modeling [4]. This method is slightly more efficient than relying on Kipnis-Shamir [17] or Minors, but the efficiency of our attack is mainly because the MinRank instance is not particularly hard. From the MinRank solutions, we finish the key recovery by solving linear equations. In that respect, the last step is much simpler than the one proposed by [6, §6.3.1] for even-rank HFE in characteristic 2. This is because our MinRank problem does not have the spurious solutions of [6, §6.3.1] (the same linear combinations produce matrices of rank 4 and not 2). For instance, our last step turns out to be much closer to that of [6] for HFE in odd characteristic, see [6, Theorem 9].

Navigation. The article is organized as follows. In Section 2, we establish notation for the paper, introduce the HPPC scheme, and discuss the relevant concepts for the attacks we develop. In Section 3, we present an analysis of a direct forgery attack on HPPC, proving that the claims of semi-regularity of the HPPC polynomial system are in error while extending the recent line of work on

the direct attack in application to big-field multivariate schemes. In Section 4, we establish the Q-rank property of HPPC that forms the basis for a rank attack and illustrate how the solution of the MinRank instance can be used to perform a full key recovery. Our solving method based on Support-Minors is described in Section 5, where we briefly establish its complexity and present the performance of our implementation.

2 Preliminaries

2.1 Field extension in HPPC

Let $\mathbb{F}_{2^n}$ be a degree-n extension of $\mathbb{F}_2$. The HPPC scheme constructs this extension explicitly as $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/\langle f(x)\rangle$, where $f \in \mathbb{F}_2[x]$ is an irreducible polynomial of degree n. Let $\phi : \mathbb{F}_2[x]/\langle f(x)\rangle \to \mathbb{F}_2^n$ be the isomorphism that maps a polynomial residue class to its coefficient representation as a column vector, i.e.,

$$\phi : \sum_{i=0}^{n-1} g_i x^i \mapsto \boldsymbol{g} = (g_0, \ldots, g_{n-1})^\mathsf{T}.$$

The canonical basis of $\mathbb{F}_2^n$, denoted by $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n)$, is such that $\boldsymbol{e}_i = \phi(x^{i-1})$ for all $i \in \{1..n\}$. Finally, let $\boldsymbol{C}_f \in \mathbb{F}_2^{n\times n}$ be the companion matrix of f. By definition, we have

$$\forall A \in \mathbb{F}_2[x]/\langle f(x)\rangle,\ \boldsymbol{C}_f \cdot \phi(A) = \phi(xA),$$

where the notation " $\cdot$ " refers to a matrix-vector product.

For convenience, we will refer to elements of the quotient ring $\mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle$ as polynomials throughout this article. A linearized polynomial is a polynomial that induces an $\mathbb{F}_2$-linear map $\mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$, i.e., all its monomials are of the form X^{2^j} for $j \in \{0..n-1\}$. A linearized permutation polynomial is a linearized polynomial that permutes $\mathbb{F}_{2^n}$, or equivalently, whose unique root in $\mathbb{F}_{2^n}$ is zero. The HPPC scheme heavily relies on matrix representations of linearized polynomials, as defined below.

Definition 1. *Let $\ell \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle$ be a linearized polynomial and let ϕ be a fixed $\mathbb{F}_2$-isomorphism between $\mathbb{F}_{2^n}$ and $\mathbb{F}_2^n$. We define the matrix $\boldsymbol{M}_\ell \in \mathbb{F}_2^{n\times n}$ as the unique matrix such that*

$$\forall A \in \mathbb{F}_{2^n},\ \boldsymbol{M}_\ell \cdot \phi(A) = \phi(\ell(A)).$$

This matrix is explicitly given by

$$\boldsymbol{M}_\ell = \left[\phi(\ell(1)) \ldots \phi(\ell(x^{n-1}))\right].$$

Conversely, we will sometimes write $\ell_{\boldsymbol{N}}$ for the linearized polynomial associated to the matrix $\boldsymbol{N} \in \mathbb{F}_2^{n\times n}$ by the equation above.

The core operation in HPPC is to realize the product between two elements in $\mathbb{F}_{2^n}$. For that purpose, the scheme considers the matrix

$$\boldsymbol{M} = \left[\boldsymbol{I}_n \; \boldsymbol{C}_f \; \ldots \; \boldsymbol{C}_f^{n-1}\right] \in \mathbb{F}_2^{n \times n^2}. \tag{1}$$

Lemma 1. *Let A, B represent arbitrary elements in $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/\langle f(x)\rangle$ and let $\boldsymbol{M} \in \mathbb{F}_2^{n \times n^2}$ be the matrix defined by Equation (1). We have*

$$\boldsymbol{M} \cdot \phi(A) \otimes \phi(B) = \phi(AB).$$

Proof. By $\mathbb{F}_q$-linearity, it is sufficient to prove the statement for $A = x^{i-1}$, $i \in \{1..n\}$ and for an arbitrary element $B \in \mathbb{F}_2[x]/\langle f(x)\rangle$. In this case, we have just seen that $\phi(A) = \boldsymbol{e}_i$, so that the vector $\phi(A) \otimes \phi(B)$ is the column vector of length n^2 with a non-zero block at position i equal to $\phi(B)$. Thus

$$\boldsymbol{M} \cdot \phi(A) \otimes \phi(B) = \boldsymbol{C}_f^{i-1} \cdot \phi(B) = \phi(x^{i-1}B) = \phi(AB),$$

where the first equality comes from the definition of $\boldsymbol{M}$ and the second one is by definition of the companion matrix. □

2.2 Polynomials

In this paper, the total degree (also referred to as the standard degree) of a polynomial $P \in \mathbb{F}_{2^n}[X]$ will be denoted by $\deg(P)$. We will also make use of the notion of "$\mathbb{F}_2$-degree", as named in [10, §2.1], although the terminology is not standard. For instance, the same concept is referred to as the "q-degree" in [12].

Definition 2 ($\mathbb{F}_2$-degree). *The $\mathbb{F}_2$-degree of a polynomial $P \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle$, denoted $\deg_{\mathbb{F}_2}(P)$, is defined as*

$$\deg_{\mathbb{F}_2}(P) = \max\{\mathrm{wt}_2(e) \mid X^e \textit{ appears in } P \textit{ with a non-zero coefficient}\},$$

where $\mathrm{wt}_2(e)$ is the Hamming weight of the base-2 representation of e.

In the context of big-field schemes, the public key or the secret key can be represented either as a univariate polynomial in $\mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle$ (the *univariate representation*), or as a set of Boolean multivariate polynomials (the *multivariate representation*). This latter view is enabled by the following isomorphism derived from ϕ:

$$\mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle \to (\mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2 - x_1, \ldots, x_n^2 - x_n\rangle)^n. \tag{2}$$

This map induces a bijection between univariate polynomials P over $\mathbb{F}_{2^n}$ of $\mathbb{F}_2$-degree d and n-tuples of multivariate polynomials p_i over $\mathbb{F}_2$ of degree d, and we refer to [12, §4] for more details.

We will consider polar forms of multivariate homogeneous quadratic polynomials in characteristic 2, either over $\mathbb{F}_2$ or over $\mathbb{F}_{2^n}$. The definition of the polar form in this case is recalled below.

Definition 3 (Polar form). *Let q be a homogeneous quadratic polynomial in characteristic 2. The polar form of q is the bilinear map q' defined by*

$$q'(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x} + \boldsymbol{y}) + q(\boldsymbol{x}) + q(\boldsymbol{y}).$$

The main difference with the odd characteristic case is that there is no longer 1-to-1 correspondence between quadratic forms and symmetric bilinear forms in characteristic 2.

