Regev's attack on hyperelliptic cryptosystems

Razvan Barbulescu[1,2] and Gaetan Bisson[3]

[1] Institut de mathematiques de Bordeaux (Univ. Bordeaux, CNRS, Bordeaux INP)
[2] Institute of Mathematics of the Romanian Academy
[3] Laboratoire GAATI, University of French Polynesia

Abstract. Hyperelliptic curve cryptography (HECC) is a candidate to standardization which is a competitive alternative to elliptic curve cryptography (ECC). We extend Regev's algorithm to this setting. For genus-two curves relevant to cryptography, this yields a quantum attack up to eight times faster than the state-of-the-art. This implies that HECC is slightly weaker than ECC. In a more theoretical direction, we give a mathematical proof that Regev's algorithm obtains its full speedup with respect to Shor's when the genus is high, a setting which is already known to be inadequate for cryptography.

1 Introduction

Hyperelliptic curve cryptography (HECC), introduced by Koblitz in [Kob89], is an alternative to elliptic curve cryptography (ECC). The fastest implementations of HECC include [BS25,BCHL16,RSSB16,NTRKR$^+$24] and make it competitive to ECC; see [SRS$^+$23,YSG24] for a comparison. Embedded implementations include [FBV08] and [GCT17]. The fastest classical attack on ECC, Pollard's rho algorithm, also applies to HECC and has the same cost up to a very narrow difference in terms of classical operations; see [BCM14]. A precise estimation of the fastest quantum attack, Shor's algorithm, shows that ECC and HECC offer the same security in terms of the number of quantum gates required to implement the attack; see [CGHZ23]. NIST standardized elliptic curve cryptography in 1999 (see [The99]) and the patents on hyperelliptic curve cryptography have been withdrawn or expired, which further stimulated research on HECC. Additionally, pairing-based cryptography provides advanced primitives which can be based on hyperelliptic curves; see e.g. [KT08,FS11,Dry12,AFK24].

Hyperelliptic curves are a type of algebraic curves to which commutative groups admitting efficient group laws can be associated. More precisely, to a hyperelliptic curve C of genus g, one associates its Jacobian variety $\mathrm{Jac}(C)$ which is a commutative algebraic group of dimension g. Curves of genus $g = 1$ are called elliptic curves and, for them, the Jacobian variety coincides with the curve. The security of nearly all ECC and HECC primitives relies on the presumed hardness of the discrete logarithm problem (DLP) in an elliptic curve or in the Jacobian variety of an hyperelliptic curve; this includes the Diffie-Hellman key exchange, digital signature algorithm, etc. We note that recent developments

such as [FT19] propose alternative primitives based on the presumed hardness of isogeny computations instead of the DLP.

The state-of-the-art classical attack on the DLP in elliptic curves (ECC) and in Jacobian varieties of genus-two hyperelliptic curves (HECC) is Pollard's rho algorithm: its complexity is $\tilde{O}(Q^{1/2})$ where Q is the group order. Storing a group element requires $\log_2 Q$ bits and the group law has a quasilinear cost in $\log_2 Q$; see [Can87]. Note that more effective attacks exist for $g \geq 3$; see [Gau00,ADH94]. This makes ECC and genus-two HECC very attractive for classical cryptography.

Turning to quantum attacks, Shor's algorithm has variants for integer factorization and DLP in various groups. The algorithm has never been successfully run on a quantum computer (on examples larger than factoring $35 = 5 \cdot 7$) although this could conceivably be achieved in the coming years; see e.g. [Gid25]. It is thus important to identify cases where it can be implemented with minimal quantum resources, even when they are not actively used in current cryptographic applications. Therefore we seek to estimate the precise cost of quantum attacks on hyperelliptic curves, including those of genus three and more.

Shor's algorithm is based on the quantum Fourier transform (QFT). To factor an n-bit integer or to compute a discrete logarithm in a group whose order is an n-bit integer, it uses between n and $2n$ qubits depending on the quantum implementation and uses between $\tilde{O}(n^2)$ and $\tilde{O}(n^3)$ quantum gates; see e.g. [WLL+25]. The sliding window technique in the implementation of ECC and HECC (see [CFA+05, Chapter 9]) offers very important speedups despite higher requirements in terms of quantum memory; see [Gid19].

A series of QFT-based algorithms improve upon Shor's without reducing the number of quantum gates. They use several quantum runs which can be done in parallel; the results are combined in a classical post-treatment phase. Ekerå and Håstad [EH17] first proposed a variant for the short DLP; later, Ekerå [Eke19,Eke24] proposed a variant for the DLP with fewer restrictions, which we will simply call "Ekerå's algorithm" henceforth. Regev [Reg25] proposed a quantum algorithm for integer factorization which was later extended to the DLP in multiplicative groups of finite fields by Ekerå and Gärtner [EG24]; for simplicity we call these two variants as "Regev's algorithm."

Ekerå and Gärtner [EG25] compared Ekerå's and Regev's algorithms by implementing both on a classical simulator of a quantum computer. They obtained a precise estimation of the cost which showed an advantage for Ekerå's algorithm. One drawback of Regev's algorithm is that it uses a larger number of qubits: it was initially of size $\tilde{O}(n^{\frac{3}{2}})$ but was reduced to $O(n)$ by Ragavan and Vaikuntanathan in [RV24,Rag24].

Regev's algorithm offers an advantage compared to Ekerå's in that its correctness has been rigorously proven by Pilatte [Pil24]. In its initial form [Reg25], Regev's algorithm was heuristic as it assumed that a certain lattice was sufficiently random. To eliminate this heuristic, Pilatte [Pil24] introduced and analysed a modification of the algorithm. Barbulescu, Barcau, and Pasol [BBP24] noted that Regev's initial variant fails to factor infinitely many integers.

Barbulescu, Barcau and Pasol [BBP25] extended Regev's algorithm to elliptic curves. We compare the DLP in an n-bit multiplicative group of a finite field to the DLP in an n-bit elliptic curve. With respect to Shor's, Regev's algorithm [BBP25] has a speedup of $d = (\log n)^{\frac{1}{2}}$, which is unbounded albeit smaller than the speedup in the multiplicative group. The resulting algorithm is heuristic because this new case falls outside of Pilatte's proof.

1.1 Our contribution

In this work, we extend Regev's algorithm to Jacobian varieties of hyperelliptic curves of any genus g and we show that this yields a heuristic speedup of $d = \min(g, \sqrt{n})$ over Shor's algorithm. The number d is the main parameter in Regev's algorithm and we let $n = \log_2 |\mathbb{G}|$ where $\mathbb{G}$ is the group in which the discrete logarithm problem is to be solved, which we denote additively. Each run of Regev's algorithm requires $(d + \frac{n}{d})$ additions in $\mathbb{G}$. The parameter d must thus be taken as close as possible to $\sqrt{n}$ and the speedup over Shor's, which uses n additions, is approximated by d.

Without getting into technical details, which will be given in Section 2, Regev's algorithm requires d "small" elements in the group, i.e. elements which can be written on a small number of bits. If H is a hyperelliptic curve of genus g, the elements of $\mathrm{Jac}(H)$ can be represented in the form $P_1 + \cdots + P_r - r(\infty)$ where $r \leq g$, where $P_1, \ldots, P_r$ are points on H, and where ∞ denotes the point at infinity. Here, we select as "small" those elements of the form $P - \infty$, which corresponds to taking $r = 1$. This set of "small" elements in the Jacobian has also been used in state-of-the-art classical attack on hyperelliptic curves of high genus (see [Gau00]).

In the case $g \approx \sqrt{n}$, we prove that Pillate's proof extends to hyperelliptic curves. The following statement is a simplified version of Theorem 4.

Theorem 1. *There exists a proven quantum algorithm which solves the discrete logarithm problem in the Jacobians of a curve of genus $\sqrt{n}^{1+o(1)}$ using $\sqrt{n}$ runs, which can be done in parallel, of a quantum procedure which uses $\tilde{O}(n^{\frac{3}{2}})$ gates.*

This new case where Regev's algorithm is proven to be correct increases the confidence that the heuristics of the algorithm hold in general.

When $g = 2$, which includes hyperelliptic curves proposed for standardisation, a different method is needed. We explain how the idea in [BBP25] extends to genus-two curves. On specific curves proposed for cryptographic deployment, we show that practical speedups of eight can be achieved.

Putting our results into context. As all cryptographic assumptions, the hardness of DLP in Jacobian varieties of genus-two curves was subjected to a period of scrutiny. For reference, the DLP in elliptic curves was proposed in 1985 (see [Mil85,Kob87]) and standardized in 1999 (see [The98,The99]); lattice-based cryptography was introduced in 1996-2009 (see [HPS98,Reg09] and standardized in 2020 (see [Moo21]). Although HECC was proposed in 1987, its development

followed the steps of ECC about fifteen to twenty years later. For instance, point counting algorithms for elliptic curves were designed in 1985 and, in 2006, Bernstein [Ber06] noted that the standardization of HECC was "premature" because of the lack of practical point-counting algorithm for hyperelliptic curves.

