Elliptic Curve Cryptography on Standard Curves Using the Edwards Addition Law

Vincent Verneuil[1,2]

[1] Inside Contactless,
41, parc du Golf, 13 856 Aix-en-Provence cedex 3, France
vverneuil@insidefr.com
[2] Institut de Mathématiques de Bordeaux,
351, cours de la Libération, 33 405 Talence cedex, France

Abstract. In this paper we address the issue of performing scalar multiplication over standard curves using the Edwards unified addition law. In particular we detail the required mapping for curves having cofactor 1 and discuss the efficiency of this method for scalar multiplication.

Keywords: Elliptic Curve Cryptography, Scalar Multiplication, Edwards Curves, Standard Curves, Side-Channel Countermeasures.

1 Introduction

Scalar multiplication on elliptic curves is the main operation of many cryptographic protocols using elliptic curves such as ECDSA, ECDH, EC ElGamal, etc. Indeed it is often the most time consuming operation of these protocols and many studies have focused on speeding-up this operation. Moreover it may manipulate secret keys and thus be threatened by the side-channel analysis [16, 17] when implemented on embedded devices such as smart cards.

Short Weierstrass equation elliptic curves $y^2 = x^3 + ax + b$ over a large characteristic prime field $\mathbb{F}_p$ require to distinguish points additions (i.e. $P_1 + P_2$, $P_1 \neq P_2$) from point doublings ($P_1 + P_1$) which may yield to side channel leakages if no particular countermeasure is applied. Homogeneous and Jacobian projective coordinates are generally employed to avoid the expensive inversions in $\mathbb{F}_p$ of the affine group law [6] but still have different formulas for points additions and doublings.

Counterfeiting side-channel analysis may be achieved through several ways. Firstly, regular algorithms may be used, these algorithms either hide sensitive conditional branching by adding dummy point operations [7] or have a naturally regular structure [13–15]. Another countermeasure consists in using a unified addition law when available, which performs point additions and doublings using the same sequence of field operations and thus cannot be distinguished from one another[3]. Finally, the atomicity principle may be used to protect any algorithm by

[3] Such formulas has been proposed by Brier and Joye in [4] for affine and homogeneous projective coordinates but are very expensive.

expressing formulas using sequences of a single field operation pattern. Its exact cost varies for each algorithm but is generally reasonable [5, 18, 10].

More recently Edwards presented in [8] a unified addition law for elliptic curves of equation $x^2+y^2 = c^2(1+x^2y^2)$. Then Bernstein and Lange generalized this law in [3] to the larger group of curves $x^2+y^2 = 1+dx^2y^2$ which covers about 1/4 of all the elliptic curves over non-binary finite fields. Finally, Bernstein et al. presented in [2] the twisted Edwards curves of equation $ax^2 + y^2 = 1 + dx^2y^2$ with the corresponding unified addition law. This equation covers even more curves than the previous one and has a comparable addition cost. Moreover when the unified addition law is not necessary, Edwards and twisted Edwards curves provide a very fast doubling formula. Those curves are thus of the utmost interest for scalar multiplication implementors.

On the other hand, the implementation of elliptic curve cryptography in industrial products has required the development of standards. There are currently two main standards: NIST FIPS 186-3 [9] and the German Brainpool standard [19]. Indeed, other standards such as ANSI X9.62 [1], ISO 15946 [12], IEEE P1363 [11] and SECG [20] mainly provide pointers to NIST curves.

Fast and secure implementations of the elliptic curve scalar multiplication for industrial products should thus be optimized for NIST and/or Brainpool curves. There are 5 NIST curves over $\mathbb{F}_p$ with parameters of sizes 192, 224, 256, 384, and 521 and 7 Brainpool curves over $\mathbb{F}_p$ with parameters of sizes 160, 192, 224, 256, 320, 384, and 512. All have cofactor 1.

We study in this paper the explicit mappings to perform scalar multiplications of points lying on standard curves over $\mathbb{F}_p$ using Edwards and twisted Edwards addition laws. These mappings are described in [3, 2] but i) we thought that descriptions were not explicit enough for implementors desiring to use them, and ii) we did not found in the literature if they effectively lead to fast implementations on standard curves.