We will finally use the notion of Q-rank, whose relevance will become clearer when discussing rank attacks. An explicit definition of Q-rank can be found in [11, Definition 2], although this concept was already implicit in earlier works – for example, in [16] and [12, §4.4]. In this paper, we adopt the following formulation.

Definition 4 (Q-rank). *Let $P(X) \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle$ be a polynomial whose monomials are all of $\mathbb{F}_2$-degree 2 and let $Q(P) \in \mathbb{F}_{2^n}[X_0, \ldots, X_{n-1}]$ be the quadratic form obtained by setting $X_0 = X, X_1 = X^2, \ldots, X_{n-1} = X^{2^{n-1}}$ in this polynomial. The Q-rank of P is defined as the rank of the skew-symmetric matrix representing the polar form of $Q(P)$.*

2.3 Description of HPPC

This section gives the multivariate and the univariate representation of the HPPC scheme, and we refer to [25,26] for a more detailed description. We also include a discussion on equivalent keys. This concept was not discussed in [25,26] but is common in the analysis of big-field multivariate schemes.

Multivariate representation. The HPPC public key consists of n homogeneous quadratic polynomials $(p_1, \ldots, p_n)$, where each p_i belongs to $\mathbb{F}_2[\boldsymbol{x}] = \mathbb{F}_2[x_1, \ldots, x_n]$. In the following, we will write $\mathcal{P}(\boldsymbol{x})$ for the column vector in $\mathbb{F}_2[\boldsymbol{x}]^n$ whose entries correspond to these polynomials. In other words, there is a public matrix $\boldsymbol{A} \in \mathbb{F}_2^{n \times n^2}$ such that $\mathcal{P}(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{x} \otimes \boldsymbol{x})$. The private key contains the following data:

- two invertible matrices $\boldsymbol{S}$ and $\boldsymbol{T}$ in $\mathbb{F}_2^{n \times n}$;
- two invertible matrices $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ in $\mathbb{F}_2^{n \times n}$ that represent linearized permutation polynomials. The matrix $\boldsymbol{L}_1 = \boldsymbol{M}_{\ell_1} \in \mathbb{F}_2^{n \times n}$ represents an arbitrary linearized permutation polynomial ℓ_1 while $\boldsymbol{L}_2 = \boldsymbol{M}_{\ell_2} \in \mathbb{F}_2^{n \times n}$ is picked as the representation of a monic linearized permutation polynomial ℓ_2 with maximal exponent 2^d;
- the matrix $\boldsymbol{M} \in \mathbb{F}_2^{n \times n^2}$ defined by Equation (1).

The public key is obtained by the composition of these private components, namely

$$\mathcal{P}(\boldsymbol{x}) = \boldsymbol{T}\boldsymbol{M}(\boldsymbol{L}_1 \otimes \boldsymbol{L}_2\boldsymbol{L}_1)(\boldsymbol{S} \otimes \boldsymbol{S})(\boldsymbol{x} \otimes \boldsymbol{x}).$$

For all security levels, the value of d used in the definition of ℓ_2 is chosen as equal to 10. The provided parameter sets simply differ according to the value of n, i.e., $n = 128$ for HPPC128, $n = 192$ for HPPC192, and $n = 256$ for HPPC256.

Univariate representation. To interpret all operations over $\mathbb{F}_{2^n}$, we fix $X = \phi^{-1}(\boldsymbol{x}) \in \mathbb{F}_{2^n}$. Using the above notation, there exists a linearized polynomial $\ell_{\boldsymbol{S}}$ such that $\boldsymbol{S}\boldsymbol{x} = \phi(\ell_{\boldsymbol{S}}(X))$. Then, we see that $\boldsymbol{L}_1\boldsymbol{S}\boldsymbol{x} = \phi(\ell_1 \circ \ell_{\boldsymbol{S}}(X))$ and $\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{S}\boldsymbol{x} = \phi(\ell_2 \circ \ell_1 \circ \ell_{\boldsymbol{S}}(X))$. From there, Lemma 1 applied to $A = \ell_1 \circ \ell_{\boldsymbol{S}}(X)$ and $B = \ell_2 \circ \ell_1 \circ \ell_{\boldsymbol{S}}(X)$ shows that

$$\mathcal{P}(\boldsymbol{x}) = \boldsymbol{T} \circ \phi \circ (F \circ \ell_{\boldsymbol{S}}(X)) = \left(\boldsymbol{T} \circ \phi \circ F \circ \phi^{-1} \circ \boldsymbol{S}\right)(\boldsymbol{x}),$$

where $F(X) \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle$ is the product $\ell_1 \times (\ell_2 \circ \ell_1)$.

Signing with the private key boils down to computing the roots of a polynomial of the form $G - Z$, where the scalar $Z \in \mathbb{F}_{2^n}$ is related to the message and where $G(X) = X \times \ell_2(X)$. This operation can be done efficiently because this polynomial is of small standard degree $2^d + 1$. In the following, the term "central polynomial" will mostly refer to the polynomial G. We have

$$\begin{aligned}\mathcal{P}(\boldsymbol{x}) &= \left(\boldsymbol{T} \circ \phi \circ G \circ \ell_1 \circ \phi^{-1} \circ \boldsymbol{S}\right)(\boldsymbol{x}) = \left(\boldsymbol{T} \circ \phi \circ G \circ \phi^{-1} \circ \boldsymbol{L}_1\boldsymbol{S}\right)(\boldsymbol{x}) \\ &= \left(\boldsymbol{T}' \circ \phi \circ G \circ \phi^{-1} \circ \boldsymbol{S}'\right)(\boldsymbol{x}), \end{aligned} \tag{3}$$

where $\boldsymbol{S}' = \boldsymbol{L}_1\boldsymbol{S}$ and $\boldsymbol{T}' = \boldsymbol{T}$.

Equivalent keys. Equivalent keys are a standard concept in the cryptanalysis of multivariate schemes. Two private keys are said to be equivalent if they correspond to the same public key. In the context of HPPC, equivalent keys will contain a univariate polynomial $\widetilde{G}$ that has the same "shape" – see [30] for the origin of this terminology – as the genuine central polynomial G, i.e., whose standard degree is as small as that of G. Since the construction of the scheme and the constraints on an equivalent central map are similar to those in HFE, we will estimate the number of HPPC equivalent keys by [30, Theorem 4.2] which we restate (adapted to our notation) here:

Theorem 1. *(Theorem 4.2 in [30]) For $(\boldsymbol{S}, F, \boldsymbol{T}) \in \mathbb{F}_2^{n\times n} \times \mathbb{F}_{2^n}[X] \times \mathbb{F}_2^{n\times n}$ an HFE private key, we have*

$$n \cdot 2^{2n}(2^n - 1)^2$$

equivalent private keys.

This result will allow us to keep the same degrees of freedom as in the HFE case for our key recovery attack.

2.4 Attacks on big-field schemes

We study both the direct attack and rank attacks on HPPC. The direct attack is an algebraic attack that is generic to any multivariate scheme. In the context of digital signatures, the direct attack is used for forgery. In contrast, rank attacks are scheme-specific key recovery attacks. Both attacks employ Gröbner basis algorithms as a subroutine.

Direct attack. Direct attacks attempt to solve the affine system $\mathcal{P}(\boldsymbol{x}) = \boldsymbol{t}$, where $\mathcal{P}$ is the public key and $\boldsymbol{t}$ is a public vector related to the message, by employing Gröbner basis algorithms. A critical concept in this context is that of degree fall polynomials, as this notion is often used to estimate the complexity of Gröbner basis methods.