Today, one can assert that HECC has received at least as much attention as ECC at the moment of its standardization. Hyperelliptic curves are the object of significant research efforts. However, NCSC [Nat25] and NIST [Nat23] seem to indicate that DLP-based cryptography will be "disallowed" after 2035. A hypothetical standardization would offer no time for the deployment of HECC.

For the purpose of proving quantum advantage, HECC is a valuable candidate. On the one hand, at a given key size, it is estimated to provide the same level of security as ECC; on the other hand, Regev's algorithm or a combination of Regev's and Ekerå's can use a better d parameter.

Roadmap. In Section 2, we review mathematical background on hyperelliptic curves; in Section 3, we recall Regev's algorithm and explain the importance of "small" group elements. In Section 4, we analyse Regev's algorithm in the case of genus-g hyperelliptic curves for large values of g and obtain a rigorous algorithm with a speedup as large as in the case of the integer factorisation. In Section 5, we treat the case of genus-two curves, which is considered for standardisation in HECC and propose concrete choices of small points for some curves used in implementations. Finally, in Section 6, we discuss the path towards an implementation of Regev's algorithm in its initial form or of combinations with Ekerå's algorithm.

2 Background on hyperelliptic curves

2.1 The group law associated to a hyperelliptic curve

We refer to the excellent handbook [CFA+05] for the necessary background on elliptic and hyperelliptic curves.

A hyperelliptic curve H is given by an equation of the form $y^2+h(x)y = f(x)$ where h and f are two polynomials satisfying $\deg h < \deg f$. In odd characteristic, it can be put under the form $y^2 = f(x)$ with f of odd degree (see [CFA+05, Equation 4.2, page 74]); all the curves in the present article are of this form. Their coefficients lie in the so-called field of definition of the curve; we denote it by k and it will either be a finite field $\mathbb{F}_q$ or the field of the rationals $\mathbb{Q}$. The rational points of H are solutions $(x, y) \in k^2$ to the curve equation, together with the so-called point at infinity, which we denote by ∞. The genus of H is the integer g such that $\deg f = 2g+1$; the particular case $g = 1$ is that of elliptic curves.

From the set of rational points of H, the Jacobian variety $\mathrm{Jac}(H)$ is constructed as follows. First consider divisors, that is, formal combinations of the form $n_1P_1 + \cdots + n_kP_k$ where the P_i's are points on H and the n_i's are relative integers. Elements of $\mathrm{Jac}(H)$ can be naturally represented as reduced degree-zero divisors, that is, divisors of the form $P_1 + \cdots + P_r - r(\infty)$ where $r \leq g$

and the P_i's contain no two points of the form (x, y) and $(x, -y)$; see [CFA+05, Lemma 4.135]. The group law can be computed by adding divisors as formal combinations and reducing the resulting divisor.

An alternative definition of the Jacobian is to identify it with the class group of the function field of H, that is, $K = k[x, y]/\langle y^2 - f(x)\rangle$; see [CFA+05, Proposition 4.139]. The elements of the class group can be represented by integral ideals of the function field, which have a two-element representation $(u(x), P(x, y))$ for some $u(x) \in k[x]$ and $P(x, y) \in k[x, y]$. The degree of P is less than the degree of H in y, so it is at most 1 and we simply represent the elements of the Jacobian as couples of polynomials $(u(x), v(x)) \in (k[x])^2$; this is the Mumford representation; see [CFA+05, Theorem 4.145]. Each element admits a unique reduced representation whereby $u(x)$ is monic and $\deg v(x) < \deg u(x) \leq g$. This representation is also suitable for efficient group law computation: Cantor's algorithm [Can87] achieves this by exploiting multiplications and GCD's in the ring $k[x]$.

Elements of Jacobian varieties can be converted from divisor representation to Mumford representation and vice-versa by interpolating and factoring polynomials, respectively. This can be done in classical polynomial time; see [CFA+05, Section 14.3.2]. Figure 1 illustrates both representations and their conversion. The result below gives a precise estimation of the cost of changing representation.

Lemma 1 (Section 10.2 in [vzGG03]). *Let H be a hyperelliptic curve in Weierstrass form defined over $\mathbb{F}_q$ whose genus is g. Let $P_1, \ldots, P_r$ be points on $H(\mathbb{F}_q)$ for some $r \leq g$, such that no two have the same x-coordinate. The Mumford representation of $(P_1)+\cdots+(P_r)-r(\infty)$ can be computed in $O(n \log^2 n)$ classical binary operations.*

Note that, for any integer $r \leq g$, a reduced divisor of the form $(P_1) + \cdots + (P_r) - r(\infty)$ corresponds to a reduced Mumford representation with $u(x)$ of degree r. The generic case is $r = g$ and, indeed, the group order has the same bit size as q^g.

More precisely, by the Mordell–Weil theorem, the number of points on an hyperelliptic curve H of genus g defined over a finite field $k = \mathbb{F}_q$ satisfies

$$|\#H(\mathbb{F}_q) - (q + 1)| \leq 2g\sqrt{q}$$

and the order of its Jacobian variety lies in the interval

$$\#\operatorname{Jac}_{\mathbb{F}_q}(H) \in \left[(\sqrt{q} - 1)^{2g}, (\sqrt{q} + 1)^{2g}\right].$$

See [CFA+05, Corrolaries 5.80 and 5.81].

Over finite fields $k = \mathbb{F}_q$ and assuming $q \geq g$, as is always the case in cryptography, evaluating the group law of $\operatorname{Jac}(H)$ using Cantor's algorithm uses $O(g \log g)$ operations in $\mathbb{F}_q$. The binary cost of such operations is bounded by $M(q) = O(\log q \log\log q)$ by [HvdH21]. Since the binary size n of the order of the Jacobian variety of a hyperelliptic curve H satisfies $n = \log_2 |\operatorname{Jac}(H)| \approx g \log_2 q$ by the Mordell–Weil bound, we deduce that computing its group law takes $O(n(\log n)^2)$ binary operations.

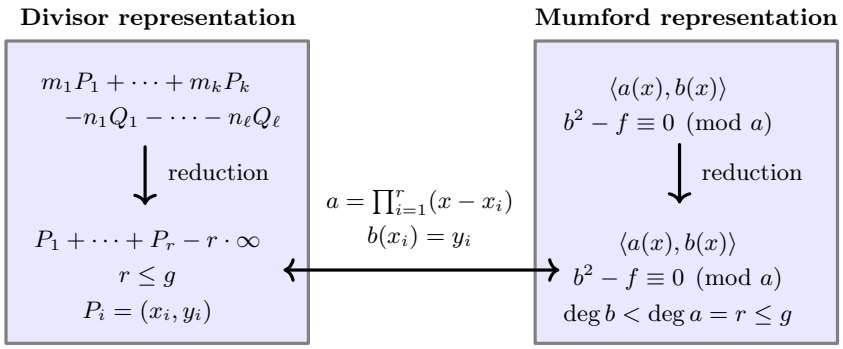

Fig. 1: The divisor and Mumford representations of the elements of the Jacobian of a hyperelliptic curve of genus g.

2.2 The Riemann hypothesis for hyperelliptic curves

Recall that the Jacobian of an algebraic curve is the class group of its function field. Broadly speaking, function fields are analogous to number fields. For example, Bach [Bac90] proved that the class group of a number field is generated by "small" ideals under the Generalized Riemann Hypothesis (GRH); later, Müller, Stein and Thiel [MST99] proved that the Jacobian of a hyperelliptic curve of large genus is generated by "small" elements of the Jacobian. In this section, we consider the function field K of a hyperelliptic curve H defined over a finite field different from $\mathbb{F}_2$. For such fields, the Riemann hypothesis is proven (see [CFA+05, Theorem 8.3]); this allows us to obtain a rigorous proof of our algorithm for certain values of the genus g. This section may be skipped on a first reading; its content is only required for the rigorous proof of correction of our algorithm (see Theorem 4 and the preceding lemma).

We recall results from [Ros02]. Let H be a hyperelliptic curve defined over a finite field $k = \mathbb{F}_q$ with q odd and denote by K its function field. Elements of the Jacobian variety can be defined as divisors of H or, equivalently, ideals of the ring of integers $\mathcal{O}_K$; thus, we refer to $\mathrm{div}(H)$ in place of $\mathrm{Jac}(H)$. The norm of a divisor $\mathfrak{U}$ is the integer $N(\mathfrak{U}) = q^{\deg \mathfrak{U}}$. A character on $\mathrm{div}(H)$ of finite order is a group morphism $\chi : (\mathcal{O}_K/\mathfrak{F})^* \to \mathbb{C}^*$ for some place $\mathfrak{F}$. It induces a map $\chi : \mathrm{div}(H) \to \mathbb{C}^*$. The L-function associated to χ is

$$L(s, \chi, K) = \sum_{\mathfrak{U}} \frac{\chi(\mathfrak{U})}{N(\mathfrak{U})^s} \quad (\Re(s) > 1),$$

where the sum is over integral divisors of $\mathcal{O}_K$. Setting $u = q^{-s}$ we have

$$L(s, \chi, K) = \prod_{\mathfrak{P}} \frac{1}{1 - \chi(\mathfrak{P}) u^{\deg \mathfrak{P}}}, \tag{1}$$

where the product is over prime ideals of $\mathcal{O}_K$.