2 Edwards Curves

In [3] Bernstein and Lange mention that every elliptic curve having a point of order 4 has some quadratic twist birationally equivalent to an Edwards curve. Unfortunately all NIST and Brainpool curves have cofactor 1, nonetheless we can consider these curves over an extension field $\mathbb{F}_{p^e}$ in which they have a point of order 4.

Assume that a point $P = (x_1, y_1)$ lie on an elliptic curve

$$E_1 : y^2 = x^3 + ax + b$$

with $p > 3$ and $4a^3 + 27b^2 \in \mathbb{F}_p^*$. Assume also that $E_1(\mathbb{F}_p)$ has prime cardinal $n > 2$. We aim at computing the scalar multiplication $Q = k \cdot P$, $k \in \mathbb{Z}$.

The roots of the polynomial $\psi_4^* = \psi_4/\psi_2 = x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - a^3 - 8b^2$ give the x-coordinates of the points of order 4 of E_1. Since this polynomial has no root over $\mathbb{F}_p$ we need to consider an extension field $\mathbb{F}_{p^e}$ to find such roots. The degree of ψ_4^* ensures that $e \leq 6$. More precisely if ψ_4^* is irreducible over $\mathbb{F}_p$ then $e = 6$. This is the case for all NIST and Brainpool curves, except NIST P-224 for which ψ_4^* splits in two polynomials of degree 3.

Consider an irreducible polynomial $A(t)$ of degree e and r a root of ψ_4^* in $\mathbb{F}_p[t]/A(t)$. Notice that ψ_4^* is a suitable choice for A in most cases, which gives also $r = \bar{t}$. Then compute s a square root of $r^3 + ar + b$. If $s \notin \mathbb{F}_{p^e}$ it is necessary to consider an extension field of degree $2e$ instead of e.

Fact 1 *The minimal extension field to find a point of order 4 (r, s) is $\mathbb{F}_{p^6}$ for all NIST and Brainpool curves over $\mathbb{F}_p$.*

Let $(r_2, 0)$ denote $2(r, s)$ from the elliptic curve points group law and having order 2. We shall translate the curve E_1 to ensure that the point $(0, 0)$ belongs to it. This is done by considering the isomorphic curve obtained with the map $(x, y) \mapsto (x - r_2, y)$. This gives

$$E_2 : y^2 = x^3 + \lambda_1 x^2 + \lambda_2 x$$

on which lie $(r_1, s_1) = (r - r_2, s)$ the point of order 4, and $(x_2, y_2) = (x_1 - r_2, y_1)$ the point corresponding to P.

Let compute $d = 1 - 4{r_1}^3/{s_1}^2$ and $\lambda_3 = r_1/(1 - d)$. Then consider E_3 the following quadratic twist of E_2 :

$$E_3 : \lambda_3 y^2 = x^3 + \lambda_1 x^2 + \lambda_2 x$$

If λ_3 is a square in $\mathbb{F}_{p^6}$ then E_3 is isomorphic to E_2 and thus to E_1. This is the case of all NIST and Brainpool curves. The base point on E_3 becomes $(x_3, y_3) = (x_2, y_2/\sqrt{\lambda_3})$. Notice that $\lambda_1 = 2r_1(1 + d)/(1 - d)$ and $\lambda_2 = r_1^2$.

Now we consider the elliptic curve obtained by the substitution $(x, y) \mapsto (x/r_1, y/r_1)$. Since $r_1 \neq 0$ – otherwise (r_1, s_1) would be of order 2 – E_3 is isomorphic to the so obtained elliptic curve :

$$E_4 : \frac{1}{1-d} y^2 = x^3 + \frac{2(1+d)}{1-d} x^2 + x$$

The base point thus becomes $(x_4, y_4) = (x_3/r_1, y_3/r_1)$.