Definition 5 (Degree fall, degree fall polynomial). *A degree fall on a set of polynomials $\{f_1, \ldots, f_m\} \subset \mathbb{F}_q[x_1, \ldots, x_n]$ is a relation of the form*

$$\sum_{i=1}^{n} g_i f_i = h,$$

for which $deg(h) < \max_i (deg(g_i f_i))$. The polynomial h in such a relation will be called a degree fall polynomial and we will say that it is obtained as a degree fall from degree $\max_i (deg(g_i f_i))$ to degree $deg(h)$.

It is easy to see that degree falls are the relations in $\mathbb{F}_q[x_1, \ldots, x_n]$ corresponding to syzygies in the graded module generated by the homogeneous components of the f_i's of highest degree. In particular, Definition 5 includes degree fall polynomials that correspond to trivial syzygies in this module. However, these degree falls play no role in the complexity analysis. In Proposition 1 below, the term "'non-trivial" will refer to degree falls that do not correspond to trivial syzygies. We caution the reader that, in certain exceptional cases, such degree falls may still be redundant for the computation.

Early attempts to estimate the complexity of solving the direct system $\mathcal{P}(\boldsymbol{x}) = \boldsymbol{t}$ in the case of HFE tried to grasp the behavior of the Gröbner basis algorithm from certain (univariate) polynomial combinations over the extension field with low $\mathbb{F}_2$-degree. This method was initiated in [16, §4] and formalized in [12, §4]. Since then, the same type of analysis has been applied to other big-field schemes [21,20,22]. The key observation is that, via the isomorphism defined in Equation (2), two-powerings P^{2^j} correspond to n-tuples of linear combinations of the p_i's, and that any product $H(X)P(X)$ corresponds to n polynomial combinations (indexed by j) of the form $\sum_i h_{i,j} p_i$, where $\deg(h_{i,j}) = \deg_{\mathbb{F}_2}(H)$. In Section 3, we will use the following lemma.

Lemma 2 (Lemma 1 in [22]). *Let $P(X) \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle$ be the univariate representation of the public key of a big-field scheme, and let $H(X) \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} - X\rangle$ be such that*

$$\deg_{\mathbb{F}_2}(HP) < \deg_{\mathbb{F}_2}(H) + \deg_{\mathbb{F}_2}(P).$$

Then, in the public system $\phi(P) = (p_1, \ldots, p_n)$, there are n degree falls from degree $\deg_{\mathbb{F}_2}(H) + \deg_{\mathbb{F}_2}(P)$ to degree $\deg_{\mathbb{F}_2}(HP)$.

Rank attacks. In these attacks, the main step is to solve an instance of the MinRank problem [8]. The MinRank problem consists in finding a linear combination, over a finite field, of a given collection of matrices such that the resulting

matrix has rank at most r, the specified target rank. In the context of big-field schemes, this target rank is often related to the notion of Q-rank, as given in Definition 4. In the case of HFE without any modifiers, the central polynomial with standard degree D can be represented by a skew-symmetric matrix whose unique non-zero block lies in the top-left corner and has size $d \times d$, with $d \leq \lceil \log_2(D) \rceil$. As a result, the Q-rank is upper bounded by $\lceil \log_2(D) \rceil$, and this bound has been adopted as the target rank in rank attacks. In particular, in HPPC, the central polynomial is designed to mimic an HFE polynomial with maximal Q-rank, with the hope of resisting MinRank attacks.

3 Direct attack

As been explained above, a sequence of recent results [21,20,22] has been able to tie algebraic relations among univariate polynomials over the extension field to degree fall relations in the Gröbner basis computation on the system over the small field that is used for forgery. This line of work can be viewed as the natural extension of the analysis of the direct attack on HFE of [16,12] in the context of lower Q-rank maps, see [9,18]. We found that these techniques may be directly applied to HPPC. Even though the direct attack still seems less efficient than the practical key recovery attack that we present afterward, it is important to note its existence, especially when considering the initial security claims.

The HPPC submission document [25] seems to indicate that the HPPC public key equations behave as a semi-regular system based on experiments in SageMath. Before moving on to our analysis, we want to point out some inaccuracies of [25, §7] regarding the notion of semi-regularity. A first issue is that the submission treats the field equations $x_i^2 + x_i = 0$ as an extra set of n equations added to the system $\mathcal{P}(\boldsymbol{x}) = \boldsymbol{t}$ and adopts the generating series $(1-t^2)^{2n}/(1-t)^n$ for the whole, see [25, §7.1.3]. This implicitly assumes that the field equations have a generic behavior, which is not true. For instance, the correct generating series for a Boolean semi-regular system with n quadratic equations in n variables is $(1+t)^n/(1+t^2)^n$ [5]. At this stage, we stress that the HPPC system has no reason to behave as such. A second issue is related to the SageMath experiments. To estimate the degree at which the system is solved, [25, §7.1.3] computes d_{reg}, the index of the first non-positive coefficient of the generating series. In [25, §7.2], this theoretical value is compared to an experimental degree d_{exp} obtained using the `degree_of_semi_regularity()` command. However, this command *assumes* that the input equations are semi-regular, which means that d_{exp} is not the result of an experiment. In particular, it is not a surprise that $d_{\text{exp}} = d_{\text{reg}}$, but this equality does not tell anything about the behavior of the equations.

It turns out that the public equations of HPPC cannot be analyzed as a semi-regular system, even a Boolean one. In the next subsection, we exhibit degree fall polynomials that tend to suggest that these public equations are much easier to solve than comparable semi-regular equations.

3.1 Degree fall polynomials

Our partial analysis of degree fall polynomials is presented in Propositions 1 and 2 below, which are stated with Faugère's F_4 algorithm [14] in mind. This algorithm proceeds through a sequence of steps, each processing a selected set of polynomial pairs. As F_4 follows the so-called normal strategy, which always selects pairs of minimal degree, degree fall polynomials are incorporated into the next step immediately upon discovery.

Proposition 1 *There are $2n$ non-trivial degree falls from degree 3 to degree 2 at the first step in degree 3.*

Proof. Without loss of generality, we can apply Lemma 2 to $F = \ell_1 \times (\ell_2 \circ \ell_1)$ instead of $P = \ell_T \circ F \circ \ell_S$ because the composition with ℓ_S and ℓ_T does not affect the degree of the polynomials. Then, in characteristic 2, both ℓ_1^2 and $(\ell_2 \circ \ell_1)^2$ are linearized polynomials. Therefore, we can use Lemma 2 twice with $H = \ell_1$ and $H = \ell_2 \circ \ell_1$ respectively, which yields $n + n = 2n$ degree fall polynomials from degree $3 = 1 + 2$ to degree 2. □

These initial degree falls trigger another set of $2n$ degree falls at the next step of the algorithm that we can still understand.