By the Generalized Riemann Hypothesis (see [Ros02, Theorem 5.10] or [FR10, Equation 2.3]), we know that there exist $\omega_1(\chi), \ldots, \omega_{\deg \mathfrak{F}-1}(\chi)$ such that $|\omega_i(\chi)| = q^{\frac{1}{2}}$ and

$$L(s, \chi, K) = (1-u)^{1+(-1)^{\mathrm{ord}(\chi)}} \prod_{i=1}^{\deg \mathfrak{F}-1} (1 - \omega_i(\chi) u), \tag{2}$$

where $\mathrm{ord}(\chi) \equiv 1 \pmod 2$ if the order of χ is odd and 0 otherwise. Note that $1/\omega_1(\chi), \ldots, 1/\omega_{2g-2}(\chi)$ are the zeros of $L(s, \chi, K)$.

The following result and its proof are classical; a similar argument can be found in [FR10, Lemma 2.1].

Lemma 2. *Let χ be a non-trivial Dirichlet character modulo $\mathfrak{F}$. Then, for any $\epsilon > 0$ and any $m \geq 1$, one has*

$$\left| \sum_{\deg \mathfrak{U} \leq m} \chi(\mathfrak{U}) \Lambda(\mathfrak{U}) \right| \leq \frac{3}{2} (\deg \mathfrak{F}) q^{\frac{m}{2}} \tag{3}$$

where Λ denotes the analogue of von Mangoldt's function defined by

$$\Lambda(\mathfrak{U}) = \begin{cases} \deg \mathfrak{P} & \text{if } \mathfrak{U} = \mathfrak{P}^e \text{ is a prime power,} \\ 0 & \text{otherwise.} \end{cases}$$

Proof. We consider the truncation modulo u^{m+1} in the u-expansion of $u \frac{\partial}{\partial u}(\log L_\chi)$. From Equation (1) we have

$$\begin{aligned} u \frac{\partial}{\partial u}(\log L_\chi) &= (-u) \sum_{\mathfrak{P}} \frac{\partial}{\partial u} \log(1 - \chi(\mathfrak{P}) u^{\deg \mathfrak{P}}) \\ &= (-1) \sum_{\mathfrak{P}} -\chi(\mathfrak{P}) (\deg \mathfrak{P}) u^{\deg \mathfrak{P}} (\sum_{e \geq 0} \chi(\mathfrak{P})^e u^{e \deg \mathfrak{P}}) \\ &= \sum_{\mathfrak{U}} \chi(\mathfrak{U}) \Lambda(\mathfrak{U}) u^{\deg \mathfrak{U}}. \end{aligned}$$

First consider the case where the character χ is odd. From Equation (2), we have

$$\begin{aligned} u\frac{\partial}{\partial u}(\log L_\chi) &= u\frac{\partial}{\partial u}\sum_{i=1}^{\deg \mathfrak{F}-1}\log(1-\omega_i(\chi)u) \\ &= (-u)\sum_{i=1}^{\deg \mathfrak{F}-1}\frac{\omega_i(\chi)}{1-\omega_i(\chi)u} \\ &= (-u)\sum_{i=1}^{\deg \mathfrak{F}-1}\omega_i(\chi)(\sum_{e\geq 0}\omega_i(\chi)^e u^e) \\ &= (-1)\sum_{e\geq 1}\left(\sum_{i=1}^{\deg \mathfrak{F}-1}\omega_i^e\right)u^e. \end{aligned}$$

For all $e \leq m$, the coefficient of u^e in the series above is

$$\begin{aligned} \left|[u^e]u\frac{\partial}{\partial u}(\log L_\chi)\right| &\leq \sum_{i=1}^{\deg \mathfrak{F}-1}|\omega_i|^e \\ &\leq (\deg \mathfrak{F})q^{\frac{e}{2}}. \end{aligned}$$

The last inequality uses the Generalized Riemann Hypothesis for H. Finally, the desired expression is the sum of the coefficients for all $e \leq m$, so

$$\begin{aligned} \left|\sum_{\mathfrak{U}}\chi(\mathfrak{U})\Lambda(\mathfrak{U})\right| &\leq \sum_{e\geq 1}[u^e]\sum_{\mathfrak{U}}\chi(\mathfrak{U})\Lambda(\mathfrak{U}) \\ &\leq (\deg \mathfrak{F})\sum_{e=1}^{m}q^{\frac{e}{2}} \\ &\leq (\deg \mathfrak{F})q^{\frac{m}{2}}. \end{aligned}$$

The case where the character χ is even is similar but includes coefficients corresponding to $e-1$:

$$\begin{aligned} \left|\sum_{\mathfrak{U}}\chi(\mathfrak{U})\Lambda(\mathfrak{U})\right| &\leq \sum_{e\geq 1}([u^e]+[u^{e-1}])\sum_{\mathfrak{U}}\chi(\mathfrak{U})\Lambda(\mathfrak{U}) \\ &\leq (\deg \mathfrak{F})(q^{\frac{m}{2}}+q^{\frac{m-1}{2}}) \\ &\leq \frac{3\deg \mathfrak{F}}{2}q^{\frac{m}{2}}. \end{aligned}$$

3 Extending Regev's algorithm to hyperelliptic curves

3.1 Regev's algorithm in a nutshell

Consider an abelian group $\mathbb{G}$ written in additive notation. Given two group elements u and v such that v is a multiple of u, the discrete logarithm problem

(DLP) consists in finding an integer x such that $v = [x]u$; this is the discrete logarithm $\log_u v$. Let n be the number of bits in the order of $\mathbb{G}$, that is, $n = \lceil \log_2 |\mathbb{G}| \rceil$. To solve the DLP, Shor's algorithm computes, in quantum superposition, $[z]u$ for all n-bit integers z. Regev's algorithm evaluates, also in superposition, expressions of the form $[z_1]b_1 + \cdots + [z_d]b_d$ where d is a parameter to be optimized, the coefficients $z_1, \ldots, z_d$ are n/d-bit integers, $b_d = u$, $b_{d-1} = v$, and $b_1, \ldots, b_{d-2}$ are carefully-selected elements of $\mathbb{G}$.

Regev's algorithm makes use of the Gaussian function which, for a constant $s > 0$, is defined by

$$\rho_s : \begin{cases} \mathbb{R}^d \longrightarrow \mathbb{R} \\ \quad z \longmapsto \exp(-\pi \frac{\|z\|^2}{s}). \end{cases}$$

To any lattice $\mathcal{L}$ of dimension d in $\mathbb{R}^d$ one can associate a dual lattice $\mathcal{L}^*$. There exists a classical algorithm of polynomial time complexity which computes a basis of $\mathcal{L}$ from an approximation of a basis of $\mathcal{L}^*$. The following lattice is central in Regev's algorithm:

$$\mathcal{L} = \left\{ (z_1, \ldots, z_d) \in \mathbb{Z}^d : \sum_{i=1}^{d} [z_i] b_i = 0 \right\}. \tag{4}$$

Algorithm 1 is a verbatim adaptation of [EG24, Section 2.1] to the case of hyperelliptic curves.

When we say that an element is "small" we refer to its natural height.

Definition 1. *The (natural) norm of a rational a/b is $N(a/b) := \max(|a|, |b|)$. Its (natural) height is $h(a/b) = \log_2 N(a/b)$. Similarly, for a rational fraction $a(t)/b(t) \in \mathbb{F}_q[t]$ for a finite field $\mathbb{F}_q$ the norm is $N(a/b) = \max(q^{\deg a}, q^{\deg b})$. For a point (x, y) of an algebraic curve over $\mathbb{Q}$ or $\mathbb{F}_q(t)$ the norm and the height are those of the x-coordinate. The norm of an element of a Jacobian in Mumford representation, $\langle a(x), b(x) \rangle$ the norm and the height are those of $a(x)$. Note that the height corresponds, up to a factor two, to the bit size to store the ement on a computer. It might differ by a factor two from other definitions of the natural height in the literature.*

Under the following assumption, Ekerå and Gärtner proved the correctness of Regev's algorithm for a family of commutative groups $\mathbb{G}$ and for elements $b_1, \ldots, b_d$ selected by a given probabilistic algorithm.

