11-1-2020

Efficient elliptic curve Diffie-Hellman computation at the 256-bit security level

Kaushik Nath
Indian Statistical Institute, Kolkata

Palash Sarkar
Indian Statistical Institute, Kolkata

IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Efficient elliptic curve Diffie-Hellman computation at the 256-bit security level

ISSN 1751-8709
Received on 26th November 2019
Revised 20th April 2020
Accepted on 23rd April 2020
E-First on 24th July 2020
doi: 10.1049/iet-ifs.2019.0620
www.ietdl.org

Kaushik Nath[1] ✉, *Palash Sarkar*[1]

[1]*Applied Statistics Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata, India*

✉ *E-mail: kaushikn_r@isical.ac.in*

Abstract: In this study, the authors introduce new Montgomery and Edwards form elliptic curves targeted at the 256-bit security level. To this end, they work with three primes, namely $p_1 := 2^{506} - 45$, $p_2 := 2^{510} - 75$ and $p_3 := 2^{521} - 1$. While p_3 has been considered earlier in the literature, p_1 and p_2 are new. They define a pair of birationally equivalent Montgomery and Edwards form curves over all the three primes. Efficient 64-bit assembly implementations targeted at Skylake and later generation Intel processors have been made for the shared secret computation phase of the Diffie-Hellman key agreement protocol for the new Montgomery curves. Curve448 of the Transport Layer Security, Version 1.3 is a Montgomery curve which provides security at the 224-bit security level. Compared to the best publicly available 64-bit implementation of Curve448, the new Montgomery curve over p_1 leads to a 3–4% slowdown and the new Montgomery curve over p_2 leads to a 4.5–5% slowdown; on the other hand, 29 and 30.5 extra bits of security, respectively, are gained. For designers aiming for the 256-bit security level, the new curves over p_1 and p_2 provide an acceptable trade-off between security and efficiency.

1 Introduction

One of the most extensively used modern cryptographic primitives is the Diffie-Hellman (DH) [1] key agreement protocol. Koblitz [2] and Miller [3] have independently shown that the DH protocol can be instantiated using cyclic groups arising from the theory of elliptic curves. Among the various models of elliptic curves, the Montgomery form [4] provides the most efficient model for implementing DH key agreement. The famous and widely deployed Curve25519 [5] is a Montgomery form curve. As part of the Transport Layer Security (TLS) protocol, Version 1.3 [6], RFC 7748 [7] specifies two elliptic curves, namely Curve25519 and Curve448, for DH key agreement. Curve25519 provides security at the 128-bit security level and Curve448 provides security at the 224-bit security level.

Various cryptographic primitives targeted at the 256-bit security level have been proposed in the literature. For example, both SHA-2 and SHA-3 have variants for the 256-bit security level [8]. In the context of public key cryptography, there are proposals for cryptographic pairings targeted at the 256-bit security level [9, 10]. A general purpose elliptic curve called E-521 has been proposed in [11] for the 256-bit security level.

In view of the above discussion, design and implementation of ECDH key agreement protocol at the 256-bit security level is a relevant research problem. TLS, Version 1.3, however, does not include a 256-bit secure solution. A possible reason for this omission is the apprehension that the computation of key agreement at the 256-bit security level will be significantly slower than that at the 224-bit security level. While, there will indeed be a slowdown, to the best of our knowledge, the magnitude of this slowdown is presently unknown. Consequently, it is not clear whether such a slowdown is an acceptable trade-off for achieving higher security.

We consider the following four primes: $2^{506} - 45$, $2^{510} - 75$, $2^{521} - 1$ and $2^{448} - 2^{224} - 1$. For convenience of notation, we will denote $2^{506} - 45$ as $p506{-}45$, $2^{510} - 75$ as $p510{-}75$, $2^{521} - 1$ as $p521{-}1$ and $2^{448} - 2^{224} - 1$ as $p448{-}224{-}1$. Fix a prime p. Given $A \in \mathbb{F}_p \setminus \{-2, 2\}$ and $B \in \mathbb{F}_p \setminus \{0\}$, the Montgomery curve $E_{M,A,B}$ over $\mathbb{F}_p$ is given by the equation $E_{M,A,B}: By^2 = x^3 + Ax^2 + x$. Given $a, d \in \mathbb{F}_p \setminus \{0\}$ and $a \neq d$, the twisted Edwards curve $E_{E,a,d}$ is given by the equation $E_{E,a,d}: au^2 + v^2 = 1 + du^2v^2$. For convenience of notation, a Montgomery curve $E_{M,A,1}$ will be denoted as $M[A]$; an Edwards curve $E_{E,1,d}$ will be denoted as $E[d]$; and a twisted Edwards curve $E_{E,-1,d}$ will be denoted as $\widetilde{E}[d]$. If we wish to emphasise the underlying prime p, then we will write $M[p, A]$, $E[p, d]$ and $\widetilde{E}[p, d]$ instead of $M[A]$, $E[d]$ and $\widetilde{E}[d]$, respectively.

Our contributions: In this work, we propose new curves at 256-bit security level and perform efficient 64-bit implementation of ECDH key agreement. A summary of the new curves is given in Table 1. Also, for comparison, we include the two curves $M[156326]$ and $E[39082/39081]$ at the 224-bit security level which are part of TLS, Version 1.3. The curve $M[156326]$ has been named Curve448 in [7].

The prime $p521{-}1$ has been considered earlier in [12] which proposed the Weierstrass form curve P-521. This prime was later considered in [11] which introduced the curve $E[p521{-}1, {-}376014]$ (and named it E-521) as part of a suite of general purpose high security elliptic curves. Using the isogenies given in [13], it can be shown that E-521 is 4-isogenous to $M[1504058]$ shown in Table 1. To the best of our knowledge, neither of the curves $M[1504058]$ or $E[376015/376014]$ appear earlier in the literature. Also, to the best of our knowledge, the primes $p506{-}45$ and $p510{-}75$ have not been considered earlier in the literature and so the question of proposing curves over the corresponding fields does not arise.

From Table 1, we observe that $M[p506{-}45, 996558]$, $M[p510{-}75, 952902]$ and $M[p521{-}1, 1504058]$ provide 29, 30.5 and 36.5 bits more security compared to $M[p448{-}224{-}1, 156326]$ (i.e. Curve448).

