Paper

Faster Point Scalar Multiplication on Short Weierstrass Elliptic Curves over F_p using Twisted Hessian Curves over F_{p^2}

Michał Wroński

Military University of Technology, Warsaw, Poland

Abstract—This article shows how to use fast F_{p^2} arithmetic and twisted Hessian curves to obtain faster point scalar multiplication on elliptic curve E_{SW} in short Weierstrass form over F_p. It is assumed that p and $\#E_{SW}(F_p)$ are different large primes, $\#E(F_q)$ denotes number of points on curve E over field F_q and $\#E^t_{SW}(F_p)$, where E^t is twist of E, is divisible by 3. For example this method is suitable for two NIST curves over F_p: NIST P-224 and NIST P-256. The presented solution may be much faster than classic approach. Presented solution should also be resistant for side channel attacks and information about Y coordinate should not be lost (using for example Brier-Joye ladder such information may be lost). If coefficient A in equation of curve $E_{SW} : y^2 = x^3 + Ax + B$ in short Weierstrass curve is not of special form, presented solution is up to 30% faster than classic approach. If $A = -3$, proposed method may be up to 24% faster.

Keywords—elliptic curve cryptography, hardware implementations, twisted Hessian curves.

1. Introduction

The point scalar multiplication is used in many cryptographic applications, which are based on elliptic curve discrete logarithm problem (ECDLP). In this article a faster arithmetic on elliptic curves in short Weierstrass form E_{SW} over F_p is considered, where p is large prime and $\#E_{SW}$ is also prime. If twist of such a curve E^t_{SW} has its order $\#E^t_{SW}$ divisible by 3, then twisted Hessian curves arithmetic over F_{p^2} may be used to speed up point scalar multiplication on $E_{SW}(F_p)$. It is possible because $\#E_{SW}(F_{p^2}) = \#E_{SW}(F_p) \cdot \#E^t_{SW}(F_p)$. If $3|\#E^t_{SW}$ then $3|\#E_{SW}(F_{p^2})$. Hence, it is possible to find twisted Hessian curve $E_{TH}(F_{p^2})$ isomorphic to $E_{SW}(F_{p^2})$. If it is needed to make point scalar multiplication by $k \in \{2, \ldots, \#E_{SW}(F_p) - 2\}$ of point $P \in E_{SW}(F_p)$, to get in result point $Q \in E_{SW}(F_p)$ where $Q = [k]P$, it is not necessary to use short Weierstrass curve arithmetic. If ϕ is isomorphism from $E_{SW}(F_{p^2})$ to $E_{TH}(F_{p^2})$, so: $\phi : E_{SW}(F_{p^2}) \rightarrow E_{TH}(F_{p^2})$ then for every point $P \in E_{SW}(F_p)$ (then of course also $P \in E_{SW}(F_{p^2})$) may be found $P' \in E_{TH}(F_{p^2})$ for which $P' = \phi(P)$. To compute Q may be used formula $Q = \phi^{-1}\left([k]\phi(P)\right)$. One can see that $[k]\phi(P) = [k]P' = Q'$ and finally $\phi^{-1}(Q') = Q$. In hardware implementation F_{p^2} arithmetic, if is properly implemented, may be almost as fast as F_p arithmetic. Because twisted Hessian curves arithmetic for F_{p^2} fields (where $p \neq 2,3$) is complete (point addition, doubling, addition of neutral and addition of opposite point are computed using the same formulas), it is possible to gain faster solution, resistant for side channel attacks in hardware implementations (especially in FPGA chips). Due to the fact that any information about value of Y coordinate should not be lost, Brier-Joye ladder for point scalar multiplication [1] is not considered in this article.

2. Arithmetic in F_{p^2}

The field F_{p^2} is generated by irreducible polynomial of degree 2 with coefficients from F_p. The main goal of this article is to get fast arithmetic in F_{p^2}. Only an irreducible polynomials of form $f(t) = t^2 \pm c$ are considered, where c is small positive integer.

Every element $A \in F_{p^2}$ may be written as $A = a_1t + a_0$, where $a_0, a_1 \in F_p$.