Proposition 2 *There are $2n$ degree falls from degree 3 to degree 2 at the second step in degree 3.*

Proof. We apply the same reasoning to $G_1 = \ell_1^2 \times (\ell_2 \circ \ell_1)$ and $G_2 = \ell_1 \times (\ell_2 \circ \ell_1)^2$ which are the result of the first step. Then, as above, the polynomials $\ell_1^2 G_1$ and $(\ell_2 \circ \ell_1)^2 G_2$ are products of one polynomial of $\mathbb{F}_2$-degree 1 and one of $\mathbb{F}_2$-degree 2 but they have $\mathbb{F}_2$-degree 2. Lemma 2 then gives $n + n = 2n$ degree falls. Finally, note that the polynomial $(\ell_2 \circ \ell_1) \times G_1 = \ell_1 \times G_2$ is also of $\mathbb{F}_2$-degree 2. However, as it is equal to F^2, it will trigger polynomials that are redundant to the computation[1]. □

Remark 1 (Dependency with respect to d) *The number of linearly independent degree falls at step 2 degree 3 may differ from the number of degree falls predicted by Proposition 2 if the value of d is much smaller than 10. Intuitively, we cannot square the polynomial ℓ_1 endlessly because we will get redundancy, i.e., extra linear relations, coming from the polynomial $\ell_2 \circ \ell_1$.*

3.2 Experiments on the direct system

We present the earliest steps of the F_4 algorithm and the associated degree fall polynomials on the direct system for several values of d and n in Appendix A. In our tests, we always ensure that the equations have a solution. The last step of

[1] Such polynomials would be counted as degree falls according to Lemma 2, since the definition used in [22], from which the lemma is taken, does not take into account their usefulness.

the algorithm that we give yields sufficiently many degree 1 polynomials so that Magma's default F_4 restarts from there and the subsequent steps are cheaper.

We note that Propositions 1 and 2 accurately predict the number of degree falls in degree 3 observed in our experiments. However, estimating the number of degree falls in higher degrees appears to be more complex. In particular, this quantity seems to depend not only on the degree d but also on the specific monomials present in ℓ_2. Our results correspond to a dense linearized polynomial ℓ_2 of standard degree 2^d, but different behavior was observed when using a sparser polynomial. We have not pursued this estimation further, as rank attacks seemed more efficient at this stage of our work. Still, we want to mention that a deeper analysis of such degree fall polynomials in higher degrees has already led to better attacks on other big-field schemes [21,20,22].

The rest of the paper presents our efficient rank attack to recover an equivalent key. In Section 4, we give the MinRank problem that we consider and we show how a solution to this problem can be used to complete the attack. In Section 5, we solve this MinRank problem using the Support-Minors modeling.

4 Rank attack

The simple observation behind our rank attack is that the Q-rank of the HPPC polynomial $F = \ell_1 \times (\ell_2 \circ \ell_1)$ is equal to 2, regardless of the value of d. Indeed, both ℓ_1 and $\ell_2 \circ \ell_1$ are linear forms in $\mathbb{F}_{2^n}[X_0, \ldots, X_{n-1}]$ with the identification of Definition 4, and the quadratic form $Q(F)$ is the product of these two linear forms. This leads us to considering a rank-2 MinRank problem that is pretty similar to the one proposed by [6] to cryptanalyze HFE.

Our description will closely follow [6]. Let $(\theta_1, \ldots, \theta_n)$ be an $\mathbb{F}_2$-basis of $\mathbb{F}_{2^n}$ and let $\boldsymbol{M}_n \in \mathbb{F}_{2^n}^{n \times n}$ be the Moore matrix defined by

$$\boldsymbol{M}_n = \begin{bmatrix} \theta_1 & \theta_1^2 & \ldots & \theta_1^{2^{n-1}} \\ \theta_2 & \theta_2^2 & \ldots & \theta_2^{2^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_n & \theta_n^2 & \ldots & \theta_n^{2^{n-1}} \end{bmatrix}. \tag{4}$$

We consider the more convenient isomorphism $\psi : \mathbb{F}_{2^n} \to \mathbb{F}_2^n$ given by this matrix, namely

$$\begin{array}{rccc} & \mathbb{F}_{2^n} & \to & \mathbb{F}_2^n \\ \psi : & X & \mapsto & (\boldsymbol{M}_n^{-1})^{\mathsf{T}} \begin{bmatrix} X \\ \vdots \\ X^{2^{n-1}} \end{bmatrix}. \end{array}$$

Its inverse is given by

$$\begin{array}{rccc} & \mathbb{F}_2^n & \to & \mathbb{F}_{2^n} \\ \psi^{-1} : & \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} & \mapsto & \left((\boldsymbol{M}_n)^{\mathsf{T}} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \right) [1], \end{array}$$

where $\boldsymbol{v}[1]$ refers to the top coordinate of the column vector $\boldsymbol{v}$ [6, Proposition 2]. Coming back to HPPC, there exists an invertible matrix $\boldsymbol{N} \in \mathbb{F}_2^{n\times n}$ such that the isomorphism ϕ used in the scheme is $\phi = \boldsymbol{N} \circ \psi$. The expression of $\boldsymbol{N}$ depends on the polynomial f appearing in the definition of ϕ but we will not need it in our attack. Moreover, we see in the univariate representation of the public key given by Equation (3) that the $\boldsymbol{L}_1$ matrix can be "absorbed" into $\boldsymbol{S}'$. Therefore, we may proceed as if $\boldsymbol{S}'$ and $\boldsymbol{T}'$ were random invertible matrices and reason on the equation

$$\begin{aligned}\mathcal{P} &= \boldsymbol{T}'\boldsymbol{N} \circ \psi \circ G \circ \psi^{-1} \circ \boldsymbol{N}^{-1}\boldsymbol{S}' \\ &= \overline{\boldsymbol{T}} \circ \psi \circ G \circ \psi^{-1} \circ \overline{\boldsymbol{S}},\end{aligned}$$

where $\overline{\boldsymbol{S}} = \boldsymbol{N}^{-1}\boldsymbol{S}'$ and $\overline{\boldsymbol{T}} = \boldsymbol{T}'\boldsymbol{N}$. From this new equation, the rest follows as in [6]. We have

$$\begin{aligned}\begin{bmatrix} p_1(\boldsymbol{v}) \\ \vdots \\ p_n(\boldsymbol{v}) \end{bmatrix} = \overline{\boldsymbol{T}}(\boldsymbol{M}_n^{-1})^{\mathsf{T}} \begin{bmatrix} G^{2^0}(\psi^{-1}(\overline{\boldsymbol{S}}\boldsymbol{v})) \\ \vdots \\ G^{2^{n-1}}(\psi^{-1}(\overline{\boldsymbol{S}}\boldsymbol{v})) \end{bmatrix} &= \overline{\boldsymbol{T}}(\boldsymbol{M}_n^{-1})^{\mathsf{T}} \begin{bmatrix} \boldsymbol{v}^{\mathsf{T}}\overline{\boldsymbol{S}}^{\mathsf{T}}\boldsymbol{M}_n\boldsymbol{G}^{*0}\boldsymbol{M}_n^{\mathsf{T}}\overline{\boldsymbol{S}}\boldsymbol{v} \\ \vdots \\ \boldsymbol{v}^{\mathsf{T}}\overline{\boldsymbol{S}}^{\mathsf{T}}\boldsymbol{M}_n\boldsymbol{G}^{*n-1}\boldsymbol{M}_n^{\mathsf{T}}\overline{\boldsymbol{S}}\boldsymbol{v} \end{bmatrix} \\ &= \overline{\boldsymbol{T}}(\boldsymbol{M}_n^{-1})^{\mathsf{T}} \begin{bmatrix} \boldsymbol{v}^{\mathsf{T}}\overline{\boldsymbol{S}}^{\mathsf{T}}\boldsymbol{M}_n\boldsymbol{G}^{*0}\boldsymbol{M}_n^{\mathsf{T}}\overline{\boldsymbol{S}}\boldsymbol{v} \\ \vdots \\ \boldsymbol{v}^{\mathsf{T}}\overline{\boldsymbol{S}}^{\mathsf{T}}\boldsymbol{M}_n\boldsymbol{G}^{*n-1}\boldsymbol{M}_n^{\mathsf{T}}\overline{\boldsymbol{S}}\boldsymbol{v} \end{bmatrix},\end{aligned}$$

where $\boldsymbol{G}^{*i}$ is at this stage any matrix that represents the quadratic form $Q(G^{2^i})$ for $i \in \{0..n-1\}$. This matrix is not unique and to proceed further we need to consider symmetric representations (or equivalently skew-symmetric ones in the even characteristic case). As in [17,6], we take $\boldsymbol{A}+\boldsymbol{A}^{\mathsf{T}}$ instead of $\frac{\boldsymbol{A}+\boldsymbol{A}^{\mathsf{T}}}{2}$ since we are in characteristic 2 (this is basically the matrix representing the polar form of the quadratic form). Assuming such a representation, the relevant matrix $\boldsymbol{G}^{*0}$ is given in Lemma 3.