Assumption 1 (Assumption 1 in [EG24]) *There exists a constant K such that, with probability $1 - o(1)$, the lattice $\mathcal{L}$ in Equation 4 has a basis formed of vectors of norm at most $T = \exp(Kn/d)$.*

Theorem 2 (Theorem 1 in [EG24]). *Let K be a constant. Let $\mathbb{G}$ be a commutative group whose order is an n-bit integer and where the group law can be computed in $\tilde{O}(n)$ binary operations. Let $d = d(n) = O(\sqrt{n})$. Assume that $\mathbb{G}$ has a subset S such that for any $b_1, \ldots, b_d \in S$ and any $\varepsilon \in \{0, 1\}^d$, the sum $\sum_{i=1}^{d} [\varepsilon_i] b_i$ can be computed in $\tilde{O}(n)$ binary operations. Make Assumption 1 for*

Algorithm 1 Regev's algorithm for hyperelliptic curves.
Its two variants are: Regev's initial variant (Line 2) and Pilatte's variant (Line 4).

Input: – a hyperelliptic curve H defined over $\mathbb{F}_q$ whose genus is g
– an integer parameter $d \leq \min(g, \sqrt{n})$
– $u, v \in \mathrm{Jac}_{\mathbb{F}_q}(H)$ such that v is a multiple of u
– a choice between Regev's and Pilatte's variants

Output: $\log_u v$

1: **if** the desired variant is Regev's **then**
2: $P_1, \ldots, P_d \leftarrow d$ points of $H(\mathbb{F}_q)$ with smallest heights
3: **else** ▷ the desired variant is Pilatte's
4: $P_1, \ldots, P_d \leftarrow d$ points of $H(\mathbb{F}_q)$ drawn uniformly with height $\leq 100d \log_2 d$
5: **end if**
6: $b_1 \leftarrow (P_1) - (\infty), \ldots, b_{d-2} \leftarrow (P_{d-2}) - (\infty)$; $b_{d-1} \leftarrow v$; $b_d \leftarrow u$
7: **for** d parallel runs **do** ▷ quantum
8: Prepare $\psi_1 \leftarrow \sum_{(z_1,\ldots,z_d)} \rho(z)|z\rangle|0\rangle$ where $\rho = \rho_s$ with $s = 1/\lceil\sqrt{2d}\rceil$.
9: Compute $\psi_2 \leftarrow \sum_{(z_1,\ldots,z_d)} \rho(z)|z\rangle|f(z)\rangle$ where $f(z) = [z_1]b_1 + \cdots + [z_d]b_d$.
10: Apply the QFT to ψ_2 and obtain $\psi_3 \leftarrow \sum_{(y_1,\ldots,y_d)} \widehat{\rho \cdot f}(y) \cdot |y\rangle$.
11: Measure ψ_3 to obtain a vector in $\mathbb{R}^d$ which approximates a vector of $\mathcal{L}^*$ drawn uniformly at random.
12: **end for**
13: Use the previously measured vectors to compute a basis of $\mathcal{L}$ and deduce the desired discrete logarithm. ▷ classical

$\mathbb{G}$ and a set of d elements $b_1, \ldots, b_d$ in S. Then there exists a probabilistic algorithm to compute, with probability $1 - o(1)$, any discrete logarithm in $\mathbb{G}$ using $d + 4$ calls to a quantum procedure of $\tilde{O}\left((d + \frac{n}{d})n\right)$ quantum gates.

3.2 An extension of Regev's algorithm to Jacobian varieties

Regev's algorithm for discrete logarithm [EG24] applies to any commutative group $\mathbb{G}$ and any choice of base elements $b_1, \ldots, b_{d-2}$ as soon as the following conditions are satisfied:

(i) The cost of Line 9 is $\tilde{O}(n)$.
(ii) Assumption 1 holds for $\mathbb{G}$ and the b_i's.

We shall prove that condition (i) is satisfied by the choice of elements $b_1, \ldots, b_d$ made in Line 6 of Algorithm 1.

As for condition (ii), Ekerå and Gärtner noted in [EG24, Example in Section 2.3] that it fails when there exists an element $b \in \mathbb{G}$ such that $b_i = [i]b$ for $i = 1, 2, \ldots, d-2$. More generally, condition (ii) is unlikely to hold when there is a "systematic" linear relation among the elements $b_1, \ldots, b_{d-2}$. This is not the case in Algorithm 1 because this set of generators has been implemented in Adleman's classical algorithm which solves the DLP in sub-exponential time; see [ADH94].

Pilatte [Pil24, Section 1.4] pointed out that condition (ii) cannot be satisfied by a set $\{b_1, \ldots, b_{d-2}\}$ which generate a proper subgroup of $\mathbb{G}$. This is called the

subgroup obstruction. In Regev's initial variant of Algorithm 1, two contemporaneous results of Enge [Eng02, Corrolary 4.2] and Voloch [Vol00, Main theorem] state that, when $g \leq n/(2\log n)$ the subgroup obstruction is avoided. Indeed, a sufficient condition is that $q > (8g-1)^2$, which is true when $g \leq n/(2\log n)$ and $n \geq 8$. Note however that in Regev's variant we cannot eliminate Assumption 1. Alternatively, when $g \geq \sqrt{n}$ one can replace q with $q^{2^{\log_2 n}}$ so as to satisfy the condition; the runtime of Regev's algorithm is then multiplied only by a factor $\log_2 n$ leaving unchanged the complexity $\tilde{O}(n^{3/2})$.

In Pilatte's variant, condition (ii) is a consequence of the Generalized Riemann Hypothesis for a well-chosen L-function. Since it is proven for function fields, we obtain a proof of Theorem 1, as claimed in the introduction.

In Sections 4 and 5 our main concern will be to satisfy condition (i) and only afterwards we specify whether the corresponding variant of Regev's algorithm is rigorous or heuristic.

Step 9 of Algorithm 1 is done using Pippenberger's multi-scalar product; see [Pip80]. Similar to the double-and-add algorithm, this algorithm exploits the sums $\sum_{i=1}^{d}[\varepsilon_i]b_i$ for $\varepsilon \in \{0,1\}^d$. In Pippenger's algorithm the values of these sums are stored in a lookup table. In quantum computing, memory access has a different behaviour, thus Regev's algorithm [Reg25] avoids using memory and, instead, modifies the multi-scalar product so that only a constant number of b_i's are stored and any sum $\sum_{i=1}^{d}[\varepsilon_i]b_i$ is computed on the fly in time $\tilde{O}(n)$. The elements b_i with $i \leq d-2$ are selected so that they can be stored using few bits; see [BBP25]. We will later explain why this allows computing the multi-scalar product in time $\tilde{O}(n)$.

Question 1. How can one generate small elements of the Jacobian variety of a hyperelliptic curve of genus g?

The answer depends not only on the bit size $n = g\log q$ of the Jacobian variety but also on its genus g. From now on, we deal separately with two cases:

- g is large, i.e. it is an increasing function of n, in Section 4;
- g is small, i.e. is bounded (typically, $g = 2$), in Section 5.

In each case, we discuss how large a value of d can be achieved, analyze the resulting complexity of Regev's attack, and compare it with Shor's.

4 Speedup for large genera

The DLP variant of Regev's algorithm was first restricted [EG24] to computing discrete logarithms in $\mathbb{G} = \mathbb{F}_p^\times$. Careful scrutiny of the proof shows that it applies equally to any group where the lattice of Equation (4) admits a basis of relatively short vectors and where a set of elements can be summed much faster than general elements of $\mathbb{G}$. These conditions are to be compared to the definition of Regev-friendly groups in [BBP25] and are made precise in the following result.

From here on, $\mathbb{G}$ is the Jacobian variety of a hyperelliptic curve H of genus g defined over a finite field $\mathbb{F}_q$. In view of the Mumford representation which we

rely on, there is a natural choice for the small, independent elements $b_1, \ldots, b_{d-2}$ to be used in Regev's algorithm:

Answer 1 (to Question 1, for large values of g) *The elements $b_1, \ldots, b_{d-2}$ are selected to be of form $(P)-(\infty)$ for points P on the hyperelliptic curve $H(\mathbb{F}_{q^\lambda})$ for a parameter $1 \leq \lambda \leq \log n$.*

4.1 Regev's variant

Theorem 3 (under Assumption 1). *Let H be a hyperelliptic curve of genus g defined over $\mathbb{F}_q$ and let n denote the number of bits of the cardinality of its Jacobian variety $\mathbb{G} = \mathrm{Jac}(H)$. Consider the problem of computing $\log_u v$ for two elements $u, v \in \mathbb{G}$. Let $d = \min(g, \sqrt{n})$ and let $P_1, \ldots, P_{d-2}$ be points on H. Set $b_1 = (P_1) - (\infty)$, ..., $b_{d-2} = (P_{d-2}) - (\infty)$ as well as $b_{d-1} = v$ and $b_d = u$. Assume $g \leq n/(2 \log n)$ and that Assumption 1 holds for $\mathbb{G}$ and $b_1, \ldots, b_d$.*

Then Regev's variant of Algorithm 1 computes $\log_u v$ with a success probability of $1 - o(1)$ using d calls to a quantum procedure of $\tilde{O}\left((d + \frac{n}{d})n\right)$ gates.