To assess the performance of the new curves, we have carried out a 64-bit assembly implementation of the DH shared secret computation over the new Montgomery curves. Field elements are represented using a number of 64-bit words or limbs. Our target processors were the Skylake and later generation Intel processors. So, we chose the packed or saturated limb representation of field elements. Further details of field representation are provided in Section 4.

Timing measurements were taken on the Skylake and the Kaby Lake processors. For comparison, we have considered the best previously reported [14] 64-bit implementation of the shared secret computation phase of the DH protocol over Curve448 on the

Table 1 Montgomery and Edwards curves at the 256-bit security level proposed in this work along with Curve448 of TLS, Version 1.3. In the table, $M[156326]$ is Curve448

Prime	Security	Mont	Base Pt (Mont)	Ed	Base Pt (Ed)
$p448-224-1$	223	$M[156326]$	$(5, \cdot)$	$E[39082/39081]$	$(\cdot, -3/2)$
$p506-45$	252	$M[996558]$	$(3, \cdot)$	$E[249140/249139]$	$(\cdot, 2)$
$p510-75$	253.5	$M[952902]$	$(4, \cdot)$	$\tilde{E}[-238225/238226]$	$(\cdot, 3/5)$
$p521-1$	259.5	$M[1504058]$	$(8, \cdot)$	$E[376015/376014]$	$(\cdot, 9/7)$

Skylake and Kaby Lake processors. Detailed cycle counts are reported later. Below we summarise the main findings. The following statements refer to the DH shared secret computations over the mentioned curves.

i. $M[p506-45, 996558]$ is about 1.3–1.4% faster than $M[p510-75, 952902]$.
ii. $M[p506-45, 996558]$ is about 19% faster than $M[p521-1, 1504058]$.
iii. $M[p506-45, 996558]$ is about 3–4% slower than Curve448.
iv. $M[p510-75, 952902]$ is about 4.5–5% slower than Curve448.
v. $M[p521-1, 1504058]$ is about 21–22% slower than Curve448.

While $M[p521-1, 1504058]$ provides 36.5 bits of extra security compared to Curve448, the slowdown is also quite significant. On the other hand, $M[p506-45, 996558]$ and $M[p510-75, 952902]$ provide 29 and 30.5 bits of extra security compared to Curve448 and the slowdowns for these curves are much less marked. So, if security around the 256-bit security level is desired, either of the curves $M[p506-45, 996558]$ or $M[p510-75, 952902]$ seem to provide a reasonable trade-off between speed and security.

The curve $E[p506-45, 249140/249139]$ which is birationally equivalent to $M[p506-45, 996558]$ can be used for the key generation phase. The small base point on $E[p506-45, 249140/249139]$ is helpful for fixed base scalar multiplication. Also, curve $E[p506-45, 249140/249139]$ can be used to implement a signature scheme following the approach used for EdDSA [15]. Similarly, if the curve $M[p510-75, 952902]$ is used for shared secret computation of the DH protocol, then the curve $\tilde{E}[p510-75, -238225/238226]$ which is birationally equivalent to $M[p510-75, 952902]$ can be used for key generation and also for instantiation of a signature scheme following [15].

We have made the source codes of our implementations publicly available at the following link: https://github.com/kn-cs/mont256-dh.

2 Montgomery and (twisted) Edwards form elliptic curves

Let p be a prime and $\mathbb{F}_p$ be the finite field of p elements. Following TLS, Version 1.3, we consider elliptic curves over $\mathbb{F}_p$, where p is a large prime. Montgomery curve $E_{M,A,B}$ and twisted Edwards curve $E_{E,a,d}$ have already been defined. In our applications, we will have $B = 1$ and a to be either 1 or -1. If $a = 1$, then the corresponding curve is simply called an Edwards form curve (instead of twisted Edwards form curve). If a is a square and d is not a square in $\mathbb{F}_p$, then the addition formula in $E_{E,a,d}$ is complete [16]. In this case, $E_{E,a,d}$ is called a complete twisted Edwards curve. Further, if $a = -1$, then particularly efficient addition formulas are known [17].

If $p \equiv 1 \bmod 4$, -1 is a square modulo p. In this case, if d is a non-square, the addition formula over $E_{E,-1,d}$ is both complete and the fastest. On the other hand, if $p \equiv 3 \bmod 4$, -1 is a non-square modulo p and so the addition formula over $E_{E,-1,d}$ is not guaranteed to be complete. In this case, one considers the Edwards curve $E_{E,1,d}$ with d a non-square so that the addition formula is complete. It is not, however, the fastest. If the base point on $E_{E,1,d}$ is small, then the difference in the number of operations between the addition formulas on $E_{E,-1,d}$ and $E_{E,1,d}$ is small. More concretely, if the base point on $E_{E,1,d}$ is $(\cdot, 2)$, then this difference is just two left shifts. See [18] for details.

For $p \equiv 3 \bmod 4$, addition formula over $E_{E,-1,d}$ is not guaranteed to be complete making constant time implementation of scalar multiplication problematic. On the other hand, for the verification phase of a signature scheme based on the EdDSA template [15], constant time implementation is not an issue. For this application, one may move from $E_{E,1,d}$ to $E_{E,-1,d'}$, for some d' (see below) using a birational equivalence and perform the main computation of signature verification over $E_{E,-1,d'}$.

We refer to [4, 19, 20] for background theory and further details about Montgomery form curves. For (twisted) Edwards curves, we refer to [16, 21, 22].

2.1 Montgomery-Edwards connection

RFC7748 [7] of TLS, Version 1.3 specifies both Montgomery and Edwards form curves for a given security level. In the present state of knowledge, the shared secret computation of the DH key agreement is performed best on a Montgomery form curve. On the other hand, the key generation phase as well as the computations required for an elliptic curve signature scheme based on the template in [15] are performed best on an Edwards form curve.

Edwards and Montgomery curves can be connected by either birational equivalences or by isogenies. For example, for the 128-bit security level, Curve25519 and Ed25519 are birationally equivalent. Similarly, at the 224-bit security level, Curve448 (i.e. $M[p448\text{-}224\text{-}1, 156326]$) and $E[p448\text{-}224\text{-}1, 39082/39081]$ are birationally equivalent. Additionally, Curve448 is 4-isogenous to $E[p448\text{-}224\text{-}1, -39081]$ [7]. The curve $E[p448\text{-}224\text{-}1, -39081]$ was proposed in [23] where it was named Ed448-Goldilocks and it has been called Edwards448 in [7].