Let's assume $A, B \in F_{p^2}$, where $A = a_1t + a_0$ and $B = b_1t + b_0$. Then $A \pm B = (a_1t + a_0) \pm (b_1t + b_0) = (a_1 \pm b_1)t + (a_0 \pm b_0)$. Addition and subtraction are not complex operations and may be computed in only one processor machine cycle. Although fast F_{p^2} arithmetic is presented in [2] to speed-up pairing, in this article is showed its different application. It is also showed how to fast compute inversion of element in F_{p^2}, based on idea presented in [3].

2.1. Multiplication

Multiplication is crucial operation in elliptic curve arithmetic. However, it is not the most time-consuming operation (it is inversion), during point scalar multiplication many times it is needed to compute multiplication in field over which elliptic curve is defined. Inversion is computed only once, at the end of all computations.

Let $A, B \in F_{p^2}$, where $A = a_1t + a_0$ and $B = b_1t + b_0$. Let $f(t) = t^2 \pm c$. Then using Karatsuba algorithm, element C is computed as:

$$C = A \cdot B = (a_1b_0 + a_0b_1)t + a_0b_0 \mp ca_1b_1 = Rt \mp Mc + N,$$

where:
$L = (a_1 + a_0)(b_1 + b_0)$,
$M = a_1 b_1$,
$N = a_0 b_0$,
$R = L - M - N = a_1 b_0 + a_0 b_1$.

One can notice that:

$$c_1 = R \quad \text{and} \quad c_2 = \mp Mc + N .$$

Multiplication in F_{p^2} requires 3 multiplications in F_p, 5 additions/subtractions in F_p and 1 multiplication in F_p by small constant.

Although in software applications the multiplication in F_{p^2} is still more complex than multiplication in F_p, using parallelism in hardware it is possible to compute it in almost the same time.

The total number of processor cycles required to make multiplication in F_{p^2} is $MAX\{T_M + 2, T_M + \lceil \log_2 c \rceil + 1\}$. One can see that the smaller c is chosen, the less operations are required to compute the result.

In the case when $p \equiv 3 (mod\ 4)$, an irreducible polynomial of form $f(t) = t^2 + 1$ may be chosen and then the cost of multiplication equals $MAX\{T_M + 2, T_M + \lfloor \log_2 1 \rfloor + 1\} = T_M + 2$ processor cycles. In the case $p \equiv 5 (mod\ 8)$, the irreducible polynomial of form $f(t) = t^2 - 2$ may be chosen and then the complexity of multiplication reach $MAX\{T_M + 2, T_M + \lceil \log_2 2 \rceil + 1\} = T_M + 2$ processor cycles.

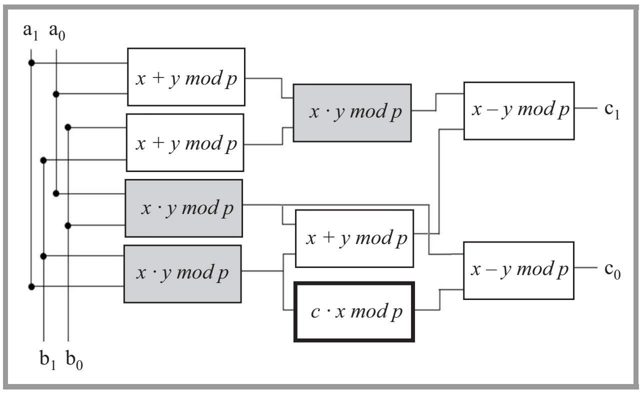


Fig. 1. Scheme of parallel multiplication in F_{p^2}.

2.2. Inversion

It is possible for every $A \in F_{p^2}$ to get its inversion A^{-1} by computing only one inversion of element from F_p. Hence, the method for irreducible polynomial of form $f(t) = t^2 \pm c$ is shown base on idea presented in [3]:

$$A = \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} \quad \text{and} \quad A^{-1} = \begin{bmatrix} b_1 \\ b_0 \end{bmatrix} .$$

If

$$M = \begin{bmatrix} a_0 & a_1 \\ \mp a_1 c & a_0 \end{bmatrix} ,$$

then

$$M \cdot \begin{bmatrix} b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 \\ \mp a_1 c & a_0 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} .$$

The coefficients in matrix M may be taken from general form of element $C = A \cdot B$.