Proof. The above statement is proven as a direct application of Theorem 2; what is needed is to check that it is well-defined and that the assumptions are satisfied.

The condition $g \leq n/(2 \log n)$ allows to avoid the obstruction in Assumption 1. It also implies that

$$\log q \geq 2 + \log d.$$

Since, by the Mordell-Weil theorem, we have $\#H(\mathbb{F}_q) > q - 2g\sqrt{q}$, there are at least $q/2$ points on H and thus at least $q/4$ pairs of points sharing the same x-coordinate. One can then select d points with pairwise distinct x-coordinates.

It remains to check that, for any $\varepsilon \in \{0,1\}^d$, the sum $\sum_{i=1}^{d-1} [\varepsilon_i] b_i$ can be evaluated in time $\tilde{O}(n)$. For this, we apply Lemma 1:

$$\text{cost} = g \log^2 g (\log q \log\log q) \leq n \log^2 n = \tilde{O}(n). \tag{5}$$

Note that Regev's algorithm obtains its full speedup with respect to Shor's algorithm whenever $g \geq \sqrt{n}$.

4.2 Pilatte's variant

For a rigorous analysis of Regev's algorithm, we use the results in Section 2.2 and rely on the following results.

Lemma 3 (discussion following Proposition 2.9 in [Pil24]). *Let K/k be a number field (resp. the function field of a hyperelliptic curve of genus g defined over $\mathbb{F}_q$) and χ a character modulo a divisor $\mathfrak{F}$ of K. Let $\mathrm{div}(K)$ be the divisor group of K, that is, the set of integral ideals (resp. principal divisors on H). Let*

$b_1, \ldots, b_{d-2}$ be drawn uniformly at random from elements of $\mathrm{div}(K)$ *with norm* $\leq d^{100d}$*. Assume that there exists* $\alpha < 1$ *such that, for all* $x > 0$*, one has*

$$\left| \sum_{\substack{\mathfrak{U} \in \mathrm{div}(K) \\ \mathrm{N}(\deg \mathfrak{U}) \leq x}} \chi(\mathfrak{U}) \Lambda(\mathfrak{U}) \right| \ll x^{\alpha} (\log \mathrm{N}(\deg \mathfrak{F}) + \log x)^2. \tag{6}$$

Then Assumption 1 holds for $\mathbb{G} = \mathrm{cl}(\mathcal{O}_K)$ *(resp.* $\mathbb{G} = \mathrm{Jac}(H)$*) and* $b_1, \ldots, b_d$ *for any fixed pair* (b_{d-1}, b_d)*.*

Remark 1. As an intermediate result, Pilatte [Pil24, Appendix A] proves that the above result implies

$$\left(\sum_{\substack{\mathfrak{P} \in \mathrm{div}(K) \text{ prime} \\ \mathrm{N}(\mathfrak{P}) \leq x}} \chi(\mathfrak{P}) \right) \left(\sum_{\substack{\mathfrak{P} \in \mathrm{div}(K) \text{ prime} \\ \mathrm{N}(\mathfrak{P}) \leq x}} 1 \right)^{-1} \leq x^{1-\alpha} (\log \mathrm{N}(\mathfrak{F}) + \log x)^3.$$

This is also proven in [FR10, Lemma 2.1].

Theorem 4. *Applied to Jacobian varieties of hyperelliptic curves of genus* $g = n^{\frac{1}{2}+\epsilon(n)}$ *for a function* $\epsilon(n) \geq 0$ *such that* $\epsilon(n) \to 0$*, Pilatte's variant of Regev's algorithm for hyperelliptic curves (Algorithm 1) computes discrete logarithms with* $O(\sqrt{n})$ *calls to a quantum procedure of* $\tilde{O}(n^{\frac{3}{2}})$ *gates.*

Proof. In order to apply Theorem 2, we check its hypotheses. We set $d = \sqrt{n}$. By Lemma 2 the assumptions of Lemma 3 are met. Hence, the group $\mathbb{G} = \mathrm{Jac}(H)$ and $b_1, \ldots, b_d$ satisfy Assumption 1.

Recall that any divisor whose Mumford representation is $(a(x), b(x))$ has norm q^{λ} where $\lambda = \deg a$. To convert to divisor representation, we consider the points of $H(\mathbb{F}_{q^{\lambda}})$ for the integer λ which best approximates the following:

$$q^{\lambda} \approx d^{100d}.$$

Since $g = n^{\frac{1}{2}+\epsilon(n)}$ we have $\lambda \leq \log(n)$.

By Lemma 1, the computation of $\sum_{i=1}^{d-2} [\varepsilon_i] b_i$ has complexity

$$\mathrm{cost} = g \log g (\log(q^{\lambda}) \log\log(q^{\lambda}) = O(n \log^3 n) = \tilde{O}(n).$$

By Theorem 2, the cost of Regev's algorithm is $\tilde{O}((d + \frac{n}{d})n)$ and we conclude because $d = \sqrt{n}$.

5 Speedup for small genera

We now turn to addressing Question 1 for small values of g. The case $g = 1$ has been treated in [BBP25]. Our answer here applies to any small value $g \geq 2$ but,

for simplicity, we state it for the specific case $g = 2$; this is the most relevant genus for cryptographic applications.

The previous section uses elements of the form $(P) - (\infty)$ which, in small genera, are not "small": when $g = 1$, all elements of the Jacobian are of this form and, when $g = 2$, all elements of the Jacobian is a sum of two such divisors. Instead, we describe a strategy which yields an improved speedup of up to $d = 8$ for Regev's attack compared to Shor's on cryptographically relevant curves. The key idea is to search for twist curves admitting "small" elements $b_1, \ldots, b_{d-2}$.

5.1 A general strategy to find small elements on Jacobian varieties

We recall the strategy of [BBP25] for elliptic curves and extend it to hyperelliptic curves. An hyperelliptic curve $H : y^2 = f(x)$ defined over a finite field $k = \mathbb{F}_q$ may be lifted to the rationals by lifting the coefficients of its equation. We denote the associated Jacobian variety by $\mathrm{Jac}_{\mathbb{Q}}(H)$. The Mordell–Weil theorem, which states that $\mathrm{Jac}_{\mathbb{Q}}(H) \simeq T \times \mathbb{Z}^r$ for a finite group T and a positive integer r, is constructive and allows us to extract a free family from any set of elements of $\mathrm{Jac}_{\mathbb{Q}}(H)$; see e.g. [Sch95]. This yields bases for the free part of $\mathrm{Jac}_{\mathbb{Q}}(H)$ of which the elements typically have small height. One may then reduce these elements to $\mathrm{Jac}_{\mathbb{F}_q}(H)$ so as to obtain small, independent elements to be used as $b_1, \ldots, b_{d-2}$ in Regev's algorithm. Step 9 is done in $\mathrm{Jac}_{\mathbb{Q}}(H)$ so that the coefficients of the intermediate results can be represented on a small number of bits and hence the arithmetic is fast.

We obtain the following answer.

Answer 2 (to Question 1, for small values of g) *Compute a basis for the Mordell–Weil group of the Jacobian variety* $\mathrm{Jac}_{\mathbb{Q}}(H)$ *over* $\mathbb{Q}$*. Reduce the elements of this basis back to* $k = \mathbb{F}_q$*. Furthermore, as we will see, a twist of the curve* H *may be used in place of* H *itself.*

It can happen that the lift $\mathrm{Jac}_{\mathbb{Q}}(H)$ has rank zero, that is, it admits no non-torsion divisor. To nevertheless exploit this technique, we make use of an idea introduced in [BBP25], that is, to replace H by an isomorphic curve over $\mathbb{F}_q$ which has positive rank over $\mathbb{Q}$.

If $H : y^2 = f(x)$ denotes a hyperelliptic curve, we defined its δ-twist to be the curve

$$H_\delta : \delta y^2 = f(x).$$

If its Jacobian variety has large Mordell–Weil rank and if δ is a square in $\mathbb{F}_q$ then the map $(x, y) \mapsto (x, \sqrt{\delta} y)$ is an explicit map from H to H_δ, called a birational equivalence, which induces an isomorphism between the associated Jacobian varieties over $\mathbb{F}_q$. Hence, the discrete logarithm problem of $\mathrm{Jac}_{\mathbb{F}_q}(H)$ may be reduced to that of $\mathrm{Jac}_{\mathbb{F}_q}(H_\delta)$; since the latter has large rank over $\mathbb{Q}$, a free set of small-height generators for $\mathrm{Jac}_{\mathbb{Q}}(H_\delta)$ can then be used in Regev's algorithm. Concretely, if $r = \operatorname{rank} \mathrm{Jac}_{\mathbb{Q}}(H_\delta)$, we can run Regev's algorithm with the parameter $d = r + 2$. We illustrate this in Figure 2.