We provide below some explicit birational equivalences between Montgomery and Edwards form curves. These can be obtained by composing the elementary birational equivalences provided in [16, 22]. The verification of these birational equivalences, on the other hand, can be done by direct substitution.

Case $p \equiv 3 \bmod 4$*:* Let $E_{M,A,B}: By^2 = x^3 + Ax^2 + x$ be a Montgomery curve and $E_{E,1,d}: u^2 + v^2 = 1 + du^2v^2$ be an Edwards curve over $\mathbb{F}_p$. Note that -1 is not a square in $\mathbb{F}_p$.

i. If $(A + 2)/B$ is a square in $\mathbb{F}_p$, then the map

$$(x, y) \mapsto (u, v) = \left(\frac{\delta x}{y}, \frac{x - 1}{x + 1}\right) \tag{1}$$

where $\delta^2 = (A + 2)/B$ is a birational equivalence from $E_{M,A,B}$ to $E_{E,1,d}$ with exceptional points $y = 0$ and $x = -1$. Conversely, the map

$$(u, v) \mapsto (x, y) = \left(\frac{1 + v}{1 - v}, \frac{\delta(1 + v)}{u(1 - v)}\right) \tag{2}$$

is a birational equivalence from $E_{E,1,d}$ to $E_{M,A,B}$ with exceptional points $u = 0$ and $v = 1$. The relation between A and d is $(A + 2)/4 = 1/(1 - d)$.

ii. If $(A - 2)/B$ is a square in $\mathbb{F}_p$, then the map

$$(x, y) \mapsto (u, v) = \left(\frac{\delta x}{y}, \frac{x + 1}{x - 1}\right) \tag{3}$$

where $\delta^2 = (A-2)/B$ is a birational equivalence from $E_{M,A,B}$ to $E_{E,1,d}$ with exceptional points $y = 0$ and $x = 1$. Conversely, the map

$$(u, v) \mapsto (x, y) = \left(\frac{v+1}{v-1}, \frac{\delta(v+1)}{u(v-1)}\right) \tag{4}$$

is a birational equivalence from $E_{E,1,d}$ to $E_{M,A,B}$ with exceptional points $u = 0$ and $v = 1$. The relation between A and d is $(A-2)/4 = 1/(d-1)$.

Suppose that d is not a square so that the addition formula over $E_{E,1,d}$ is complete. Since both d and -1 are not squares, $-d$ is a square. So, the map

$$(u, v) \mapsto (\hat{u}, \hat{v}) = \left(\gamma u, \frac{1}{v}\right) \tag{5}$$

where $-\gamma^2 = d$ is a birational equivalence with exceptional points $v = 0$ from the Edwards curve $E_{E,1,d}: u^2 + v^2 = 1 + du^2v^2$ to the twisted Edwards curve $E_{E,-1,-1/d}: -\hat{u}^2 + \hat{v}^2 = 1 + (-1/d)\hat{u}^2\hat{v}^2$.

Case $p \equiv 1 \bmod 4$: Let $E_{M,A,B}: y^2 = x^3 + Ax^2 + x$ be a Montgomery curve and $E_{E,-1,d}: -u^2 + v^2 = 1 + du^2v^2$ be an Edwards curve over $\mathbb{F}_p$. Note that -1 is a square in $\mathbb{F}_p$.

i. If $(A+2)/B$ is a square in $\mathbb{F}_p$, then the map

$$(x, y) \mapsto (u, v) = \left(\frac{\delta x}{y}, \frac{x-1}{x+1}\right) \tag{6}$$

where $-\delta^2 = (A+2)/B$ is a birational equivalence from $E_{M,A,B}$ to $E_{E,-1,d}$ with exceptional points $y = 0$ and $x = -1$. Conversely, the map

$$(u, v) \mapsto (x, y) = \left(\frac{1+v}{1-v}, \frac{\delta(1+v)}{u(1-v)}\right) \tag{7}$$

is a birational equivalence from $E_{E,-1,d}$ to $E_{M,A,B}$ with exceptional points $u = 0$ and $v = 1$. The relation between A and d is $(A+2)/4 = 1/(1+d)$.

ii. If $(A-2)/B$ is a square in $\mathbb{F}_p$, then the map

$$(x, y) \mapsto (u, v) = \left(\frac{\delta x}{y}, \frac{x+1}{x-1}\right) \tag{8}$$

where $-\delta^2 = (A-2)/B$ is a birational equivalence from $E_{M,A,B}$ to $E_{E,-1,d}$ with exceptional points $y = 0$ and $x = 1$. Conversely, the map

$$(u, v) \mapsto (x, y) = \left(\frac{v+1}{v-1}, \frac{\delta(v+1}{u(v-1)}\right) \tag{9}$$

is a birational equivalence from $E_{E,-1,d}$ to $E_{M,A,B}$ with exceptional points $u = 0$ and $v = 1$. The relation between A and d is $(A-2)/4 = -1/(d+1)$.

2.2 Security properties

Let n and n_T be the orders of $E(\mathbb{F}_p)$ and its quadratic twist, respectively. Let ℓ and h (resp. ℓ_T and h_T) be such that $n = h \cdot \ell$ (resp. $n_T = h_T \cdot \ell_T$). Suppose that ℓ and ℓ_T are primes. Cryptography is done over an ℓ-order subgroup of $E(\mathbb{F}_p)$.

The parameters h and h_T are called the co-factors of the curve and its twist, respectively. For a Montgomery curve, the curve order n is a multiple of 4. Also, $n + n_T = 2(p+1)$. From this, it is easy to argue that if $p \equiv 3 \bmod 4$, then the minimum value of (h, h_T) is (4, 4), while if $p \equiv 1 \bmod 4$, then the minimum value of (h, h_T) is either (8, 4) or (4, 8).

The embedding degrees k and k_T of the curve and its twist are defined as follows. The parameter k (resp. k_T) is the smallest positive integer such that $\ell \mid (p^k - 1)$ (resp. $\ell_T \mid (p^{k_T} - 1)$).

The complex multiplication field discriminant D of E is defined in the following manner. Let $t = p + 1 - n$. By Hasse's theorem, $|t| \leq 2\sqrt{p}$ and in the cases that we considered $|t| < 2\sqrt{p}$ so that $t^2 - 4p$ is a negative integer; let s^2 be the largest square dividing $t^2 - 4p$; define $D = (t^2 - 4p)/s^2$ if $t^2 - 4p \bmod 4 = 1$ and $D = 4(t^2 - 4p)/s^2$ otherwise.