Finally we consider the Edwards elliptic curve E_5:

$$E_5 : x^2 + y^2 = 1 + dx^2y^2$$

Bernstein and Lange have shown that E_4 – and therefore E_1 – is birationally equivalent to E_5. The point corresponding to P is given by $(x_5, y_5) = (2x_4/y_4, (x_4 - 1)/(x_4 + 1))$.

It is now possible to compute $(u_5, v_5) = k \cdot (x_5, y_5)$ using the Edwards addition formula in $\mathbb{F}_{p^6}$. Notice that d is a square for all NIST and Brainpool curves, thus the Edwards addition law is not *complete* – i.e. some special inputs may require particular handling.

The corresponding results on E_4,E_3 and E_2 are respectively obtained as $(u_4, v_4) = ((1+v_5)/(1-v5), 2(1+v_5)/u_5(1-v5))$, $(u_3, v_3) = (u_4 r_1, v_4 r_1)$, $(u_2, v_2) = (u_3, v_3\sqrt{\lambda_3}))$. Finally $Q = (u_2 + r_2, v_2)$.

This method allowed to successfully perform scalar multiplications on the 5 NIST curves and the 7 Brainpool curves defined over $\mathbb{F}_p$ using the Edwards addition law. However it implies handling coordinates in $\mathbb{F}_{p^6}$ during the scalar multiplication and thus increases operations cost by a factor at least 6 – which is very optimistic. On the other hand, the speed-up provided by the Edwards formula depends on the exact context but cannot reasonably exceed 50%. Consequently the cost of handling coordinates in $\mathbb{F}_{p^6}$ clearly overcomes the gain of using the Edwards addition law.

3 Twisted Edwards curves

In [2] Bernstein et al. mention that every elliptic curve having three points of order 2 is isogenous to a twisted Edwards curve. We thus experiment to find such points in an extension of $\mathbb{F}_p$ in order to perform the scalar multiplication on a twisted Edwards curve.

Let E_1, P and Q be defined as previously. Points of order 2 have by definition their y-coordinate to 0. We thus need to find r_0, r_1, r_2 the roots of the polynomial $x^3 + ax + b$.The three points of order 2 are then $(r_0, 0)$, $(r_1, 0)$ and $(r_2, 0)$.

Fact 2 *The minimal extension field to find three points of order 2 is $\mathbb{F}_{p^3}$ for all NIST and Brainpool curves over $\mathbb{F}_p$.*

As previously we shall translate the curve E_1 to ensure that the point of order 2 $(0, 0)$ belongs to it. Again, this is done by considering the isomorphic curve obtained by applying $(x, y) \mapsto (x - r_0, y)$. This yields

$$E_2 : y^2 = x^3 - (r_1 + r_2)x^2 + (r_1 r_2)x$$

on which lies $(x_2, y_2) = (x_1 - r_0, y_1)$ the point corresponding to P.

Then we consider the isogenous curve E_3 given by :

$$E_3 : y^2 = x^3 + 2(r_1 + r_2)x^2 + (r_1 - r_2)^2 x$$

on which lies $(x_3, y_3) = ({y_2}^2/{x_2}^2, y_2(r_1 r_2 - {x_2}^2)/{x_2}^2)$.

Then we consider the Montgomery curve E_4, isomorphic to E_3, given by :

$$E_4 : \frac{1}{r_1 - r_2} y^2 = x^3 + \frac{2(r_1 + r_2)}{r_1 - r_2} x^2 + x$$

on which lies $(x_4, y_4) = (x_3/(r_1 - r_2), y_3/(r_1 - r_2))$.

Bernstein et al. have shown that every Montgomery curve is birationally equivalent to a twisted Edwards curve, in particular E_4 is birationally equivalent to :

$$E_5 : 4r_1x^2 + y^2 = 1 + 4r_2x^2y^2$$

on which lies $(x_5, y_5) = (x_4/y_4, (x_4 - 1)/(x_4 + 1))$ the point corresponding to P.

The scalar multiplication on E_5 becomes $(u_5, v_5) = (k/2 \mod n'/4) \cdot (x_5, v_5)$ with $n' = |E_1(\mathbb{F}_{p^3})|$. From the Weil formula we have $n' = p^3 + 1 - t(t^2 - 3p)$ where $t = p + 1 - n$.