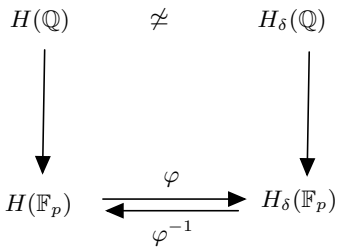

Fig. 2: Illustration of the method of lifting hyperelliptic curves from $\mathbb{F}_q$ to $\mathbb{Q}$.

To evaluate the effectiveness of our cryptographic attack, we consider the viewpoint of practical computations (Section 5.3). But first, we recall the main tools needed to study curves defined over the rationals.

5.2 Interlude on hyperelliptic curves defined over $\mathbb{Q}$

The Mordell–Weil theorem asserts the existence of an isomorphism $\mathrm{Jac}(H) \simeq T \times \mathbb{Z}^r$ for some finite group T and some integer r called the rank of the Jacobian variety or, by abuse of terminology, of H.

The Neron–Tate pairings $\langle , \rangle$, implemented in [Hol12] and in Magma [BCP97], gives $\mathrm{Jac}_{\mathbb{Q}}(H)$ the structure of a lattice. Given a set $\{b_1, \ldots, b_m\}$ of elements in $\mathrm{Jac}_{\mathbb{Q}}(H)$, the pairing matrix is defined as

$$M = \langle b_i, b_j \rangle_{\substack{1 \le i \le m \\ 1 \le j \le m}}.$$

To extract a set of generators of the subgroup generated by $\{b_1, \ldots, b_d\}$ one extracts a sub-matrix of M having the same rank as M. In particular, an element b of the Jacobian has finite order if and only if $\langle b, b \rangle = 0$.

From a theoretical point of view, to compute the generators of $\mathrm{Jac}_{\mathbb{Q}}(H)$ one must consider all divisors of norm up to the bound given by the BSD (see Magma's documentation for the Mordell-Weil function and [Zim96]). Once a set of generators is obtained, one applies Weil descent (see [Sch95,Gal01]). However the theoretical bound is typically prohibitively expensive to run on all but the very smallest curves of genus two.

We use an early abort strategy which allows us to obtain results of cryptographic relevance. It rests on the heuristic that, for a large proportion of the curves H, the generators of $\mathrm{Jac}_{\mathbb{Q}}(H)$ have well distributed sizes among those which are allowed by the proven upper bounds. Hence, if a Jacobian has large rank, it must admit at least one divisor whose norm is very small. Although not rigorously proven, this heuristic serves its purpose very well in practice, allowing us to derive several results of cryptographic significance. To this extent, we employed the following strategy:

1. Compute the rational points of the Jacobian variety of norm less than a given bound, typically 10^3 in our computations. This is done by Magma using Hensel's lift of the solutions of a Diophantine equation; see e.g. [Poo02].
2. Compute the Neron–Tate pairing on the points found and extract a set of generators.
3. This yields a lower bound on the rank of the Jacobian variety; if it is large enough, compute the full Mordell–Weil structure of the variety to find for potential larger divisors.

5.3 Results on a list of cryptographically relevant curves

The Buhler–Koblitz curve "4GLV127-BK". This curve has been considered for cryptographic use; see [BK98,BCM14]. It is defined as $H : y^2 = x^5 + 17$ over $\mathbb{F}_q$ where $q = 2^{64}(2^{63} - 27443) + 1$; its Jacobian variety $\mathrm{Jac}(H)$ has a 254-bit prime group order.

Using Magma [BCP97] we searched for "twist curves" of the form $H_\delta : y^2 = x^5 + 17\delta$ whose Jacobian variety has large rank over $\mathbb{Q}$. We proceeded as follows. First, for each integer $\delta \in \{1, \ldots, 128 \cdot 10^4\}$, we enumerated all rational points on $H_\delta = \mathrm{Jac}_{\mathbb{Q}}(H_\delta)$ of height less than 10^3 and extracted a reduced basis; this quickly gave us a lower bound on $\mathrm{rank}(J_\delta)$. Second, when this lower bound was two or greater, we determined the full Mordell–Weil group of J_δ and its rank under the GRH; a timeout was reached for three values of δ (2450, 131625, and 273904) which were consequently excluded. This entire computation ran on 128 cores for a total time of 448 960 seconds or about one hour of elapsed time. The distribution of $\mathrm{rank}(J_\delta)$ is as in Figure 3.

$\mathrm{rank}(\mathrm{Jac}_{\mathbb{Q}}(H_\delta); 10^3)$	0	1	2	3	4	5	6	7	≥ 8
$\#\{\delta\}$	1 265 371	14 282	191	89	50	9	3	2	0

Fig. 3: Early abort strategy while searching for large rank twists of the genus-two curve $y^2 = x^5 + 17$. The first row indicates the rank of the subgroup generated by the divisors of norm less than 10^3.

Since the multiplicative group $\mathbb{F}_q^\times$ of the finite field $\mathbb{F}_q$ is cyclic, at least one tenth of the possible values for δ are tenth powers in $\mathbb{F}_q$ and, as a consequence, yield a birational map

$$\varphi : \begin{cases} H \longrightarrow H_\delta \\ (x, y) \longmapsto (\delta^{1/5} x, \delta^{1/2} y) \end{cases}$$

defined over the base field. For instance, $\delta = 3576896$ is a tenth power in $\mathbb{F}_q$ and the variety $\mathrm{Jac}_{\mathbb{Q}}(H_\delta)$ has rank six. A basis in Mumford representation is as

follows:

$$
\begin{aligned}
b_1 &= (x - 8, -7800),\\
b_2 &= (x + 36, -584),\\
b_3 &= (x - 332, -2008392),\\
b_4 &= (x^2 - 77x - 3228, 911x + 21452),\\
b_5 &= (x^2 - \tfrac{165}{49}x - \tfrac{30636}{49}, \tfrac{109269}{343}x + \tfrac{37988}{343}),\\
b_6 &= (x^2 + \tfrac{48552}{529}x + \tfrac{4131648}{529}, -\tfrac{29522176}{36501}x + \tfrac{40433336}{12167}).
\end{aligned}
$$

This allows us to attack the discrete logarithm problem of $\mathrm{Jac}_{\mathbb{F}_q}(H)$ by transporting it to $\mathrm{Jac}_{\mathbb{F}_q}(H_\delta)$ where our generalized Regev's attack exploits these small, independent elements. This yields a DLP attack which, overall, is eight times faster than Shor's.

The Kawazoe-Takahashi pairing-friendly curves. Hyperelliptic curves of the form $y^2 = x^5 + ax$ have been proposed for cryptography, especially for pairing-based applications [KT08,Dry12]. In this example, we consider the curve with parameter $a = 3$ which is used for pairings constructions for various large primes q (see [KT08]). We search twists of the form $y^2 = x^5 + a\delta x$. Over finite fields where δ is a non-zero eighth power, due to the birational map

$$
\begin{cases}
y^2 = x^5 + \delta a x \longrightarrow y^2 = x^5 + ax\\
\qquad\qquad (x, y) \longmapsto \left(\delta^{1/4}x, \delta^{5/8}y\right),
\end{cases}
$$

the discrete logarithm problem of $\mathrm{Jac}_{\mathbb{F}_q}(H_\delta)$ is equivalent to that in $\mathrm{Jac}_{\mathbb{F}_q}(H)$.

Therefore, to attack such curves using Regev's algorithm, it suffices to find one which has large rank over the rationals. A search performed similarly to the above example exhibited the twist curve $y^2 = y^5 + 3\delta x$ for $\delta = 1027957$, that is,

$$y^2 = x^5 + 3083871x$$

of which the Jacobian variety has rank six over $\mathbb{Q}$. A basis in Mumford representation is as follows:

$$
\begin{aligned}
b_1 &= (x - 7, -4648),\\
b_2 &= (x - 25, -9320),\\
b_3 &= (\tfrac{1}{2}x^2 - \tfrac{1905}{32}x + \tfrac{121}{32}, \tfrac{83999}{128}x - \tfrac{33671}{128}),\\
b_4 &= (\tfrac{1}{2}x^2 - \tfrac{19325}{289}x + \tfrac{440553}{578}, \tfrac{61169405}{78608}x - \tfrac{1015474665}{78608}),\\
b_5 &= (\tfrac{1}{2}x^2 - \tfrac{6808057}{93312}x + \tfrac{3679375}{93312}, \tfrac{17702059859}{20155392}x + \tfrac{3431801275}{20155392}),\\
b_6 &= (\tfrac{1}{2}x^2 - \tfrac{154320553}{3317888}x + \tfrac{11013825}{3317888}, -\tfrac{1966303381237}{4273439744}x + \tfrac{1143254859885}{4273439744}).
\end{aligned}
$$

Consequently, we can attack the discrete logarithm problem in this curve up to eight times faster than Shor's algorithm.