Lemma 3. *The skew-symmetric matrix $\boldsymbol{G}^{*0}$ such that $Q(G)(X,\dots,X^{2^{n-1}}) = (X,\dots,X^{2^{n-1}})\boldsymbol{G}^{*0}(X,\dots,X^{2^{n-1}})^{\mathsf{T}}$ can be defined from its first row, which contains non-zero entries only in position $(1,j)$ for $j \in \{2..d+1\}$ and such that the entry in position $(1,d+1)$ is equal to 1.*

Proof. Recall that $G(X) = X \times \ell_2(X)$, where $\ell_2 = \sum_{j=0}^{d-1} c_i X^{2^i} + X^{2^d}$ is a monic linearized polynomial of standard degree 2^d. Thus, the quadratic form

$$Q(G) = \textstyle\sum_{1\le i\le j\le n} b_{i,j} X^{2^{i-1}} X^{2^{j-1}}$$

is such that $b_{1,j} = c_{j-1}$ for $j \in \{1..d\}$, $b_{1,d+1} = 1$ and $b_{i,j} = 0$ if $i \ge 2$ and/or $j \ge d+2$. The expression of $\boldsymbol{G}^{*0}$ can be easily deduced from that of $Q(G)$. □

If we now let $\boldsymbol{G}^{*0} = (g_{i,j})_{i,j}$, where row and columns indexes range from 0 to $n-1$, we have $\boldsymbol{G}^{*k} = (g^{2^k}_{i-k,j-k})_{i,j}$ for any $k \in \{0..n-1\}$, where all differences

$i-k$ or $j-k$ must be understood modulo n (see for example [28, Proposition 1]). In the following, we will write Frob_k for the $\mathbb{F}_q$-linear map that maps the matrix with entries $m_{i,j}$ to the one with entries $m_{i-k,j-k}^{2^k}$, so that $\boldsymbol{G}^{*k} = \mathrm{Frob}_k(\boldsymbol{G}^{*0})$. Finally, we set $\boldsymbol{U} = (\overline{\boldsymbol{T}}^{\mathsf{T}})^{-1}\boldsymbol{M}_n$ and $\boldsymbol{W} = \overline{\boldsymbol{S}}^{\mathsf{T}}\boldsymbol{M}_n$. Overall, we obtain

$$\left[\boldsymbol{P}_1 \dots \boldsymbol{P}_n\right](\boldsymbol{U} \otimes \boldsymbol{I}_n) = \left[\boldsymbol{W}\boldsymbol{G}^{*0}\boldsymbol{W}^{\mathsf{T}} \ \dots \ \boldsymbol{W}\boldsymbol{G}^{*n-1}\boldsymbol{W}^{\mathsf{T}}\right]. \tag{5}$$

Here we use the fact that the matrices $\boldsymbol{W}\boldsymbol{G}^{*k}\boldsymbol{W}^{\mathsf{T}}$ all have rank 2. As in [6], we will retrieve one column of a matrix denoted by $\widetilde{\boldsymbol{U}}$ that plays the same role as $\boldsymbol{U}$ in Equation (5) but which is part of a possibly different equivalent key by solving a rank-2 MinRank problem with matrices over $\mathbb{F}_2$ but solutions over $\mathbb{F}_{2^n}$.

Our approach is to solve MinRank first to find such a $\widetilde{\boldsymbol{U}}$ and then to solve linear equations to recover the rest of the private key components. This method is similar to the ones proposed by [17,6] to attack HFE in odd characteristic. In particular, it highly differs from the key recovery technique devised in [6, §6.3.1] for HFE in characteristic 2. In [6, §6.3.1], the two attack steps could not be separated. There, MinRank was solved at the same time as retrieving the rest of the private key, which is both less natural and less efficient.

4.1 Recovering $\widetilde{\boldsymbol{U}}$

From Equation (5), we obtain

Proposition 3 *Let $\boldsymbol{U} \in \mathbb{F}_{2^n}^{n\times n}$ be the secret matrix $(\overline{\boldsymbol{T}}^{\mathsf{T}})^{-1}\boldsymbol{M}_n$. The columns of $\boldsymbol{U}$ are solutions (over $\mathbb{F}_{2^n}$) to the MinRank problem defined by the matrices $\boldsymbol{P}_i \in \mathbb{F}_2^{n\times n}$ for $i \in \{1..n\}$, with target rank 2.*

In Section 5, MinRank will be solved using the Support-Minors modeling. Before moving on, let us mention that the instance does not admit spurious solutions comparable to those of the form $\lambda\boldsymbol{G}^{*i} + \mu\boldsymbol{G}^{*i+1}$ for even rank HFE (we refer to [15] and [6, §6.3] for more details). Indeed, such matrices are here of expected rank 4 so that they are not problematic (this can be easily seen from the expression of $\boldsymbol{G}^{*i}$ that may be derived from Lemma 3). This is what explains that our attack method is close to the one employed for HFE in odd characteristic and deviates from the one of [6, §6.3.1] for HFE in characteristic 2.

The next subsection details how to finish the key recovery from an arbitrary rank 2 solution $\boldsymbol{Z} = \sum_{i=1}^{n} u_i\boldsymbol{P}_i \in \mathbb{F}_{2^n}^{n\times n}$. This solution is expected to correspond to a unique Frobenius power of G and thus to a column of a matrix $\widetilde{\boldsymbol{U}}$ equivalent to $\boldsymbol{U}$. Finally, it is standard how to reconstruct the full matrix $\widetilde{\boldsymbol{U}}$ from such a column, using [6, Proposition 6] or equivalently the special form of $\boldsymbol{M}_n$.

4.2 Recovering the rest of the equivalent key

As announced above, we will recover the rest of the private key components by solving linear equations. We consider the permutation matrix $\boldsymbol{P} = \begin{bmatrix}\boldsymbol{0}_{n-1} & 1\\ \boldsymbol{I}_{n-1} & 0\end{bmatrix}$.

For an arbitrary matrix $\boldsymbol{A} \in \mathbb{F}_{2^n}^{n \times n}$ and for any index $k \in \{0..n-1\}$, we have

$$\mathrm{Frob}_k(\boldsymbol{A}) = \boldsymbol{P}^k \boldsymbol{A}^{[k]} \boldsymbol{P}^{-k} = \boldsymbol{P}^k \boldsymbol{A}^{[k]} (\boldsymbol{P}^k)^{\mathsf{T}},$$

where the matrix $\boldsymbol{A}^{[k]}$ is obtained from $\boldsymbol{A}$ by raising each entry to the power 2^k.