Remark 1. Notice that $k/2 \mod n'/4$ has the size of k if k is even but is 3 times longer if k is odd. However this is not a real issue since one may only consider even scalars or compute $(k-1) \cdot P$ on the twisted Edwards curve and add P on the Weierstrass curve.

Notice also that the twisted Edwards addition law is complete if r_1 is a square in $\mathbb{F}_{p^3}$ but not r_2. It means that among the three roots of $x^3 + ax + b$ at least one has to be a square and at least one a non-square. This property was always fulfilled except for NIST P-224.

The corresponding results on E_4,E_3 and E_2 can be respectively recovered as $(u_4, v_4) = ((1+v_5)/(1-v5), (1+v_5)/u_5(1-v5))$, $(u_3, v_3) = (u_4(r_1-r_2), v_4(r_1-r_2))$, $(u_2, v_2) = ({v_3}^2/4{u_3}^2, v_3((r_1 - r_2)^2 - {u_3}^2)/8{u_3}^2)$. Finally $Q = (u_2 + r_0, v_2)$.

We successfully performed scalar multiplications on the 12 NIST and Brainpool curves over $\mathbb{F}_p$ using the twisted Edwards addition law. However, it implies computations in $\mathbb{F}_{p^3}$, whose cost is lower than previously but still clearly overcomes the gain of using the twisted Edwards formula.

4 Conclusion

We pointed out in this article that standard elliptic curves over $\mathbb{F}_p$ require to perform scalar multiplication over an extension field of degree 6 to apply the Edwards addition law or degree 3 for the twisted Edwards addition law. However such an extension degree is not affordable with regard to the speed-up provided by Edwards and twisted Edwards formulas. This is especially the case in the embedded devices context which is leading in the industrial elliptic curves cryptography development due to its memory and power constraints.

In conclusion we guess that, as long as there will be no standard elliptic curves over $\mathbb{F}_p$ with cofactor 4, speeding-up the scalar multiplication of Weierstrass curves will remain a topic of interest in the industrial context.

References

1. ANSI X9.62–2005. *Public Key Cryptography for The Financial Service Industry : The Elliptic Curve Digital Signature Algorithm (ECDSA)*. American National Standards Institute, November 2005.
2. Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 389–405. Springer, 2008.
3. Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. Cryptology ePrint Archive, Report 2007/286, 2007. `http://eprint.iacr.org/`.
4. O. Billet and M. Joye. The Jacobi Model of an Elliptic Curve and Side-Channel Analysis. Cryptology ePrint Archive, Report 2002/125, 2002.
5. Benoît Chevallier-Mames, Mathieu Ciet, and Marc Joye. Low-cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity. *IEEE Transactions on Computers*, 53(6):760–768, 2004.
6. Henri Cohen, Gerhard Frey, et al. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman & Hall/CRC, 2005.
7. J.-S. Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES '99*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 1999.
8. Harold M. Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44:393–422, 2007.
9. FIPS PUB 186-3. *Digital Signature Standard*. National Institute of Standards and Technology, March 2006. Draft.
10. C. Giraud and V. Verneuil. Atomicity Improvement for Elliptic Curve Scalar Multiplication. In Julien Iguchi-Cartigny, Dieter Gollmann, and Jean-Louis Lanet, editors, *CARDIS 2010*, volume 6035 of *Lecture Notes in Computer Science*, pages 80–101. Springer, 2010.
11. IEEE Std 1363a-2004. *IEEE Standard Specifications for Public-Key Cryptography – Amendment 1: Additional Techniques*. IEEE Computer Society, September 2004.
12. ISO/IEC 15946-1. *Information Technology – Security Techniques – Cryptographic techniques Based on Elliptic Curves – Part 1: General*, 2002.
13. M. Joye and S.-M. Yen. The Montgomery Powering Ladder. In B.S. Kaliski Jr., Ç.K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 291–302. Springer, 2002.
14. Marc Joye. Highly regular right-to-left algorithms for scalar multiplication. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 135–147. Springer, 2007.
15. Marc Joye. Highly regular m-ary powering ladders. In *16th Annual Workshop on Selected Areas in Cryptography (SAC 2009)*, August 2009.
16. P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.
17. P. Kocher, J. Jaffe, and B. Jun. Introduction to Differential Power Analysis and Related Attacks. Technical report, Cryptography Research Inc., 1998.
18. Patrick Longa. Accelerating the Scalar Multiplication on Elliptic Curve Cryptosystems over Prime Fields. Master's thesis, School of Information Technology and Engineering, University of Ottawa, Canada, 2007.
19. M. Lochter and J. Merkle. *Elliptic Curve Cryptography Brainpool Standard Curves and Curve Generation*. ECC Brainpool, March 2010. RFC 5639 `http://tools.ietf.org/html/rfc5639`.
20. Standards for Efficient Cryptography Group (SECG). *SEC 2 : Recommended Elliptic Curve Domain Parameters*. Certicom Research, 2000.