Then transformation should be made:

$$\begin{bmatrix} b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 \\ \mp a_1 c & a_0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = M^{-1} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} .$$

Now the determinant of matrix M is equal to:

$$\det(M) = a_0^2 \pm a_1^2 c ,$$

then

$$M^{-1} = \frac{1}{\det(M)} \begin{bmatrix} a_0 & -a_1 \\ \pm a_1 c & a_0 \end{bmatrix}$$

and

$$\begin{bmatrix} b_1 \\ b_0 \end{bmatrix} = \frac{1}{\det(M)} \begin{bmatrix} a_0 & -a_1 \\ \pm a_1 c & a_0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\det(M)} \begin{bmatrix} -a_1 \\ a_0 \end{bmatrix} .$$

The computations may be done in the following 6 steps:

1. $D = a_0^2$,
2. $E = a_1^2 c$,
3. $H = E + D = \det(M)$,
4. $\overline{H} = H^{-1}$,
5. $b_1 = -a_1 \overline{H}$,
6. $b_0 = a_0 \overline{H}$.

The inversion in F_{p^2} requires 1 inversion in F_p, 4 multiplications in F_p, 1 multiplication by small constant c in F_p and 1 addition in F_p.

3. Elliptic Curves

An elliptic curve may be defined over every field K. Because in cryptographic applications only finite fields are used and in this article only fields with big characteristic $p \neq 2,3$ are considered, then all definitions below are proper for such fields but may not be proper for fields with characteristic 2 or 3.

3.1. Short Weierstrass Elliptic Curve

Every elliptic curve E over F_q with $char(F_q) \neq 2,3$ may be given in short Weierstrass form $E_{SW} : y^2 = x^3 + Ax + B$, where $-16\left(4A^3 - 27B^2\right) \neq 0$. The arithmetic on short Weierstrass curve is in general not complete (there is such a method described in [4] but it is not efficient) so different formulas for point addition, doubling, addition of neutral element and addition of opposite element are used. The short Weierstrass curve arithmetic is the fastest for

$A = -3$. In this case, points addition requires 14 multiplications and 7 additions in F_q. Mixed addition requires 11 multiplications and 7 additions in F_q. Point doubling requires 10 multiplications and 11 additions in F_q. If A is not of special form, then points addition requires 14 multiplications and 7 additions in F_q. Mixed addition requires 11 multiplications and 7 additions in F_q. Point doubling requires 12 multiplications and 12 additions in F_q. It is assumed that squaring, multiplication by vary elements and multiplication by big constant require the same time. All necessary formulas may be found in [5].

3.2. Twisted Hessian Curves

The twisted Hessian curve [6] over field F_q is given by:

$$E_{TH} : ax^3 + y^3 + 1 = dxy$$

with neutral point $(0, -1)$ in affine coordinates or by:

$$E_{TH} : aX^3 + Y^3 + Z^3 = dXYZ$$

in projective coordinates with neutral point $(0, -1, 1)$. Elements $a, d \in F_q$ and $a(27a - d^3) \neq 0$.

If $a = 1$ then $E_{TH,a,d} : x^3 + y^3 + 1 = dxy$ is Hessian curve. On twisted Hessian curves faster arithmetic than for short Weierstrass curves may be used. Moreover, on twisted Hessian curve over F_q, if $q \equiv 1(mod\ 3)$ and a is not cube in F_q, complete arithmetic may be used.

Arithmetic on twisted Hessian curves is described with all details in [6].

The best complete addition formula requires 12 multiplications, i.e. 11 multiplications of vary elements and 1 multiplication by constant and 16 additions/subtractions.

3.3. Isomorphism between Twisted Hessian Curves and Elliptic Curves in Short Weierstrass Form over Finite Fields

Let us consider computation of $Q = [k]P$ on elliptic curve $E_{SW}(F_q) : y^2 = x^3 + Ax + B$, where $Ord(P)$ is prime. If $3|\#E_{SW}(F_q)$ (it means that curve E_{SW} has 3-torsion point) and $q \equiv 1(mod\ 3)$ then for curve E_{SW} may be found isomorphic twisted Hessian curve E_{TH} with complete arithmetic (when a is not cube in F_q). Note there is not any elliptic curve over F_p having isomorphic twisted Hessian curve over F_p if $\#E_{SW}(F_p)$ is prime.

One can see that for some elliptic curves over F_p for which $\#E_{SW}(F_p)$ is prime, there is some possibility that $\#E^t_{SW}$ is divisible by 3.