SafeCurves [24] suggests that all of the parameters ℓ, ℓ_T, k, k_T and D should be large to ensure security against various known attacks. Considering twist security, the security level of a curve in terms of bits is defined to be (1/2) min $(\log_2 \ell, \log_2 \ell_T)$.

3 Curves for the 256-bit security level

Since our target is 256-bit security, we need a κ-bit prime where κ is about 512. Further, we chose to work over (pseudo-)Mersenne primes $2^m - \delta$ with δ small, so that we can leverage the efficient algorithms for arithmetic modulo such primes.

The prime $p521-1$ is a Mersenne prime and has been suggested earlier for defining elliptic curves [11, 12]. This prime provides a few bits more security than our target 256-bit security level. So, we considered some pseudo-Mersenne primes which are less than 2^{512}. We found two pseudo-Mersenne primes less than 2^{512} which may be considered for 256-bit security. These are $p506$-45 and $p510$-75. Another pseudo-Mersenne prime $2^{511}-187$ (denoted as $p511$-187) has been earlier suggested in [11]. In Section 8.2 of the Appendix, we discuss our reasons for not considering this prime.

The curve search methodology that was used is the following. Let $\alpha = (A+2)/4$. The quantity α is used in the Montgomery ladder computation and a small value for α helps the efficiency of the ladder computation. We ran a search program written in Magma that started with $\alpha = 1$ and incremented the value one by one. For each value of α, the Montgomery form curve $y^2 = x^3 + Ax^2 + x$ was generated with $A = 4\alpha - 2$ and the parameters $n, n_T, h, h_T, \ell, \ell_T$ and D mentioned in Section 2.2 were computed. The first α (and the corresponding A) for which an optimal value for (h, h_T) was obtained and the parameters ℓ, ℓ_T and D were large was considered. If the value of A was such that it is possible to obtain a birational equivalence to a (twisted) Edwards form curve, then the corresponding pair of Montgomery and Edwards form curves has been reported. Specific details of the application of this search methodology to the primes $p506-45$ and $p510-75$ are mentioned below.

Curves over $\mathbb{F}_{2^{506}-45}$*:* Let $p = 2^{506}-45$. We ran a search program to find Montgomery curves $M[p, A]$ satisfying the security criteria given in Section 2.2. The minimum positive value of A for which $(h, h_T) = (4, 4)$ and the other parameters mentioned in Section 2.2 are large is $A = 996558$. This gives the curve $M[p506-45, 996558]$. The curve $E[p506-45, 249140/249139]$ is birationally equivalent to $M[p506-45, 996558]$ using the birational equivalences given by (3) and (4). The quantity 249140/249139 is a non-square modulo $p506-45$ and so the addition formula over $E[p506-45, 249140/249139]$ is complete. The parameters for $M[p506-45, 996558]$ are given in Fig. 1 of the Appendix. The point $(3, \cdot)$ is a point of order ℓ on the Montgomery curve $M[p506-45, 996558]$; the corresponding point on the Edwards curve $E[p506-45, 249140/249139]$ is $(\cdot, 2)$. The set of scalars is defined to be $4(2^{503} + \{0, 1, \ldots, 2^{503} - 1\})$. Given a 64-byte scalar a, assuming the least significant byte ordering, the clamping function $clamp(a)$ is defined as follows: clear bits 0 and 1 of the first byte; set bit number 1 of the last byte and clear bits numbered 2 to 7 of the last byte.

Remark: Let $\alpha = (A+2)/4 = 249140$. The curves $M[p506-45, 4\alpha - 2]$ and $E[p506-45, 1-\alpha]$ can be shown to be 4-isogenous using the isogenies given in [13]. Further, using the fact that $-\alpha$ is a square in $\mathbb{F}_p$, the curves $M[p506-45, 2 - 4/\alpha]$ and $E[p506-45, 1-\alpha]$ are birationally equivalent using the birational equivalences given by (3) and (4).

Table 2 Saturated limb representations of primes related to this work

Prime	m	κ	η	ν	$64\kappa - m$
$p448$-224-1	448	7	64	64	0
$p506-45$	506	8	64	58	6
$p510-75$	510	8	64	62	2
$p521-1$	521	9	64	9	55

Curves over $\mathbb{F}_{2^{510}-75}$*:* Let $p = 2^{510}-75$. We ran a search program to find Montgomery curves $M[p, A]$ satisfying the security criteria given in Section 2.2. The minimum positive value of A for which an optimal value of (h, h_T) is obtained is $A = 793638$. In this case, neither $(A+2)$ nor $(A-2)$ is a square in $\mathbb{F}_p$. So, the birational equivalences in Section 2.1 for connecting Montgomery and Edwards curves cannot be applied. One may consider a quadratic twist of $E_{M,A,1}$. Since 2 is not a square, $E_{M,A,2}$ is a quadratic twist of $E_{M,A,1}$. Then $E_{M,A,2}$ can be connected to $E_{E,-1,d}$ using either of the birational equivalences given by (6), (7) or (8), (9). The form of d in these two cases are $(A-2)/(A+2)$ and $(A+2)/(A-2)$, respectively. Since both $(A+2)$ and $(A-2)$ are not squares, both $(A-2)/(A+2)$ and $(A+2)/(A-2)$ are squares. Consequently, the completeness of the addition formula over $E_{E,-1,d}$ is not ensured. Since $p \equiv 1 \bmod 4$, it is desirable to use birational equivalences to connect a Montgomery curve to a twisted Edwards form curve having a complete addition formula. For $A = 793638$, this does not seem to be possible using the birational equivalences in Section 2.1.

The next value of A for which an optimal value of (h, h_T) is obtained is $A = 952902$. In this case, we obtain the curves $M[p510-75, 952902]$ and $\widetilde{E}[p510-75, -238225/238226]$ which are birationally equivalent using the birational equivalences given by (6) and (7). The quantity $-238225/238226$ is a non-square modulo $p510-75$ and so the addition formula over $\widetilde{E}[p510-75, -238225/238226]$ is complete. The parameters for $M[p510-75, 952902]$ are given in Fig. 2 of the Appendix. The point $(4, \cdot)$ is of order ℓ on the Montgomery curve $M[p510-75, 952902]$; the corresponding point on the twisted Edwards curve $\widetilde{E}[p510-75, -238225/238226]$ is $(\cdot, 3/5)$. The set of scalars is set to be $8(2^{510} + \{0, 1, \ldots, 2^{510}-1\})$. Given a 64-byte scalar a, assuming the least significant byte ordering, the clamping function $clamp(a)$ is defined as follows: clear bits 0, 1 and 2 of the first byte; set bit number 5 of the last byte and clear bits numbered 6 and 7 of the last byte.