Coming back to Equation (5), we will assume without loss of generality and to clarify the description that our rank 2 matrix $\boldsymbol{Z}$ is equal to $\boldsymbol{W}\boldsymbol{G}^{*0}\boldsymbol{W}^{\mathsf{T}}$. For any index $k \in \{0..n-1\}$, we have $\boldsymbol{W}^{[k]} = \boldsymbol{W}\boldsymbol{P}^k$ by definition of $\boldsymbol{W}$ and by the special shape of $\boldsymbol{M}_n$. Therefore,

$$\begin{aligned}
\boldsymbol{Z}^{[k]} &= \boldsymbol{W}^{[k]} \left(\boldsymbol{G}^{*0}\right)^{[k]} (\boldsymbol{W}^{-1})^{[k]} \\
&= \boldsymbol{W}\boldsymbol{P}^k \left(\boldsymbol{G}^{*0}\right)^{[k]} \boldsymbol{P}^{-k}\boldsymbol{W}^{-1} \\
&= \boldsymbol{W}\mathrm{Frob}_k(\boldsymbol{G}^{*0})\boldsymbol{W}^{-1}.
\end{aligned}$$

We will use this equality for $k \in \{1..n-d-1\}$ to derive a linear system in the n variables that constitute the first row $\boldsymbol{w}$ of $\boldsymbol{W}^{-1}$. Indeed, for such an index k, the first row of the matrix $\mathrm{Frob}_k(\boldsymbol{G}^{*0})$ and thus the one of $\mathrm{Frob}_k(\boldsymbol{G}^{*0})\boldsymbol{W}^{-1}$ is the zero vector. This follows once again from Lemma 3 and then from the expression of $\mathrm{Frob}_k(\boldsymbol{G}^{*0}) = \boldsymbol{G}^{*k} = (g_{i-k,j-k}^{2^k})_{i,j}$ in function of $\boldsymbol{G}^{*0} = (g_{i,j})_{i,j}$. For each index k, the equality $\boldsymbol{w}\boldsymbol{Z}^{[k]} = \boldsymbol{0}_n$ gives n equations but only two of them are linearly independent since the rank of $\boldsymbol{Z}^{[k]}$ is 2. Overall, we get at most $2(n-d-1)$ linearly independent equations by using all the k indices.

Note that any vector proportional to the first row of $\boldsymbol{W}^{-1}$ will also yield solutions. If the parameters n and d satisfy $n \geq 2d+1$, which is the case in HPPC, these will be the only solutions. Thus, we solve the equations by fixing one variable to eventually recover a matrix $\widetilde{\boldsymbol{W}}$ equivalent to $\boldsymbol{W}$. Finally, we complete the key recovery by computing the matrix $\widetilde{\boldsymbol{G}^{*0}} = \widetilde{\boldsymbol{W}}^{-1}\boldsymbol{Z}\widetilde{\boldsymbol{W}}$. This matrix is necessarily associated with a central polynomial of HPPC shape.

5 Solving the MinRank instance with Support-Minors

In this section, we solve the MinRank problem presented in Proposition 3. First, as in the HFE case, we are still able to recover an equivalent key by fixing one variable. For instance, we will fix the last variable to 1 and target a rank 2 matrix of the form

$$\boldsymbol{Z} = \textstyle\sum_{i=1}^{n-1} u_i \boldsymbol{P}_i + \boldsymbol{P}_n,$$

where the u_i's are the remaining variables. Since the $\boldsymbol{P}_i$'s have entries in $\mathbb{F}_2$, we expect essentially n solutions in $\mathbb{F}_{2^n}^{n-1}$ obtained from one another by applying the Frobenius map on each component.

Standard algebraic modelings to solve the MinRank problem are the Minors modeling, the Kipnis-Shamir modeling, and the Support-Minors (SM) modeling. We will present a solving method based on Support-Minors. We obtained similar results with Kipnis-Shamir, which should not be surprising given the proximity between the two modelings [3].

We will now briefly describe the SM system. The starting point is a factorization $\boldsymbol{Z} = \boldsymbol{D}\boldsymbol{C}$ with unknown full-rank factors $\boldsymbol{D} \in \mathbb{F}_{2^n}^{n\times 2}$ and $\boldsymbol{C} \in \mathbb{F}_{2^n}^{2\times n}$. For $\ell \in \{1..n\}$, let $\boldsymbol{r}_\ell$ be the ℓ-th row of $\boldsymbol{Z}$ viewed as a vector of degree 1 forms in the u_i variables and let

$$\boldsymbol{C}_\ell = \begin{bmatrix} \boldsymbol{r}_\ell \\ \boldsymbol{C} \end{bmatrix}.$$

The SM modeling follows from the fact that the rank of $\boldsymbol{C}_\ell$ is at most 2 because the vector $\boldsymbol{r}_\ell$ belongs to the row space of $\boldsymbol{C}$. More precisely, the relevant system contains the set of all 3×3 minors of $\boldsymbol{C}_\ell$ for $\ell \in \{1..n\}$, i.e.,

$$\mathcal{F}_{\mathrm{SM}} = \{|\boldsymbol{C}_\ell|_{*,J}\,,\ J \subset \{1..n\},\ \#J = 3,\ \ell \in \{1..n\}\}. \tag{6}$$

By Laplace expansion along the first row of $\boldsymbol{C}_\ell$, it is known that all the equations of $\mathcal{F}_{\mathrm{SM}}$ are bilinear in the u_i's and in the 2×2 minors of $\boldsymbol{C}$. The crux was in fact to consider the latter as new variables $c_T = |\boldsymbol{C}|_{*,T}$ for $T \subset \{1..n\},\ \#T = 2$, called minor variables.

Modeling 1 (Support-Minors) *The Support-Minors modeling is the system obtained from Equation (6) by Laplace expansion along the first row of the matrices and by setting as new unknowns $c_T = |\boldsymbol{C}|_{*,T}$ for $T \subset \{1..n\},\ \#T = 2$. The system is affine bilinear in the u_i's and in these new minor variables.*

As in previous works, see [4,7,28,2], it will be convenient to restrict ourselves to SM subsystems. This is done by considering the minors constructed from only $\kappa \leq n$ columns of the $\boldsymbol{C}_\ell$ matrices. We still want the $\boldsymbol{C}_\ell$ submatrices to be of full-rank 2 so we include the 2 columns of the identity block $\boldsymbol{I}_2$ that is usually part of $\boldsymbol{C}$. In this way we obtain an SM subsystem containing $n\binom{\kappa}{3}$ affine bilinear equations in $n_u = n-1$ linear variables u_i but only $n_{c_T} = \binom{\kappa}{2} - 1$ minor variables (we fix to 1 the 2×2 minor which is the determinant of $\boldsymbol{I}_2$ exactly as in [4,2]).

5.1 "Big-field" Support-Minors

Our method is to solve a simpler quadratic system in the u_i variables derived from the (affine) Support-Minors modeling, following the strategy presented in [2] for the big-field MinRank problem of Tao, Petzoldt and Ding [28]. Our description will entirely follow [2]. Here, the total number of monomials in SM is equal to

$$n_u n_{c_T} + n_u + n_{c_T} + 1 = (n_u + 1)(n_{c_T} + 1) = n\tbinom{\kappa}{2}.$$

We have more equations than the number of monomials if and only if $n\binom{\kappa}{3} \geq n\binom{\kappa}{2}$, say $\kappa \geq 5$. Under this condition, the rank of the system is necessarily smaller than $n\binom{\kappa}{2}$ because we have a solution. As we have seen, the solution set is stable by the action of the Frobenius map on each variable. Based on this observation, [2, Assumption 1] stated that this rank was equal to $n\binom{\kappa}{2} - n$ in the case of HFE. We will rely on the same assumption in the HPPC case.