Let us see, that if $\#E_{SW}(F_p) = p + 1 - t$ over F_p then $\#E^t_{SW} = p + 1 + t$ and $\#E_{SW}(F_{p^2}) = \#E_{SW}(F_p) \cdot \#E^t_{SW}(F_p) = (p+1)^2 - t^2$ over F_{p^2}. So if $3|\#E^t_{SW}$ then $3|\#E_{SW}(F_{p^2})$. Because $p^2 \equiv 1(mod\ 3)$ for all primes $p \neq 3$, therefore a twisted Hessian curve such as $E_{TH}(F_{p^2}) : ax^3 + y^3 + 1 = dxy$ isomorphic to $E_{SW}(F_{p^2})$ exists and if a is not cube in F_{p^2}, then complete arithmetic on $E_{TH}(F_{p^2})$ may be used. Finally, point $Q \in E_{SW}$ may be computed using twisted Hessian curve over F_{p^2} instead of using short Weierstrass curve over F_p.

This rule was checked for NIST elliptic curves over F_p. For two curves, NIST P-224 and NIST P-256, may be used arithmetic on twisted Hessian curve over F_{p^2} isomorphic to $E_{SW}(F_{p^2})$.

The next important problem is how to find such twisted Hessian curve.

First, suppose that triangular elliptic curve is given by:

$$E_{TR} : \overline{y}^2 = d\overline{xy} + a\overline{y} = \overline{x}^3 \quad \text{over} \quad F_{p^2},$$

where $a, d \in F_p$.

Then the transformations can be made:

$$\left(\overline{y} + \tfrac{dx+a}{2}\right)^2 = \left(\overline{x} + \tfrac{d^2}{12}\right)^3 + \\ + \left(\tfrac{da}{2} - \tfrac{d^4}{48}\right)\left(\overline{x} + \tfrac{d^2}{12}\right) - \tfrac{d^2}{12}\left(\tfrac{da}{2} - \tfrac{d^4}{48}\right) + a^2 .$$

If

$$E_{SW} : y^2 = x^3 + Ax + B$$

then:

$$x = \overline{x} + \frac{d^2}{12},$$

$$y = \overline{y} + \frac{dx + a}{2},$$

$$A = \frac{da}{2} - \frac{d^4}{48},$$

$$B = -\frac{d^2}{12}A + a^2 .$$

For elliptic curves over F_p it is possible to extend field from F_p to F_{p^2}. Then the coefficients of such a curve over field extension still belong to F_p. If coefficients $A, B \in F_p$ of such a curve are known, to find coefficients of twisted Hessian curve $a, d \in F_{p^2}$ it is necessary to compute d as one of the roots of polynomial $J(s) = \frac{-1}{6912}s^8 - \frac{1}{24}As^4 - Bs^2 + A^2$. Note that if d is computed, then $a = (A + \frac{d^4}{48})\frac{2}{d}$, and in projective coordinates $E_{TR} : VW\,(V + dU + aW) = U^3$.

Then for triangular curve E_{TR} it is easy to find isomorphic twisted Hessian curve given by equation:

$$E_{TH,\left(d^3-27a\right),3d} : (d^3 - 27a)X^3 + Y^3 + Z^3 = 3dXYZ$$

and

$$X = U ,$$

$$Y = \omega(V + dU + aW) - \omega^2 V - aW ,$$

$$Z = \omega^2\big(V + dU + aW\big) - \omega V - aW ,$$

where ω is not trivial cubic root from 1 and $X, Y, Z, \omega \in F_{p^2}$.

Now the complete arithmetic (because presented solution must resistant for side channel attacks) may be used to compute $Q' \in E_{TH,(d^3-27a),3d}$ by $Q' = [k]P'$, where $P' = \phi(P)$. $\phi : E_{SW} \rightarrow E_{TH,(d^3-27a),3d}$ is isomorphism from E_{SW} to $E_{TH,(d^3-27a),3d}$. When Q' is known, it is necessary to find Q. It may be computed using $\phi^{-1} : E_{TH,(d^3-27a),3d} \rightarrow E_{SW}$, because $\phi^{-1}(Q') = Q$. However, $Q' \in E_{TH,(d^3-27a),3d}(F_{p^2})$

and $Q' \notin E_{TH,(d^3-27a),3d}(F_p)$, but $Q \in E_{SW}(F_{p^2})$ and $Q \in E_{SW}(F_p)$. Note that to find Q having $Q' = (X_Q, Y_Q, Z_Q)$, some more transformations are necessary.
Firstly, there a point on triangular curve