A Numerical Example

We provide in the following two examples of scalar multiplication on the NIST P-192 curve using Edwards and twisted Edwards formula with the following base point and scalar :

$P = (6020462823756886567582134805875261119166989766368846848 18, 1740503322936220314048575522802194103640234889273866506 41)$

$k = 764793464287780488015619147195181618135290770882198355914$

A.1 Edwards Mapping

$A(t) = t^6+6277101735386680763835789423207666416083908700390324961264t^4+5163398772472111007955535234149537078279480026049155003234 7t^3+6277101735386680763835789423207666416083908700390324961234t^2+4353459610560602757540369289538755752893661896372995011664t+5565725628672756385620692447509300801594978908324954804287$

$(r, s) = (t, 1232885337329876721108650781123026855340115696851748218716t^5+4932477686389072668016991071412418223708629683233024004 48t^4+3738804120027750459147372311576249545169533275370852887435t^3+2985549502162143631678647934517474989370195960933744006916t^2+345127056729561339752586048651174548525442418081292243533 5t+3675696037813148084070050989945142303640407213286917992 43)$

$d = 3420338982522812908064423807958273930294360305398794368312t^5+7096293228683146697806937966593386891854953651283738782 41t^4+6276253110032103724812950305489023427888477883448250281073t^3+8826093320057595705768052216495766672348376319910524707 06t^2+3744415841762234276757594919031605484022217101057518417 395t+2544405952845063473758844807435704955368111104330496027457$

$(x_3, y_3) = (6042801227311487248891532886824760140143018265622682125527t^5+4999440895681090877477447164012672236561385433529378727 18t^4+3280207113052709209219595093606878631724660867469997005 28t^3+4443428579974747537705574162279619493381428119369885465088t^2+2669751631708896080503479091367749654876457936583305717 70t+4908438192344199888037793697631965658617568918402742032 92, 5448940144146238690395392987575249102457906876388411728494t^5+6253498282324623590425802397073442417403878977744368969 42t^4+5371398455506405768885608633602548599109111181777341154649t^3+1062401218587368023903413873640100662769657989816371162819t^2+359220712268489341702386000795425509993716051415878885572 6t+5238832436750337898023115609617977527979118313938908189912)$

$(x_5, y_5) = (2411620920747559806654530833406361680285397433517820138225t^5+2178194511121855954145423061971735686308488106140904616061 t^4+3035145219164513855423694820214289866360537862491721611532t^3+1575841831888966463076829213240133390198510497312856768131t^2+3433775095752019020811384990062716869320098942070497063477t+6055993466415172059541564274034851443862312849531737220508, 9344763454052934094154325193271636227285836422065916841 20t^5+4901588814074284966087421037449470933422314692300849183648t^4+1560145556924236026462482284446476973220445880317716383270t^3+5934114755194262414936404990219501543462599522373770705893t^2+3433266300379969589597679547146430322784102918299141574256t+3714568610352283246178132491546257131387479417673263847505)$