The first step of the solving process is to perform linear combinations between the SM equations. Under the above assumption and by saturating the set of bilinear monomials, this will produce a total of

$$n\binom{\kappa}{2} - n - n_u n_{c_T} = n_{c_T} = \binom{\kappa}{2} - 1$$

linearly independent equations of degree 1 (recall that $n_u n_{c_T}$ refers to the number of monomials of exact degree 2). These equations necessarily involve one c_T variable from an argument similar to [2, Lemma 1], and we may use them to eliminate all the minor variables present in SM, by substitution. We can in fact directly apply [2, Lemma 1]. Indeed, the proof of this lemma simply exploits the special shape of the matrix $\boldsymbol{U}$ used in HFE, and we have the same matrix in HPPC. Overall, this step yields the announced quadratic system in the u_i's.

The second step is to solve the quadratic system using Gröbner bases. We note that this system can be made very overdefined by choosing a large value for κ. More precisely, it contains $(n-1)(\binom{\kappa}{2}-1)$ equations in $n-1$ unknowns. In the case of HFE, the argument of [2, Proposition 1] stated that a grevlex Gröbner basis could be obtained in degree 2 when $\binom{\kappa}{2} \geq n-1$.

In our experiments on HPPC, the very same assumption as [2, Assumption 1] on the rank of SM did not hold when the value κ was too small (for instance $\kappa = 5$), but it was satisfied for a larger value. In Section 5.3, we present our results for the least value of κ such that $\binom{\kappa}{2} \geq n-1$. For this number of columns, the number of linear polynomials produced at the first step was the expected one and the resulting quadratic system in the u_i's was already solved in degree 2.

5.2 Complexity analysis

The cost of solving MinRank can be derived from the results of [2] as we use the same approach. For instance, the first step can be estimated by [2, Eq. (8)] or [2, Eq. (9)] and the second step by [2, Eq. (10)] if the resulting quadratic system is solved in degree 2, with $n_u = n-1$ and $n_{c_T} = \binom{\kappa}{2} - 1$. Noting that these two values are polynomial in n and using these estimates, we deduce that both steps have polynomial complexity in the case of HPPC.

The overall attack is, in fact, practical, and we have been able to complete a key recovery for all security levels. We could recover an equivalent key in a bit less than 8 minutes for security level 2 ($n = 128$), about 1 hour and a half for security level 4 ($n = 192$), and a bit more than 7 hours for security level 5 ($n = 256$). A more detailed breakdown of the running time is given in Table 1 of the next subsection.

5.3 Concrete running times

Table 1 presents the time spent on each of the steps of the key recovery. The time given in "Step 1" corresponds to the cost of generating the degree 1 polynomials by linear algebra on the SM system. The time given in "Step 2" is the cost of solving the final quadratic system. This includes the cost of computing the

grevlex Gröbner basis (Gaussian elimination and checking that no new polynomials appear in degree 3) and the cost of the FGLM algorithm [13] to compute the variety of size n. For the sake of completeness, we also give the time spent to construct the SM system ("Build SM") and the one of the substitution of the degree 1 polynomials in SM to construct the final quadratic system ("Build quadratic"). These constructions were not detailed in Section 5.1, but they have no reason to be costly from a theoretical perspective (their rather high cost in Table 1 might be due to poor implementation).

For $n = 128$ and $n = 192$, we picked the least value of κ such that $\binom{\kappa}{2} \geq n-1$ to apply the above reasoning. For $n = 256$, the least value of κ satisfying this inequality is $\kappa = 24$. We have not been able to run Step 1 for this value of κ because we were limited by memory. Therefore, we give timings for $\kappa = 12$ columns. In particular, in this case, the time spent in Step 2 suggests that the quadratic system is solved in degree > 2.

Table 1. Attack running time on the HPPC security levels. All the numbers are in seconds and they correspond to one trial (they are not an average over several samples).

n	κ	Build SM	Step 1	Build quadratic	Step 2	Total
128	17	11.320	76.540	82.490	271.160	464.819 ($\approx 07'45''$)
192	21	49.570	655.700	627.130	4096.900	5552.319 ($\approx 1°32'32''$)
256	12	27.970	219.590	241.390	24405.100	25290.409 ($\approx 7°01'30''$)

Finally, we have not tried to find the value of κ for which Step 1 and Step 2 have roughly the same time complexity. This value will be larger than the one we give here, and thus the memory demand for Step 1 will be even more important.

6 Conclusion

This paper fully breaks HPPC because the time complexity of our rank attack is polynomial in n (we cannot simply increase the parameters). It is possible in principle to repair the scheme by modifying the central map to increase its Q-rank, but doing so would change the design and would not bring any advantage compared to older HFE-like schemes. Moreover, as with HFE, increasing the Q-rank enough to make the scheme secure (at least in any straightforward way) would make signing by the honest party unacceptably slow.

References

1. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D.: Status report on the third round of the NIST post-quantum cryptography standardization process. Tech. rep., National Institute of Standards and Technology, Gaithersburg, MD (July 2022)

2. Baena, J., Briaud, P., Cabarcas, D., Perlner, R., Smith-Tone, D., Verbel, J.: Improving Support-Minors Rank Attacks: Applications to GeMSS and Rainbow. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022. pp. 376–405. Springer (2022)
3. Bardet, M., Bertin, M.: Improvement of Algebraic Attacks for Solving Superdetermined MinRank Instances. In: Cheon, J.H., Johansson, T. (eds.) Post-Quantum Cryptography. pp. 107–123. Springer (2022)
4. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R.A., Smith-Tone, D., Tillich, J., Verbel, J.A.: Improvements of Algebraic Attacks for Solving the Rank Decoding and MinRank Problems. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 507–536. Springer (2020)
5. Bardet, M., Faugère, J.C., Salvy, B., Yang, B.Y.: Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. https://api.semanticscholar.org/CorpusID:10699714
6. Bettale, L., Faugère, J.C., Perret, L.: Cryptanalysis of HFE, Multi-HFE and Variants for Odd and Even Characteristic. Designs, Codes and Cryptography **69**(1), 1 – 52 (2013)
7. Beullens, W.: Improved Cryptanalysis of UOV and Rainbow. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. pp. 348–373. Springer (2021)
8. Buss, J.F., Frandsen, G.S., Shallit, J.O.: The Computational Complexity of Some Problems of Linear Algebra. J. Comput. Syst. Sci. **58**(3), 572–596 (Jun 1999)
9. Cartor, R., Smith-Tone, D.: EFLASH: A new multivariate encryption scheme. In: Cid, C., Jr., M.J.J. (eds.) Selected Areas in Cryptography – SAC 2018. pp. 281–299. Springer (2018)
10. Ding, J., Hodges, T.J.: Inverting HFE Systems Is Quasi-Polynomial for All Fields. In: Rogaway, P. (ed.) Advances in Cryptology – CRYPTO 2011. pp. 724–742. Springer (2011)
11. Ding, J., Perlner, R., Petzoldt, A., Smith-Tone, D.: Improved Cryptanalysis of HFEv- via Projection. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography. pp. 375–395. Springer (2018)
12. Dubois, V., Gama, N.: The Degree of Regularity of HFE Systems. In: Abe, M. (ed.) Advances in Cryptology – ASIACRYPT 2010. pp. 557–576. Springer (2010)
13. Faugère, J.C., Gianni, P.M., Lazard, D., Mora, T.: Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering **16**(4), 329–344 (1993)
14. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra **139**(1), 61–88 (1999)
15. Jiang, X., Ding, J., Hu, L.: Kipnis-Shamir Attack on HFE Revisited. In: Pei, D., Yung, M., Lin, D., Wu, C. (eds.) Information Security and Cryptology. pp. 399–411. Springer (2008)
16. Joux, A., Faugère, J.C.: Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Grobner Bases. In: Advances in Cryptology – CRYPTO 2003. pp. 44–60. Springer (2003)
17. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. In: Wiener, M. (ed.) Advances in Cryptology – CRYPTO 1999. pp. 19–30. Springer (1999)
18. Macario-Rat, G., Patarin, J.: Two-Face: New Public Key Multivariate Schemes. In: Joux, A., Nitaj, A., Rachidi, T. (eds.) Progress in Cryptology - AFRICACRYPT 2018. pp. 252–265. Springer (2018)