$$Q'' = (U_Q, V_Q, W_Q),$$

must be found, where:

$$U_Q = X_Q\,,$$

$$V_Q = -\frac{dX_Q + \omega Y_Q + \omega^2 Z_Q}{3}\,,$$

$$W_Q = -\frac{dX_Q + Y_Q + Z_Q}{3a}\,.$$

Finally from the formulas

$$x_Q = \frac{U_Q}{W_Q} + \frac{d^2}{12}\,,$$

$$y_Q = \frac{V_Q}{W_Q} + \frac{d\frac{U_Q}{W_Q} + a}{2}$$

the result $Q = (x_Q, y_Q) = [k]P$ can be found.

4. Speed-up for NIST Curves

Using presented ideas it is possible to speed-up point scalar multiplication on two NIST curves over F_p: NIST P-224 and NIST P-256. For both of these curves isomorphic twisted Hessian curves E_{TH} over F_{p^2} have coefficient a which is not cube in F_{p^2}, so it is impossible to use Hessian curves arithmetic [7], [8]. For others NIST curves over large prime fields the smallest field extensions, for which isomorphic twisted Hessian curves exist are:

- 8 for NIST P-192 and NIST P-384,
- 4 for NIST P-521.

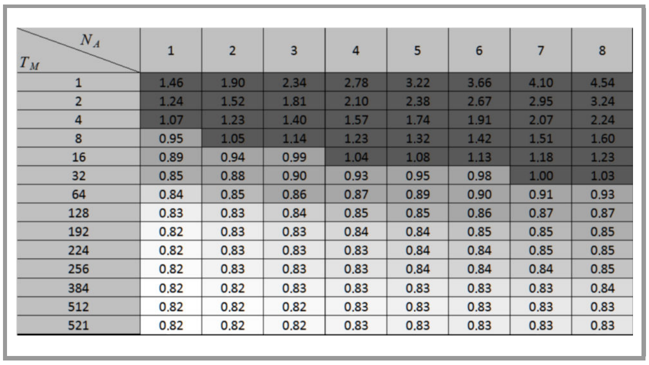
One can see that the bigger the degree of field extension is, the more resources are required to implement F_{p^n} arithmetic in hardware. Hence, the most suitable are ellipitc curves for which F_{p^2} arithmetic may be used.
For NIST P-224 it is possible to find twisted Hessian curve over F_{p^2} which is isomorphic to NIST P-224 over F_{p^2}.
The irreducible polynomial of form $f(t) = t^2 + 11$ for arithmetic in F_{p^2} may be used in this case. Multiplication using such a polynomial requires then $T_M + \lceil \log_2 11 \rceil + 1 = T_M + 5$ processor cycles, where T_M is number of processor cycles required for multiplication in F_p.
For NIST P-224 it is possible to find twisted Hessian curve over F_{p^2} which is isomorphic to NIST P-256 over F_{p^2}.
The irreducible polynomial of form $f(t) = t^2 + 1$ for arithmetic in F_{p^2} may be used in this case. Multiplication using such a polynomial requires then $T_M + 2$ processor cycles, where T_M is number of processor cycles required for multiplication in F_p.

5. Comparison with Other Methods of Point Scalar Multiplication

Arithmetic on twisted Hessian curves may be very interesting, because:

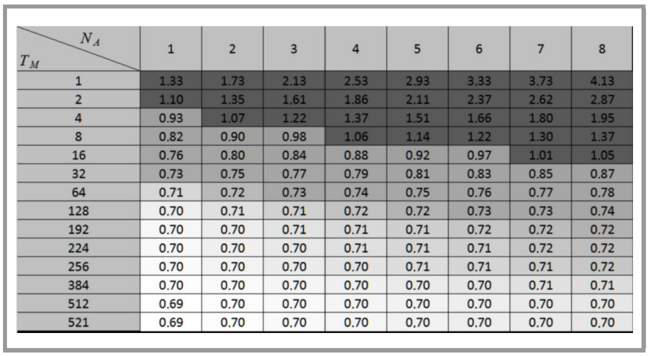
- it is faster method than classic arithmetic on NIST curves in short Weierstrass form over F_p in hardware,
- it allows to use complete formula.