Remark: Let $\alpha = (A+2)/4 = 238226$, which is a square. The curves $M[p510-75, 4\alpha - 2]$ and $\widetilde{E}[p510-75, \alpha - 1]$ can be shown to be 4-isogenous using the isogenies given in [13]. Further, $M[p510-75, 4/\alpha - 2]$ and $\widetilde{E}[p510-75, \alpha - 1]$ are birationally equivalent using the birational equivalences given by (6) and (7). $M[p510-75, 2 - 4/\alpha]$ and $\widetilde{E}[p510-75, \alpha - 1]$ are birationally equivalent using the birational equivalences given by (8) and (9).

Curves over $\mathbb{F}_{2^{521}-1}$*:* The curve E-521 [11] is same as the curve $E[p521-1, -376014]$. Using the isogenies given in [13], the curve $E[p521-1, -376014]$ is 4-isogenous to $M[p521-1, 1504058]$. This gave us $M[p521-1, 1504058]$. Since the birational equivalences in Section 2.1 are simpler than the isogenies in [13], we obtained the Edwards form curve $E[p521-1, 376015/376014]$ which is birationally equivalent to $M[p521-1, 1504058]$. The birational equivalences are given by (3) and (4). The quantity $376015/376014$ is a non-square modulo $p521-1$ and so the addition formula over $E[p521-1, 376015/376014]$ is complete. The parameters for $M[p521-1, 1504058]$ are given in Fig. 3 of the Appendix. The point $(8, \cdot)$ is a point of order ℓ on the Montgomery curve $M[p521-1, 1504058]$; the corresponding point on the Edwards curve $E[p521-1, 376015/376014]$ is $(\cdot, 9/7)$.

The set of scalars for $E_{M,1504058,1}$ is set to be $4(2^{518} + \{0, 1, \ldots, 2^{518}-1\})$. Given a 65-byte scalar a, assuming the least significant byte ordering, the clamping function $clamp(a)$ is defined as follows: clear bits 0 and 1 of the first byte; set bit number 0 of the last byte and clear bits numbered 1 to 7 of the last byte.

Remark: Let $\alpha = (A+2)/4 = 376015$. The curves $M[p521-1, 2 - 4/\alpha]$ and $E[p521-1, 1-\alpha]$ are birationally equivalent using the birational equivalences given by (3) and (4).

4 Implementation

Let $m = 1 + \lfloor \log_2 p \rfloor$. Elements of $\mathbb{F}_p$ are m-bit strings which are represented using κ 64-bit words. Each such word is termed as a limb by convention. We have used packed or saturated limb representation of the field elements, according to which, m is written as $m = \eta(\kappa - 1) + \nu$ with $1 \le \nu \le \eta$, where $\eta = 64$. So, the first $\kappa - 1$ limbs of a field element are 64 bits long and the last limb has length between 1 and 64 bits.

The four primes that we have worked with are specified in Table 2 along with their representations. For the two primes $p506-45$ and $p521-1$, the value of $64\kappa - m \ge 3$ (which means, there are three or more 'free' bits in the last limb), for the prime $p510-75$, $64\kappa - m = 2$ (which means, there are two 'free' bits in the last limb) and for the prime $p448-224-1$, $64\kappa = m$ (which means, there are no 'free' bits in the last limb). There are consequences of these features to the Montgomery ladder computation which are mentioned below.

The Skylake and later processors provide the instruction triplet known as `mulx/adcx/adox`. These instructions allow the use of two independent carry chains for efficiently multiplying/squaring two large integers having 64-bit saturated limb representation. A general algorithmic description for multiplication/squaring of 64κ-bit numbers, $\kappa \ge 4$ can be found in [25]. We have used these algorithms for the implementation of integer multiplication/squaring.

Integer multiplication/squaring of κ-limb quantities produces a 2κ-limb output. The reduction step reduces this output modulo the prime p. For reducing an element after an integer multiplication/squaring, we have used algorithms **`reduceSLMP`** and **`reduceSLPMP`** from [25]; the algorithm **`reduceSLMP`** has been used for the Mersenne prime $p521-1$, while the algorithm **`reduceSLPMP`** has been used for the pseudo-Mersenne primes $p506-45$ and $p510-75$. A full reduction reduces the output to a value less than p. For intermediate steps of the computation, it is more efficient to perform a partial reduction. This is done using a size reduction which reduces the 2κ-limb quantity to a κ-limb quantity which is either an m-bit or an $(m+1)$-bit integer. For the Mersenne prime $p521-1$, size reduction to an m-bit integer is performed while for the pseudo-Mersenne primes $p506-45$ and $p510-75$ size reduction to an $(m+1)$-bit integer is performed.

The Montgomery ladder algorithm interleaves field multiplications/squarings with field additions/subtractions. The inputs to an addition/subtraction operation are outputs of multiplication/squaring operations and the outputs of addition/subtraction operations are inputs to multiplication/squaring operations. Consider the case of the pseudo-Mersenne primes $p506-45$ and $p510-75$. The outputs of multiplication/squaring are size reduced to $(m+1)$ bits and so the inputs to addition/subtraction operations are $(m+1)$-bit quantities. The outputs of the addition/subtraction operations are required to be κ-limb quantities so that the integer multiplication/squaring algorithm can be applied to these outputs. Depending upon the sizes of the outputs of addition/subtraction operations and the relative values of η and ν, it may be possible to avoid the reduction step after an addition/subtraction operation.

In the case of addition, since the inputs are at most $(m+1)$-bit quantities, the outputs of integer addition are at most $(m+2)$-bit quantities. If $m + 2 \le 64\kappa$, then the reduction after integer addition can be avoided. This condition holds for both $p506-45$ and $p510-75$ so that the reduction operation after integer addition can be avoided in both cases.

