Sergey Grebnev and Denis Dygin. Efficient Implementation of the GOST R 34.10 Digital Signature Scheme using Modern Approaches to Elliptic Curve Scalar Multiplication.

The digital signature scheme defined by Russian national standard GOST R 34.10 is based upon operations in a subgroup of order q of a group of points of an elliptic curve over a prime finite field, where bit-length of q is about 256. Following extensions to the standard recently proposed, it becomes possible to use subgroups of larger order, namely, $2^{508} < q < 2^{512}$. In this paper we show the possibilities of reducing the losses in performance naturally appearing when changing to the larger group sizes by implementing modern approaches to representation of elliptic curves and algorithms for scalar multiplication.

Scalar and multi-scalar multiplication require the most computational effort during generation and verification processes of a digital signature, respectively. We have chosen to use the following efficiently computable representations of the exponents: windowed multibase non-adjacent form (wmbNAF) as proposed by P. Longa for scalar multiplication and joint windowed multibase non-adjacent form (jwmbNAF) as proposed by M. Kalinin for multi-scalar multiplication. We have found that, considering integers of bit-length 256 to 512 and the natural set of bases $\{2, 3\}$ which allows for efficient usage of points doubling and tripling, optimal window sizes are 9 for wmbNAF and 6 or 9 for the two variants of jwmbNAF. Using these representations with some precomputaion, we achieve an average 25% improvement in performance over traditional NAF and JSF.

We experimented with extended coordinates on Hessian curves $x^3 + y^3 + 1 = 3dxy$, projective and inverted coordinates on Edwards curves $x^2 + y^2 = c^2(1 + dx^2y^2)$ and extended coordinates on twisted Edwards curves $ax^2 + y^2 = 1 + dx^2y^2$, comparing them with traditional projective Weierstrass coordinates.

It should be noticed that GOST R 34.10 strictly requires to produce a component of a signature as an X-coordinate of a point represented in affine Weierstrass form, therefore, we should take into account an overhead caused by the change of coordinates.

We list the best timings in milliseconds acquired for our implementation of the GOST R 34.10 scheme (signature generation/verification), built with Intel ++ Composer XE 2011 using gmp 5.0.2 multi-precision arithmetics library, on a single core of an Intel Xeon 3.0GHz CPU.

Curve representation	256 bit-q	512bit-q
Projective, Weierstrass	1.12/1.43	4.23/5.23
Extended, Hessian	0.7/0.98	2.65/3.54
Projective, Edwards	0.66/0.98	2.36/3.37
Inverted, Edwards	0.7/0.98	2.56/3.48
Extended, Twisted Edwards	0.7/0.98	2.52/3.38

Thus, we show that the relative deterioration in performance of the extended GOST R 34.10 scheme over the standard currently in force could be made as small as $3.5 - 4$. Considering the usage of different representations of an elliptic curve, both Hessian and Edwards representations show a 40% improvement over Weierstrass form. Nevertheless, practical differences in performance of various forms of Edwards curves are marginal. While Hessian representation is generally slower, it may perform better in a restricted environment.