Drylo's pairing-friendly genus-2 curves with split Jacobian Drylo [Dry12] proposed the use of curves C whose Jacobian J is isogenous to the product of two

elliptic curves, i.e. there exist a number field K, an integer N and two elliptic curves E and E' and a isogeny ϕ such that

$$\phi : J \to E \times E'$$

and $\ker \phi \simeq \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$.

For example, the second curve C in [Dry12, Ex 12] is

$$\begin{cases} q = 125642457939801322085590357749816450419191871179750 \\ \qquad 5107936025480666612044658 73 \\ H/\mathbb{F}_q : \quad y^2 = x^6 + \frac{8}{5}x^3 + \frac{128}{3125}. \end{cases}$$

Note that the coefficients are elements of $K = \mathbb{Q}(\sqrt{-3})$ of small numerator and denominator, which happen to be in $\mathbb{Q}$. We used the rational reconstruction to find them from the values in $\mathbb{F}_q$ given in [Dry12]. The automorphism group of the Jacobian, computed in Magma, ensures the existence of the isogeny ϕ. We omit to give a twist of high rank because the computation of the Mordell-Weil group is not yet implemented for quadratic fields.

Small characteristic genus-2 curves In the embedded HECC implementations, one prefers the curves defined over finite fields $\mathbb{F}_q = \mathbb{F}_{2^n}$, e.g. [FBV08,GT17]. To speed up the computations, Köblitz [Kob88] proposed to use hyperelliptic curves which are base-changed from $\mathbb{F}_2$, i.e. curves of equation $y^2 + yh(x) = f(x)$ with $f, h \in \mathbb{F}_2[x]$. Köblits's work preceded the sub-exponential classical algorithm [ADH94] so he proposed the curves $y^2 + yx = x^{2g+1}$ with g taking values larger than two. Sakai and Sakurai [SS98, Sec 5.1] proposed the genus two curves

$$y^2 + (x^2 + x + 1)y = x^5 + x + 1,$$

which are seen as having coefficients in $\mathbb{F}_{2^n}$ for various values of n, e.g. $n = 89$. Similarly to [BBP25] we propose to see the curve as having an equation in x and y with coefficient in $\mathbb{F}_2(t)$. One searches for twists with large Mordell-Weil rank. We omit to propose values of the twists for specific examples because the Mordell-Weil group computation is currently unavailable in Magma.

The Regev-resistant curve of Gaudry–Schost. We note that our strategy does not apply to hyperelliptic curves whose coefficients are chose pseudo randomly in large finite fields $\mathbb{F}_q$ such as the Gaudry–Schost curve [GS04, Appendix A] implemented in [RSSB16]. Indeed, such large coefficients impede on the search for twists of large rank with small generators.

We conclude this section with the remark that, apart from the Gaudry–Schost curve, our extensive review of the literature shows that hyperelliptic curves relevant to cryptographic applications are:

- $y^2 = x^5 + b$ over $\mathbb{Q}$ with small values of b, see [BK98,BMS05];
- $y^2 = x^5 + ax$ over $\mathbb{Q}$ with small values of a, see [KT08];
- $y^2 + yh(x) = f(x)$ over $\mathbb{F}_{2^n}$ with $f, h \in \mathbb{F}_2[x]$, see [Kob88];
- genus two curves of split Jacobian, see [Dry12].

6 Conclusion and open questions

Prior work had established the complexity of Regev's algorithm for integer factorisation and the DLP in multiplicative groups of finite fields. The present work analyses it for the DLP of hyperelliptic curve of medium genus. Our results suggest that other alternatives to and variants of Shor's algorithm may have a similar complexity and open the question of proving them rigorously.

On the practical side, the idea of using small points, central to applying Regev's algorithm, proves fruitful and may be exploited in a practical implementation on a classical simulator of quantum computers, which are already available. Indeed, some implementations of quantum memory have an access cost which depends on the size of the memory; see [Gid19]. By using small points, one reduces the memory requirement. Furthermore, the use of multi-scalar products might also be interesting as a practical improvement when implementing other QFT-based algorithms.

While quantum implementations remain at their infancy, it is valuable to create an implementation of Regev's algorithm. On the one hand, it is an open question whether it can be combined with Ekerå's algorithm. On the other hand, Shor's and Ekerå's algorithm avoid the use of a double-and-add algorithm with non-constant basis. Its implementation is challenging because, on a quantum computer, doubling also requires halving.

Finally, the twist strategy outlined above allows us to obtain almost the full speedup of Regev's algorithm with respect to Shor's: for the cryptographic examples given above, the practical speedups are only about half the optimal value of $\sqrt{256} = 16$. Note also that those practical speedups are greater than those obtained for elliptic curves in [BBP25]. This technique rests on deep number theoretical questions such as estimating the proportion of Jacobian varieties of genus-two curves with rank $r \geq 2$, and to construct such curves achieving large values of r. Even when limited to the set of twists of a given curve, this question remains open.

References

ADH94. Leonard M Adleman, Jonathan DeMarrais, and Ming-Deh Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In Algorithmic Number Theory – ANTS-I, volume 877 of Lecture Notes in Computer Science, pages 28–40. Springer, 1994.

AFK24. Mónica P Arenas, Georgios Fotiadis, and Elisavet Konstantinou. Special TNFS-secure pairings on ordinary genus 2 hyperelliptic curves. In Progress in Cryptology – AFRICACRYPT 2024, volume 14861 of Lecture Notes in Computer Science, pages 285–310. Springer, 2024.

Bac90. Eric Bach. Explicit bounds for primality testing and related problems. Mathematics of Computation, 55(191):355–380, 1990.

BBP24. Razvan Barbulescu, Mugurel Barcau, and Vicenţiu Paşol. A comprehensive analysis of Regev's quantum algorithm on elliptic curves. Cryptology ePrint Archive, 2024.

BBP25. Razvan Barbulescu, Mugurel Barcau, and Vicenţiu Paşol. Extending Regev's quantum algorithm to elliptic curves. In Progress in Cryptology – LATINCRYPT 2025, volume 16129 of Lecture Notes in Computer Science, pages 234 – 266. Springer, 2025.

BCHL16. Joppe W Bos, Craig Costello, Huseyin Hisil, and Kristin Lauter. Fast cryptography in genus 2. Journal of Cryptology, 29(1):28–60, 2016.

BCM14. Joppe Bos, Craig Costello, and Andrea Miele. Elliptic and hyperelliptic curves: a practical security analysis. In Public-Key Cryptography – PKC 2014, volume 8383 of Lecture Notes in Computer Science, pages 203–220, 2014.

BCP97. Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system: the user language. Journal of Symbolic Computation, 24(3–4):235–265, 1997.

Ber06. Dan Bernstein. Elliptic vs. hyperelliptic, part 1., 2006. Elliptic Curve Cryptography (ECC) 2006. Fields Institute, Toronto, Canada.

BK98. Joe Buhler and Neal Koblitz. Lattice basis reduction, Jacobi sums and hyperelliptic cryptosystems. Bulletin of the Australian Mathematical Society, 58:147–154, 1998.

BMS05. Ezra Brown, Bruce T Myers, and Jerome A Solinas. Hyperelliptic curves with compact parameters. Designs, Codes and Cryptography, 36(3):245–261, 2005.

BS25. Peter Birkner and Peter Schwabe. Documentation of the Hector library. Available online at `https://cryptosith.org/crypto/data/hector_documentation.pdf`, 2005-2025.

Can87. David G Cantor. Computing in the Jacobian of a hyperelliptic curve. Mathematics of Computation, 48(177):95–101, 1987.

CFA+05. Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. Handbook of elliptic and hyperelliptic curve cryptography. CRC press, 2005.

CGHZ23. Chao Chen, Peidong Guan, Yan Huang, and Fangguo Zhang. Quantum circuits for hyperelliptic curve discrete logarithms over the Mersenne prime fields. Quantum Information Processing, 22(7):274, 2023.

Dry12. Robert Dryło. Constructing pairing-friendly genus 2 curves with split Jacobian. In Progress in Cryptology – INDOCRYPT 2012, volume 7668 of Lecture Notes in Computer Science, pages 431–453, 2012.

EG24. Martin Ekerå and Joel Gärtner. Extending Regev's factoring algorithm to compute discrete logarithms. In Post-Quantum Cryptography–PQC 2024, volume 14772 of Lecture Notes of Computer Science, pages 211–242. Springer, 2024.

EG25. Martin Ekerå and Joel Gärtner. A high-level comparison of state-of-the-art quantum algorithms for breaking asymmetric cryptography. IACR Communications in Cryptology, 2(1), 2025.