Consider a field subtraction of the type $a - b \bmod p$, where a and b are $(m+1)$-bit integers. To avoid handling negative numbers, a suitable multiple of p is added to a so that the result is guaranteed to be positive. Since the result will ultimately be reduced modulo p,

Table 3 CPU-cycle counts on Skylake and Kaby Lake processors for shared secret computation on the Montgomery form curves

Curve	Field	Security	Skylake	Kaby Lake	Ref.
Curve448	$\mathbb{F}_{2^{448}-2^{224}-1}$	223	536,362	521,934	[14]
$M[p506-45, 996558]$	$\mathbb{F}_{2^{506}-45}$	252	558,757	538,971	this work
$M[p510-75, 952902]$	$\mathbb{F}_{2^{510}-75}$	253.5	566,088	546,849	this work
$M[p521-1, 1504058]$	$\mathbb{F}_{2^{521}-1}$	259.5	689,588	666,044	this work

the correctness of the result is not affected by adding a multiple of p. In the case of $p506-45$, the operation $a - b \bmod p$ is performed as $(4p + a) - b$. The quantity $(4p + a) - b$ is at most an $(m + 3)$-bit quantity. Since for $p506-45$, $m + 3 < 64\kappa$, the value $(4p + a) - b$ fits in κ limbs and the reduction step is not required to be performed. On the other hand, for $p510-75$, the reduction step after subtraction has to be performed to ensure that the result fits within κ limbs. It is this feature that makes the ladder computation for $p506-45$ more efficient than the ladder computation for $p510-75$.

4.1 Timings

We have carried out the timing experiments on a single core of Skylake and Kaby Lake processors. The turbo-boost and hyper-threading features were turned off while measuring the cpu-cycles. The time stamp counter TSC was read from the CPU to RAX and RDX registers by RDTSC instruction.

Platform specifications: The specifications of the hardware and software tools used in our software implementations are given below.

Skylake: Intel® Core™ i7-6500U 2-core CPU @ 2.50 GHz. The OS was 64-bit Ubuntu 14.04 LTS and the source code was compiled using GCC version 7.3.0.

Kaby Lake: Intel® Core™ i7-7700U 4-core CPU @ 3.60 GHz. The OS was 64-bit Ubuntu 18.04 LTS and the source code was compiled using GCC version 7.3.0.

Timings in the form of cpu-cycles are provided in Table 3 for Skylake and Kaby Lake processors. For comparison, we have considered the timings of the most efficient previously published 64-bit implementation of Curve448 [14]. We downloaded the software for Curve448 and measured the cpu-cycles on the same platforms on which we have measured the cpu-cycles of our implementations. This has been done to keep the comparisons consistent.

The curves listed in Table 3 provide security at different security levels. Curves targeted at different security levels have possibly different applications. So, the main point of the results in Table 3 is to provide an understanding of the trade-off between security and efficiency. In particular, it provides an answer to the question of how much efficiency is lost in moving to higher security levels. Moving from Curve448 to either $M[p506-45, 996558]$ or $M[p510-75, 952902]$ increases security by about 30 bits, with slowdowns of 3–4% and 4.5–5%, respectively. Similarly, $M[p506-45, 996558]$ offers about 8 bits less security than $M[p521-1, 1504058]$ but, provides a speed-up of about 19%.

5 Conclusion

In this paper, we have proposed new Montgomery and Edwards form elliptic curves targeted at the 256-security level. Efficient 64-bit assembly implementations of DH shared secret computation on these curves have been made. Timings have been obtained on the Skylake and Kaby Lake processors of Intel. Compared to Curve448, two of the new curves provide 29 and 30.5 bits of additional security with slowdowns of 3–4% and 4.5–5%, respectively. Consequently, at the 256-bit security level, these two curves provide acceptable security/efficiency trade-off compared to Curve448 which provides security at the 224-bit security level.

6 Acknowledgment

The authors thank the reviewers for providing comments on an earlier version which have helped in improving the paper.

7 References

[1] Diffie, W., Hellman, M.: 'New directions in cryptography', *IEEE Trans. Inf. Theory*, 1976, **22**, (6), pp. 644–654

[2] Koblitz, N.: 'Elliptic curve cryptosystems', *Math. Comp.*, 1987, **48**, (177), pp. 203–209

[3] Miller, V.S.: 'Use of elliptic curves in cryptography'. Advances in Cryptology – CRYPTO'85, Santa Barbara, California, USA, 18–22 August 1985, pp. 417–426. Available at http://dx.doi.org/10.1007/3-540-39799-X_31

[4] Montgomery, P.L.: 'Speeding the pollard and elliptic curve methods of factorization', *Math. Comput.*, 1987, **48**, (177), pp. 243–264

[5] Bernstein, D.J.: 'Curve25519: new diffie-hellman speed records'. Public Key Cryptography – PKC 2006, 9th Int. Conf. on Theory and Practice of Public-Key Cryptography, New York, NY, USA, 24–26 April 2006 (LNCS, **3958**), pp. 207–228. Available at https://doi.org/10.1007/11745853_14

[6] 'TLS Protocol, Version 1.3, RFC 8446', 2018. Available at https://datatracker.ietf.org/doc/rfc8446/?include_text=1. Accessed on 16 September 2019

[7] Langley, A., Hamburg, M.: 'Elliptic curves for security', 2016. Accessed on 16 September 2019. Internet Research Task Force (IRTF), Request for Comments: 7748. Available at https://tools.ietf.org/html/rfc7748

[8] 'FIPS PUB 180–4: 'Secure hash standards', 2015. Available at https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

[9] Koblitz, N., Menezes, A.: 'Pairing-based cryptography at high security levels'. Cryptography and Coding, 10th IMA Int. Conf., Cirencester, UK, 19–21 December 2005 (LNCS, **3796**), pp. 13–36. Available at https://doi.org/10.1007/11586821_2

[10] Barbulescu, R., Duquesne, S.: 'Updating key size estimations for pairings', *J. Cryptol.*, 2019, **32**, (4), pp. 1298–1336. Available at https://doi.org/10.1007/s00145-018-9280-5

[11] Aranha, D.F., Barreto, P.S.L.M., Geovandro, C.C.F.P., *et al.*: 'A note on high-security general-purpose elliptic curves', IACR Cryptology ePrint Archive, 2013, **2013**, p. 647. Available at http://eprint.iacr.org/2013/647

[12] 'FIPS PUB 186–2: 'Digital signature standard', 2000. Available at https://csrc.nist.gov/csrc/media/publications/fips/186/2/archive/2000-01-27/documents/fips186-2.pdf