On the Figs. 2 and 3 the comparison between number of processor cycles required to compute point scalar multiplication is shown. It is assumed that on short Weierstrass curve over F_p in every step one doubling and one addition must be computed (then such solution is resistant for side channel attacks) and any information about value of Y is not lost. The Brier-Joye ladder may be used only for XZ coordinates, so information about Y may be lost.



N_A T_M	1	2	3	4	5	6	7	8
1	1.46	1.90	2.34	2.78	3.22	3.66	4.10	4.54
2	1.24	1.52	1.81	2.10	2.38	2.67	2.95	3.24
4	1.07	1.23	1.40	1.57	1.74	1.91	2.07	2.24
8	0.95	1.05	1.14	1.23	1.32	1.42	1.51	1.60
16	0.89	0.94	0.99	1.04	1.08	1.13	1.18	1.23
32	0.85	0.88	0.90	0.93	0.95	0.98	1.00	1.03
64	0.84	0.85	0.86	0.87	0.89	0.90	0.91	0.93
128	0.83	0.83	0.84	0.85	0.85	0.86	0.87	0.87
192	0.82	0.83	0.83	0.84	0.84	0.85	0.85	0.85
224	0.82	0.83	0.83	0.83	0.84	0.84	0.85	0.85
256	0.82	0.83	0.83	0.83	0.84	0.84	0.84	0.85
384	0.82	0.82	0.83	0.83	0.83	0.83	0.83	0.84
512	0.82	0.82	0.82	0.83	0.83	0.83	0.83	0.83
521	0.82	0.82	0.82	0.83	0.83	0.83	0.83	0.83

Fig. 2. Values of $\frac{T_{TH}}{T_{SW,-3}}$ for different number of processor cycles of T_M and different number of additions N_A required for multiplication in F_{p^n}.



N_A T_M	1	2	3	4	5	6	7	8
1	1.33	1.73	2.13	2.53	2.93	3.33	3.73	4.13
2	1.10	1.35	1.61	1.86	2.11	2.37	2.62	2.87
4	0.93	1.07	1.22	1.37	1.51	1.66	1.80	1.95
8	0.82	0.90	0.98	1.06	1.14	1.22	1.30	1.37
16	0.76	0.80	0.84	0.88	0.92	0.97	1.01	1.05
32	0.73	0.75	0.77	0.79	0.81	0.83	0.85	0.87
64	0.71	0.72	0.73	0.74	0.75	0.76	0.77	0.78
128	0.70	0.71	0.71	0.72	0.72	0.73	0.73	0.74
192	0.70	0.70	0.71	0.71	0.71	0.72	0.72	0.72
224	0.70	0.70	0.70	0.71	0.71	0.71	0.72	0.72
256	0.70	0.70	0.70	0.70	0.71	0.71	0.71	0.72
384	0.70	0.70	0.70	0.70	0.70	0.70	0.71	0.71
512	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70
521	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70

Fig. 3. Values of $\frac{T_{TH}}{T_{SW}}$ for different number of processor cycles of T_M and different number of additions N_A required for multiplication in F_{p^n}.

The results strongly depend on the number of processor cycles T_M required for multiplication in F_p and number of additions N_A required for making multiplication in F_{p^2}. N_A depends on form of irreducible polynomial $f(t)$, for example $N_A = 2$ for $f(t) = t^2 + 1$ and $N_A = 5$ for $f(t) = t^2 + 11$. Let's see that in average case k in binary form has the same number of 0 and 1. If l is length in bits of

$Ord(P)$ for which $[k]P$ is computed, then complete formula requires about l point doublings and $\frac{l}{2}$ points additions. So computing point scalar multiplication of point $P' = \phi(P)$ on twisted Hessian curve over F_{p^2} requires:

$$T_{TH} = l \cdot \left(12\left(T_M + N_A\right) + 16\right) + \frac{l}{2} \cdot \left(12\left(T_M + N_A\right) + 16\right) = \\ = \frac{3}{2} l \cdot \left(12\left(T_M + N_A\right) + 16\right)$$

processor cycles.