19. NIST Cryptographic Technology Group: Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process. NIST Computer Security Resource Center (2022), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf
20. Øygarden, M., Felke, P., Raddum, H.: Analysis of Multivariate Encryption Schemes: Application to Dob. In: Garay, J.A. (ed.) Public-Key Cryptography – PKC 2021. pp. 155–183. Springer (2021)
21. Øygarden, M., Felke, P., Raddum, H., Cid, C.: Cryptanalysis of the Multivariate Encryption Scheme EFLASH. In: CT-RSA 2020: The Cryptographers' Track at the RSA Conference 2020. p. 85–105. Springer (2020)
22. Øygarden, M., Felke, P., Raddum, H.: Analysis of Multivariate Encryption Schemes: Application to Dob and C*. Journal of Cryptology 2024 (to appear) https://eprint.iacr.org/2024/595
23. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: Maurer, U. (ed.) Advances in Cryptology – EUROCRYPT 1996. pp. 33–48. Springer (1996)
24. Rodriguez, B.G.: 3WISE: Cubic Element-Wise trapdoor based MPKC cryptosystem. NIST Round 1 submission to the Additional Call for Signature Schemes (2023), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/3wise-spec-web.pdf
25. Rodriguez, B.G.: HPPC: Hidden Product of Polynomial Composition. NIST Round 1 submission to the Additional Call for Signature Schemes (2023), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/hppc-spec-web.pdf
26. Rodriguez, B.G.: HPPC: Hidden Product of Polynomial Composition. Cryptology ePrint Archive, Paper 2023/830 (2023), https://eprint.iacr.org/2023/830
27. Smith-Tone, D.: A Total Break of the 3WISE Digital Signature Scheme. Cryptology ePrint Archive, Paper 2023/1535 (2023), https://eprint.iacr.org/2023/1535
28. Tao, C., Petzoldt, A., Ding, J.: Efficient Key Recovery for All HFE Signature Variants. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021. pp. 70–93. Springer (2021)
29. Wang, L.C., Chou, C.Y., Ding, J., Kuan, Y.L., Leegwater, J.A., Li, M.S., Tseng, B.S., Tseng, P.E., Wang, C.C.: SNOVA. NIST Round 1 submission to the Additional Call for Signature Schemes (2023), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/snova-spec-web.pdf
30. Wolf, C., Preneel, B.: Equivalent Keys in HFE, C*, and Variations. In: Dawson, E., Vaudenay, S. (eds.) Progress in Cryptology – Mycrypt 2005. pp. 33–49. Springer (2005)

A Experiments for the direct attack

This appendix contains more details on the experiments of Section 3.2. The column "F_4 step degrees" gives the degree of the polynomials that are handled at each step of the F_4 algorithm as well as the possible degree fall polynomials. For example, "3 (2:80)" means that 80 quadratic degree fall polynomials are found at a step in degree 3. As stated in Section 3.2, the last step that is presented always produces linear equations (we do not give their number here).

Our results are consistent with Proposition 1 and Proposition 2, with the exception of the two underlined values in Table 2 (we refer to Remark 1 for a possible explanation of this discrepancy).

While it seems feasible to conjecture the number of degree falls in degree 4, we have not attempted to do it. In addition to experimental differences when the polynomial ℓ_2 was sparser pointed out in the main text, we also want to stress that our results are given for one trial on a given set of parameters. In some cases, two trials for the same parameter set have yielded different results for the number of degree falls in degree 4.

Table 2. HPPC direct system with $d = 5$.

n	F_4 step degrees
40	3 (2:80), 3 (2:80), 3
50	3 (2:100), 3 (2:100), 3
60	3 (2:120), 3 (2:120), 3
70	3 (2:140), 3 (2:<u>280</u>), 3
80	3 (2:160), 3 (2:<u>240</u>), 3

Table 3. HPPC direct system with $d = 7$.

n	F_4 step degrees
40	3 (2:80), 3 (2:80), 3, 4
50	3 (2:100), 3 (2:100), 3, 4
60	3 (2:120), 3 (2:120), 3, 4
70	3 (2:140), 3 (2:140), 3, 4

Table 4. HPPC direct system with $d = 9$.

n	F_4 step degrees
46	3 (2:92), 3 (2:92), 3, 4 (3:3772), 4
47	3 (2:94), 3 (2:94), 3, 4 (3:3854), 4
48	3 (2:96), 3 (2:96), 3, 4 (3:3936), 4
49	3 (2:98), 3 (2:98), 3, 4 (3:4018), 4
50	3 (2:100), 3 (2:100), 3, 4 (3:4050), 4
51	3 (2:102), 3 (2:102), 3, 4 (3:4182), 4
52	3 (2:104), 3 (2:104), 3, 4 (3:4264), 4
54	3 (2:108), 3 (2:108), 3, 4 (3:4428), 4
55	3 (2:110), 3 (2:110), 3, 4 (3:4510), 4
56	3 (2:112), 3 (2:112), 3, 4 (3:4592), 4
58	3 (2:116), 3 (2:116), 3, 4 (3:4756), 4
60	3 (2:120), 3 (2:120), 3, 4 (3:4920), 4

Table 5. HPPC direct system with $d = 10$.

n	F_4 step degrees
40	3 (2:80), 3 (2:80), 3, 4
45	3 (2:90), 3 (2:90), 3, 4 (3:2070), 4
46	3 (2:92), 3 (2:92), 3, 4 (3:736), 4
47	3 (2:94), 3 (2:94), 3, 4 (3:752), 4
48	3 (2:96), 3 (2:96), 3, 4 (3:768), 4
49	3 (2:98), 3 (2:98), 3, 4 (3:784), 4
50	3 (2:100), 3 (2:100), 3, 4 (3:800), 4
55	3 (2:110), 3 (2:110), 3, 4 (3:880), 4 (3:6050), 4
58	3 (2:116), 3 (2:116), 3, 4 (3:928), 4 (3:5859), 4
60	3 (2:120), 3 (2:120), 3, 4 (3:960), 4 (3:6061), 4
61	3 (2:122), 3 (2:122), 3, 4 (3:976), 4
65	3 (2:130), 3 (2:130), 3, 4 (3:2665), 4
70	3 (2:140), 3 (2:140), 3, 4 (3:2380), 4

Table 6. HPPC direct system with $d = 11$.

n	F_4 step degrees
40	3 (2:80), 3 (2:80), 3, 4
45	3 (2:90), 3 (2:90), 3, 4 (3:2070), 4
46	3 (2:92), 3 (2:92), 3, 4, 5
47	3 (2:94), 3 (2:94), 3, 4, 5
48	3 (2:96), 3 (2:96), 3, 4 (3:960), 4

Table 7. HPPC direct system with $d = 20$.

n	F_4 step degrees
40	3 (2:80), 3 (2:80), 3, 4
50	3 (2:100), 3 (2:100), 3, 4, 5