[13] Costello, C., Naehrig, M.: 'Isogenies between (twisted) Edwards and Montgomery curves', 2015. Accessed on 16 September 2019. Available at https://cryptosith.org/papers/isogenies_tEd2Mont.pdf

[14] Oliveira, T., Hernandez, J.L., Hisil, H., *et al.*: 'How to (pre-)compute a ladder – improving the performance of X25519 and X448'. Selected Areas in Cryptography – SAC 2017 – 24th Int. Conf., Ottawa, ON, Canada, 16–18 August 2017 (LNCS, **10719**), pp. 172–191. Available at https://doi.org/10.1007/978-3-319-72565-9_9

[15] Bernstein, D.J., Duif, N., Lange, T., *et al.*: 'High-speed high-security signatures', *J. Cryptographic Eng.*, 2012, **2**, (2), pp. 77–89. Available at https://doi.org/10.1007/s13389-012-0027-1

[16] Bernstein, D.J., Birkner, P., Joye, M., *et al.*: 'Twisted Edwards curves'. Progress in Cryptology – AFRICACRYPT 2008, First Int. Conf. on Cryptology in Africa, Casablanca, Morocco, 11–14 June 2008 (LNCS, **5023**), pp. 389–405. Available at https://doi.org/10.1007/978-3-540-68164-9_26

[17] Hisil, H., Wong, K.K., Carter, G., *et al.*: 'Twisted Edwards curves revisited'. Advances in Cryptology – ASIACRYPT 2008, 14th Int. Conf. on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, 7–11 December 2008 (LNCS, **5350**), pp. 326–343. Available at http://dx.doi.org/10.1007/978-3-540-89255-7_20

[18] Nath, K., Sarkar, P.: '"Nice" curves', 2019. Cryptology ePrint Archive, Report 2019/1259. Available at https://eprint.iacr.org/2019/1259. Cryptology ePrint Archive, Report 2019/1259

[19] Bernstein, D.J., Lange, T.: 'Montgomery curves and the montgomery ladder', in Bos, J.W., Lenstra, A.K. (Eds.): '*Topics in computational number theory inspired by Peter L. Montgomery*' (Cambridge University Press, 2017), pp. 82–115

[20] Costello, C., Smith, B.: 'Montgomery curves and their arithmetic - the case of large characteristic fields', *J. Cryptographic Eng.*, 2018, **8**, (3), pp. 227–240. Available at https://doi.org/10.1007/s13389-017-0157-6

[21] Edwards, H.M.: 'A normal form for elliptic curves', *Bull. Am. Math. Soc.*, 2007, **44**, pp. 393–422

[22] Bernstein, D.J., Lange, T.: 'Faster addition and doubling on elliptic curves'. Advances in Cryptology – ASIACRYPT, Kuching, Sarawak, Malaysia, 2007 (LNCS, **4833**), pp. 29–50

[23] Hamburg, M.: 'Ed448-goldilocks, a new elliptic curve', 2015. Cryptology ePrint Archive, 2015/625. Available at https://eprint.iacr.org/2015/625

[24] Bernstein, D.J., Lange, T.: 'Safecurves: choosing safe curves for elliptic-curve cryptography'. Available at http://safecurves.cr.yp.to/index.html, Accessed on 23 November 2019
[25] Nath, K., Sarkar, P.: 'Efficient Arithmetic in (pseudo-)Mersenne Prime Order Fields', 2018. Cryptology ePrint Archive, Report 2018/985. Available at https://eprint.iacr.org/2018/985

8 Appendix

8.1 Curve parameters

In this section, we provide the values of the parameters of the curves considered in this work. The parameters for the curve $M[p506-45, 996558]$ are shown in Fig. 1, the parameters for the curve $M[p510-75, 952902]$ are shown in Fig. 2 and the parameters for the curve $M[p521-1, 1504058]$ are shown in Fig. 3.

8.2 Primes p511-187 and p519-91

In this section, we provide reasons for not considering the primes $p511-187$ and $p519-91$.

Case of $p511-187$*:* We have considered the primes $p506-45$ and $p510-75$ for obtaining suitable Montgomery form curves. The prime $p511-187$ has been earlier proposed in [11]. We mention the reasons for not considering this prime.

In Section 4, it has been mentioned that in the ladder algorithm, multiplications/squarings are interleaved with additions/subtractions. Also, it has been pointed out that for $p506-45$, the reduction operations after both additions and subtractions can be avoided while for $p510-75$, the reduction operation after addition can be avoided, but the reduction operation after subtraction needs to be performed. This makes the ladder algorithm for $p506-45$ more efficient than the ladder algorithm for $p510-75$.

With the above in mind, let us consider the case for $p511-187$. For this prime, $m = 511$, $\kappa = 8$, $\eta = 64$ and $\nu = 63$, i.e. elements of $p511-187$ have 8-limb representations where the first seven limbs are 64 bits each and the last limb has 63 bits. As mentioned in Section 4, for pseudo-Mersenne primes, the reduction algorithm after multiplication/squaring reduces to an $(m + 1)$-bit quantity. So, for $p511-187$, the output of the reduction algorithm after multiplication/squaring will be a 512-bit quantity. The outputs of multiplications/squarings are provided as inputs to addition/subtraction operations. Since the inputs to the addition/subtraction operations are themselves 512-bit quantities, it is not possible to avoid the reduction operations after the addition/subtraction operations. So, a summary of the comparative situation for the three primes is the following.

Case $p506-45$*:* The reduction operations after both additions and subtractions can be avoided.
Case $p510-75$*:* The reduction operations after additions can be avoided, but the reduction operations after subtractions need to be performed.
Case $p511-187$*:* The reduction operations after both additions and subtractions have to be performed.

Consequently, in terms of speed, the ladder algorithm for $p506-45$ will be the fastest, followed by the ladder algorithm for $p510-75$ and then the ladder algorithm for $p511-187$. The timing results in table show that the ladder algorithm for $p506-45$ is about 1.3–1.4% faster than the ladder algorithm for $p510-75$ at the cost of decreasing security by about 1.5 bits. In terms of efficiency, one may expect the ladder algorithm for $p510-75$ to be also about 1–2% faster than the ladder algorithm for $p511-187$ at the cost of decreasing security by about 0.5 bits. On the balance, the half-bit security gain of $p511-187$ over $p510-75$ is perhaps not worth a drop in speed.