For short Weierstrass curve over F_p computing point scalar multiplication of point P requires about:

1. If $A = -3$:

$$T_{SW,-3} = l \cdot (10T_M + 11) + l \cdot (14T_M + 7) = \\ = l \cdot (24T_M + 18)\ .$$

Hence,

$$\frac{T_{TH}}{T_{SW,-3}} = \frac{18\left(T_M + N_A\right) + 24}{24T_M + 18}\ .$$

2. If A is not of special form:

$$T_{SW} = l \cdot (12T_M + 12) + l \cdot (14T_M + 7) = \\ = l \cdot (26T_M + 19)$$

and

$$\frac{T_{TH}}{T_{SW,-3}} = \frac{18\left(T_M + N_A\right) + 24}{26T_M + 19}\ .$$

The longer is T_M, the better results solution presented in this article gives. The more additions are required for multiplication in F_{p^2}, the worse results proposed solution gives. In real applications multiplication in F_p requires often as many processor cycles as binary length of field is. For example for NIST P-256 curve T_M may take even 256 processor cycles, without cycles required for initialization and then presented solution may give better results than standard methods.

6. Conclusion

Using F_{p^2} a reasonable speed-up in hardware implementation of point scalar multiplication on elliptic curves can be achieved. The article shows how to find for some elliptic curves with cofactor 1 isomorphic twisted Hessian curves in fields extension. For two NIST curves over large prime fields: NIST P-224 and NIST P-256 the degree of such extension is 2, so it is possible to use twisted Hessian curve arithmetic over F_{p^2}. Such a solution is faster than classic approach up to 30%, if solution resistant for side channel attacks is necessary and coefficient A of short Weierstrass curve is not of special form. For $A = -3$, the presented solution may be up to 24% faster than classic approach.

Because implementation of parallel F_{p^2} arithmetic requires in hardware implementation much more resources than implementation of F_p arithmetic, the presented solution should be used only in some situations. For example, if necessary is to have arithmetic on two elliptic curves, which ensure different level of security. The first curve may use GLS method [9], because is very fast and on curve suitable for this method it is possible to use fast arithmetic in F_{p^2} and such a curve gives the security about p. Therefore, arithmetic on the second curve (which curve should give smaller security, for example about $\sqrt{p}$) may be implemented using the same F_{p^2} arithmetic, which is used for the first one. Then it is possible to use method presented in this article and such implementation is then faster than the classic one.

References

[1] E. Brier and M. Joye, "Weierstraß elliptic curves and side-channel attacks", in *Public Key Cryptography*, *LNCS*, vol. 2274, pp. 335–345. Springer, 2002 (doi: 10.007/3-540-45664-3_24).

[2] S. Ghosh, D. Mukhopadhyay, and D. Roychowdhury, "High speed flexible pairing cryptoprocessor on FPGA platform", in *Pairing-Based Cryptography – Pairing 2010*, S. Ghosh, D. Mukhopadhyay, and D. Roychowdhury, Eds. *LNCS*, vol. 6487, pp. 450–466. Springer, 2010 (doi: 10.1007/978/3/642-17455-1_28).

[3] H. Cohen and G. Frey, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. New York: Chapman & Hall/CRC, 2006.

[4] J. Renes, C. Costello, and L. Batina, "Complete addition formulas for prime order elliptic curves", Cryptology ePrint Archive, Report 2015/1060, 2015 [Online]. Available: https://eprint.iacr.org/2015/1060

[5] Explicit Formulas Database. [Online]. Available: http://hyperelliptic.org/EFD/g1p/index.html

[6] D. Bernstein, Ch. Chuengsatiansup, D. Kohel, and T. Lange, "Twisted Hessian curves", Cryptology ePrint Archive, Report 2015/781, 2015 [Online]. Available: https://eprint.iacr.org/2015/781

[7] N. P. Smart, "The Hessian form of an elliptic curve", in *Cryptographic Hardware and Embedded Systems – CHES 2001*, *LNCS*, vol. 2162, pp. 118–125. Springer, 2001 (doi: 10.1007/3-540-44709-1_11).

[8] "Recommended Elliptic Curves For Federal Government Use", National Institute of Standards and Technology, MA, USA, 1999 [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf, 1999

[9] S. Galbraith, X. Lin, and M. Scott, "Endomorphisms for faster elliptic curve cryptography on a large class of curves", *J. of Cryptol.*, vol. 24, no. 3, pp. 446–469, 2011.

Michał Wroński received in 2011 his M.Sc. degree in Computer Science from Military University of Technology in Warsaw, where he currently works. His research focuses on optimization of finite fields arithmetic and its hardware applications.

E-mail: michal.wronski@wat.edu.pl
Department of Mathematics and Cryptology
Military University of Technology
Kaliskiego st 2
00-908 Warsaw, Poland