EH17. Martin Ekerå and Johan Håstad. Quantum algorithms for computing short discrete logarithms and factoring RSA integers. In Tanja Lange and Tsuyoshi Takagi, editors, Post-Quantum Cryptography – PQC 2017, volume 10346 of Lecture Notes in Computer Science, pages 347–363. Springer International Publishing, 2017.

Eke19. Martin Ekerå. Revisiting Shor's quantum algorithm for computing general discrete logarithms. arXiv preprint arXiv:1905.09084, 2019.

Eke24. Martin Ekerå. On the success probability of quantum order finding. ACM Transactions on Quantum Computing, 5(2):1–40, 2024.

Eng02. Andreas Enge. Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. Mathematics of Computation, 71(238):729–742, 2002.

FBV08. Junfeng Fan, Lejla Batina, and Ingrid Verbauwhede. HECC goes embedded: an area-efficient implementation of HECC. In International workshop on selected areas in cryptography, pages 387–400. IEEE, 2008.

FR10. Dmitry Faifman and Zeév Rudnick. Statistics of the zeros of zeta functions in families of hyperelliptic curves over a finite field. Compositio Mathematica, 146(1):81–101, 2010.

FS11. David Mandell Freeman and Takakazu Satoh. Constructing pairing-friendly hyperelliptic curves using Weil restriction. Journal of Number Theory, 131(5):959–983, 2011.

FT19. E Victor Flynn and Yan Bo Ti. Genus two isogeny cryptography. In Post-Quantum Cryptography – PQC 2019, volume 11505 of Lecture Notes in Computer Science, pages 286–306. Springer, 2019.

Gal01. Steven D Galbraith. Weil descent of Jacobians. Electronic Notes in Discrete Mathematics, 6:459–468, 2001.

Gau00. Pierrick Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In Advances in cryptology–EUROCRYPT 2000, pages 19–34. Springer, 2000.

GCT17. Gabriel Gallin, Turku Ozlum Celik, and Arnaud Tisserand. Architecture level optimizations for kummer based hecc on fpgas. In Progess in Cryptology – INDOCRYPT 2017, volume 10698 of Lecture Notes in Computer Science, pages 44–64. Springer, 2017.

Gid19. Craig Gidney. Windowed quantum arithmetic. arXiv preprint arXiv:1905.07682, 2019.

Gid25. Craig Gidney. How to factor 2048 bit RSA integers with less than a million noisy qubits. Quantum Physics arXiv preprint, 2025. Available online at `https://arxiv.org/abs/2505.15917`.

GS04. Pierrick Gaudry and Éric Schost. Construction of secure random curves of genus 2 over prime fields. In Advances in Cryptology–EUROCRYPT 2004, volume 3027 of Lecture Notes in Computer Science, pages 239–256. Springer, 2004.

GT17. Gabriel Gallin and Arnaud Tisserand. Hardware architectures for HECC. In CryptArchi 2017: 15th International Workshops on Cryptographic architectures embedded in logic devices, 2017.

Hol12. David Holmes. Computing Néron–Tate heights of points on hyperelliptic Jacobians. Journal of Number Theory, 132(6):1295–1305, 2012.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. NTRU: A ring-based public key cryptosystem. In Algorithmic number theory – ANTS III, volume 1423 of Lecture Notes in Computer Science, pages 267–288. Springer, 1998.

HvdH21. David Harvey and Joris van der Hoeven. Integer multiplication in time $O(n \log n)$. Annals of Mathematics, 193(2):563–617, 2021.

Kob87. Neal Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48(177):203–209, 1987.

Kob88. Neal Koblitz. A family of jacobians suitable for discrete log cryptosystems. In Advances in Cryptology – CRYPTO '88, volume 403 of Lecture Notes in Computer Science, pages 94–99. Springer, 1988.

Kob89. Neal Koblitz. Hyperelliptic cryptosystems. Journal of Cryptology, 1:139–150, 1989.

KT08. Mitsuru Kawazoe and Tetsuya Takahashi. Pairing-friendly hyperelliptic curves with ordinary Jacobians of type $y^2 = x^5 + ax$. In Pairing-Based Cryptography – Pairing 2008, volume 5209 of Lecture Notes in Computer Science, pages 164–177. Springer, 2008.

Mil85. Victor S Miller. Use of elliptic curves in cryptography. In Conference on the theory and application of cryptographic techniques, pages 417–426. Springer, 1985.

Moo21. Dustin Moody. Nist PQC standardization update, 2021. Available online at `https://nist-pqcrypto-org.viacache.net/foia/20221014/pqcrypto-sept2020-moody.pdf`.

MST99. Volker Müller, Andreas Stein, and Christoph Thiel. Computing discrete logarithms in real quadratic congruence function fields of large genus. Mathematics of Computation, 68(226):807–822, 1999.

Nat23. National Institute of Science and Technology. Special publication NIST 800-186: Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters, 2023. Available online at `https://csrc.nist.gov/pubs/sp/800/186/iñAnal`.

Nat25. National Cyber Security Centre. Timelines for migration to post-quantum cryptography, 2025. Available online at `https://www.ncsc.gov.uk/pdfs/guidance/pqc-migration-timelines.pdf`.

NTRKR$^+$24. Ramireddy Nava Teja Reddy, M Kavitha, G Sudarsana Reddy, Amr Yousef, Kareem M AboRas, Ahmed Emara, and Ch Rami Reddy. Mayfly optimistic hyperelliptic curve cryptosystem. Frontiers in Computer Science, 6:1381850, 2024.

Pil24. Cédric Pilatte. Unconditional correctness of recent quantum algorithms for factoring and computing discrete logarithms. arXiv preprint 2404.16450, 2024.

Pip80. Nicholas Pippenger. On the evaluation of powers and monomials. SIAM Journal on Computing, 9(2):230–250, 1980.

Poo02. Bjorn Poonen. Computing rational points on curves. In Surveys in Number Theory, pages 223–246. AK Peters/CRC Press, 2002.

Rag24. Seyoon Ragavan. Regev factoring beyond Fibonacci: Optimizing prefactors. Cryptology ePrint Archive, 2024. Available online at `https://eprint.iacr.org/2024/636`.

Reg09. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM, 56(6):1–40, 2009.

Reg25. Oded Regev. An efficient quantum factoring algorithm. Journal of the ACM, 72(1):1–13, 2025.

Ros02. Michael Rosen. Number theory in function fields, volume 210. Springer Science & Business Media, 2002.

RSSB16. Joost Renes, Peter Schwabe, Benjamin Smith, and Lejla Batina. Kummer: Efficient hyperelliptic signatures and key exchange on microcontrollers. In Cryptographic Hardware and Embedded Systems–CHES 2016, volume 9813 of Lecture Notes in Computer Science, pages 301–320. Springer, 2016.

RV24. Seyoon Ragavan and Vinod Vaikuntanathan. Space-efficient and noise-robust quantum factoring. In Advances in Cryptology – CRYPTO 2024, volume 14925 of Lecture Notes in Computer Science, pages 107–140. Springer, 2024.

Sch95. Edward F Schaefer. 2-descent on the Jacobians of hyperelliptic curves. Journal of Humber Theory, 51(2):219–232, 1995.

SRS+23. Animesh Srivastava, Bhupender Singh Rawat, Gulbir Singh, Vivek Bhatnagar, and Shiv Ashish Dhondiyal. A review of the authentication scheme using hecc and ecc. In 2023 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), pages 1–5. IEEE, 2023.

SS98. Yasuyuki Sakai and Kouichi Sakurai. Design of hyperelliptic cryptosystems in small characteristic and a software implementation over $\mathbb{F}_{2^n}$. In Advances in Cryptology – ASIACRYPT '98, volume 1514 of Lecture Notes in Computer Science, pages 80–94. Springer, 1998.

The98. The National Institute of Standards and Technology. X9.62-1998 public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ecdsa). Available online at `https://safecurves.cr.yp.to/grouper.ieee.org/groups/1363/private/x9-62-09-20-98.pdf`, 1998.

The99. The National Institute of Standards and Technology. Digital signature standard. Available online at `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf`, 1999.

Vol00. José Felipe Voloch. Jacobians of curves over finite fields. Rocky Montains Journal of Mathematics, 30(2), 2000.

vzGG03. Joachim von zur Gathen and Jürgen Gerhard. Modern computer algebra. Cambridge University Press, 2003.

WLL+25. Siyi Wang, Xiufan Li, Wei Jie Bryan Lee, Suman Deb, Eugene Lim, and Anupam Chattopadhyay. A comprehensive study of quantum arithmetic circuits. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 383(2288):20230392, 2025.

YSG24. Anu Yadav, Prabha Sharma, and Yogita Gigras. A comparative study of elliptic curve and hyperelliptic curve cryptography methods and an overview of their applications. In International Conference on Intelligent Systems for Cybersecurity –ISCS 2024, pages 01–06. IEEE, 2024.

Zim96. Horst G Zimmer. Basic algorithms for elliptic curves. Number theory (Eger 1996), pages 541–595, 1996.