Remark: Considering a pseudo-Mersenne prime to be of the form $2^m - \delta$, the value of δ for the primes $p506-45$, $p510-75$ and $p511-187$ are 45, 75 and 187, respectively. For the 64-bit arithmetic that we have considered, this difference in the value of δ

$$
\begin{aligned}
n &= 2094969989053530796808441405969663457418650909467561465269306475581525\backslash\\
&\quad 5629699187591525063427353962358442288489890600575597198262455620557 28\backslash\\
&\quad 669755385685788,\\
\ell &= 5237424972633826992021103514924158643546627273668903663173266188953 81\backslash\\
&\quad 4074247968978812658568384905896105721224726501438992995656139053 93 21\backslash\\
&\quad 67438846421447,\\
\log_2 \ell &= 504,\\
h &= 4,\\
k &= (\ell - 1)/17,\\
n_T &= 2094969989053530796808441405969663457418650909467561465269306475581525\backslash\\
&\quad 6296987958387255222908231949627108464657926763152945998259231127458215\backslash\\
&\quad 6296145754252,\\
\ell_T &= 5237424972633826992021103514924158643546627273668903663173266188953814\backslash\\
&\quad 0742469895968138057270579874067771161644816907882364995648077818645539\backslash\\
&\quad 074036438563,\\
\log_2 \ell_T &= 504,\\
h_T &= 4,\\
k_T &= (\ell_T - 1),\\
D &= -4543123557506175626449202213951075823120804418252321163949549473 4779\backslash\\
&\quad 4195928828008585501771008503667515913450626856090280190174101869082725\backslash\\
&\quad 988805571050252,\\
\lceil \log_2(-D) \rceil &= 508.
\end{aligned}
$$

Fig. 1 *Parameters for the curve* $M[p506-45, 996558]$

$$
\begin{aligned}
n &= 335195198248564927489350624955146153186984145514809834443089036093044\backslash\\
&\quad 100751840662869613465178025950119140505107956858789953330856566350769\backslash\\
&\quad 9101693245950696,\\
\ell &= 418993997810706159361688281193932691483730181893512293053861295116305\backslash\\
&\quad 125939800828587016831472532437648925631384946073487441663570707938462\backslash\\
&\quad 387711655743837,\\
\log_2 \ell &= 507,\\
h &= 8,\\
k &= \ell - 1,\\
n_T &= 335195198248564927489350624955146153186984145514809834443089036093044\backslash\\
&\quad 100751836685970480249730319221265361087801367443752734363284030977727\backslash\\
&\quad 4115131257091204,\\
\ell_T &= 837987995621412318723376562387865382967460363787024586107722590232610\backslash\\
&\quad 251879591714926200624325798053163402719503418609381835908210077444318\backslash\\
&\quad 528782814272801,\\
\log_2 \ell_T &= 508,\\
h_T &= 4,\\
k_T &= \ell_T - 1,\\
D &= -325310369051208815436075592806876377044871206601624247034270696834 45\backslash\\
&\quad 974097335403637319053453446802029316328771464346619663200532150290780 1\backslash\\
&\quad 0832342319114820,\\
\lceil \log_2(-D) \rceil &= 510.
\end{aligned}
$$

Fig. 2 *Parameters for the curve $M[p510{-}75, 952902]$*

does not affect the speed of the multiplication algorithms. In particular, we note that it has been reported in [25] that the multiplication algorithm for $p511{-}187$ takes 118 cycles on Skylake. We have observed the same speed for the multiplication algorithms for $p506{-}45$ and $p510{-}75$. While the value of δ does not affect the speed of the multiplication algorithm for 64-bit arithmetic, it may be an issue for 32-bit or, 16-bit arithmetic.

Case of $p519{-}91$: We have considered the Mersenne prime $p521{-}1$. A reviewer has pointed out that $p519{-}91 = 2^{519} - 91$ is a pseudo-Mersenne prime which is close to $p521{-}1$ and suggested us to consider it for implementation.

Both $p519{-}91$ and $p521{-}1$ have 9 limbs using packed or saturated limb representation. So, the integer multiplication algorithms for both the primes will have the same efficiency. The difference, however, is in the efficiency of the reduction algorithm. The reduction algorithm for Mersenne primes is more efficient than the reduction algorithm for pseudo-Mersenne primes (i.e. for the same value of κ, algorithm **`reduceSLMP`** is more efficient than algorithm **`reduceSLPMP`** described in [25]). So, overall the field multiplication algorithm for $p521{-}1$ is faster than the field multiplication algorithm for $p519{-}91$. In [25], it has been reported that the field multiplication algorithm for $p521{-}1$ requires 128 cycles on Skylake. Following the suggestion of the reviewer, we have implemented the field multiplication algorithm for $p519{-}91$ and this requires 137 cycles. Since the multiplication algorithm for $p519{-}91$ itself is slower than the multiplication algorithm for $p521{-}1$, the ladder algorithm for $p519{-}91$ will also be slower than the ladder algorithm for $p521{-}1$. Due to this reason, we did not consider $p519{-}91$ for implementation.

$$
\begin{aligned}
n &= 6864797660130609714981900799081393217269435300143305409394463459185547\\
&\quad 3183397654701903506606654631398546774636260936570417277131794810169277\\
&\quad 1973685174680434092,\\
\ell &= 1716199415032652428745475199770348304317358825035826352348615864796387\\
&\quad 5795849413675475876651663657849636693659065234142604319282948702542317\\
&\quad 7993421293670108523,\\
\log_2 \ell &= 519,\\
h &= 4,\\
k &= \ell - 1,\\
n_T &= 6864797660130609714981900799081393217269435300143305409394463459185547\\
&\quad 3183397657402341612674668277711407817986522025145656966844204623118357\\
&\quad 3174371407549680212,\\
\ell_T &= 1716199415032652428745475199770348304317358825035826352348615864796387\\
&\quad 5795849414350585403168667069427851954496630506286414241711051155779587\\
&\quad 8293592851887420053,\\
\log_2 \ell_T &= 519,\\
h_T &= 4,\\
k_T &= \ell_T - 1,\\
D &= -2563609914934638872981081855263139865560965378352719883225156029512739\\
&\quad 3934984014981040276318357885224640000675728312900694622181289046423550\\
&\quad 69855506040176465004,\\
\lceil \log_2(-D) \rceil &= 523.
\end{aligned}
$$

Fig. 3 *Parameters for the curve* $M[p521-1, 1504058]$