$(u_5, v_5) = (3338563021775073422109629931257855172669205872938547929456t^5+1806851729074167634573774551481175278099574410842660666947t^4+4673054178789749506142787495338016851836251851521115922164t^3+1270682325639584583001564867621533011954970211690961310882t^2+5096188360313330717937411142038755611374198189517746443465t+4793145063747189037226039589343742100627088263323126198918, 1815426887389087292654708598859930356510180535301706364309t^5+3857849846656926882814158216759927508517260009708024804663t^4+4318015937391865862699834399829172507629462535198720840343t^3+5074827600207293986247307542569191531430258921898515155881t^2+5479129751516854254694027158300638119981336424525611064498t+1024342340679804401361714485156265354582960566628481184047)$

$(u_3, v_3) = (6042801227311487248891532886824760140143018265622682125527t^5+4999440895681090877477447164012672236561385433529378727 18t^4+3280207113052709209219595093606878631724660867469997005 28t^3+4443428579974747537705574162279619493381428119369885465088t^2+2669751631708896080503479091367749654876457936583305717 70t+5870565252401868281583577845557231684877110298699$

$7855839828, 503857869404974462517173507902893313084295202651243908284 8t^5 + 483228421729943134577849006818371435796536536841361272353 8t^4 + 4954764612336757239071117165647861458182357846061702816045t^3 + 3387813826442384618785067154819157815604909640479908411139t^2 + 4817448041364297834094169058023603474411555045759680740322t + 1760621086073593505120109802912135356157779095265669943544)$

$Q = (59817677155431369495380119563815612309320523834944663213541, 196946982289323242514886925533049376351255301199522531313 2)$

A.2 Twisted Edwards Mapping

$A(t) = t^3 + 2t + 1$

$r_0 = 6083103646888544317907094650818583011910921741494542256775 * t^2 + 4368451759798174404689572507993008788609157767765439032756 * t + 6018437617389165502597529726688888543853259421862614688607$

$r_1 = 4556865828697991142821498075117963306743488047993932093881 * t^2 + 1475183657913790228436926256431399077233929230584953774891 * t + 6075821104930654857095330766823951075657984063991909458508$

$r_2 = 1914233995186826066942986120478786513513407611292175571902 * t^2 + 1761131609797127336302524289571517719751358009566390551034 * t + 4599447484535411679787183529024932126565739149261257754 43$

$(x_3, y_3) = (1567774279891538794798584345970030882592710512965390685102t^2 + 624122645127030662417816848448246703918296188949972969638 2t + 3374287364290777484977431734834514106327229981410380498133 3, 3083704563052573594919968203614319701328991609763030251814t^2 + 933408426399387677114666626873122855519733694475392049152t + 4164184899005749115238481472342823957046889309915957989981)$

$(x_5, y_5) = (3326245108340266325698836274751743264377101002193210606469t^2 + 637178476784103578925455693539414920658734859435971111 57t + 5070922515795249246091321891841102483007737081343139104372, 8464181090635450804158791450132023054520679658253007990 68t^2 + 3212721855082997270360654922462307652073904138144768963955t + 2786444780743839629767133573681828728274372065369967297885)$

$n' = 2473304014731045340605025210196471900329508329283250192422968786506440280658405139290623424011948478837243387503855651089037364489596106925954602401360011848938243128633416 04$

$(u_5, v_5) = (5896485911085572588606767497646272327650234764836469701847t^2 + 178367726290130774410094024365932736530174220227985137421 7t + 1047827148725578075762879365083111564976027389163719612 26, 2177653810163441161082898805387568929778174452684400962065t^2 + 6548948159443124633451524340229669838804187223826200899 55t + 1427279821477246913068955360691570982228043335474060350276)$

$(u_3, v_3) = (3938259145591126335663580317058290056173278666311176620151t^2 + 571207562273103144353980929107696156132006851222677284896 1t + 4613071652742863411212401237424772137186554393429070450296, 5143088285531498596637539153296787310688945204925076743598t^2 + 3946731504147846738326029760878368117864666551965339208441t + 5732476917584366573365064797151837565843479127024684724703)$

$Q = (59817677155431369495380119563815612309320523834944663213541, 196946982289323242514886925533049376351255301199522531313 2)$