UDC 004 056 55 DOI:10.30837/rt.2019.3.198.16

A. BESSALOV, Dr. Sc. (Engineering), L. KOVALCHUK, Dr. Sc. (Engineering),
N. KUCHYNSKA, Ph.D. (Engineering), O. TELIZHENKO

SECURITY OF MODIFIED DIGITAL PUBLIC-KEY SIGNATURE EDDSA

Introduction

The Ukrainian National Standard for Digital Signature DSTU 4145-2002 has been in use about 17 years.

During this time, significant changes have occurred in the field of information technology:

- new, more powerful cryptanalysis techniques have emerged and with the growing computing capabilities, are forcing the world to look for ways to increase the resilience of existing cryptosystems;
- in the last few years, practically all cryptographic algorithms used at this level are analyzed and evaluated in terms of their security with respect to the modern and perspective post-quantum methods of cryptanalysis, which leads to revision of these algorithms (usually, with increasing size of the parameters of such algorithms);
- DSTU 4145-2002 standard uses and refers to outdated Soviet and Russian block encryption and hash functions standards, while Ukraine's 2014 adopted DSTU 7564 and DSTU 7624 National Standards, which have significantly higher performance and cryptographic security;
- new algebraic objects have appeared the use of which has many advantages (both in performance and cryptographic security) in building different cryptosystems;
- a new cryptographic algorithms, including auxiliary ones, were offered which have numerous advantages over older ones and are gradually displace them from use.

In addition, a significant drawback of DSTU 4145-2002 is that it (the only one all over the world) recommends the use of only elliptic curves over the finite field of characteristic 2, which makes the signature of this standard more than 20% slower than any other states, including Russia and Belarus.

All these changes directly affect the implementation of the current National Standard for Digital Signature DSTU 4145-2002 and indicate the need for its modernization.

An important task in choosing asymmetric crypto-algorithms is selection of parameters size and its justification. Today an international organizations, such as NIST, ETSI, and ENISA, have raised appropriate parameter requirements that can be used in the transitional to post-quantum and post-quantum period. [1], [2].

Due to the need to revise and update national digital signature standard DSTU 4145-2002 [3], the authors considered several digital signature constructions. Among the requirements to modern public-key signatures it is worth to highlight at least 128-bit security, fast signing and fast signature verification, fast keys generation, foolproof session keys, collision resistance, secure software implementation, etc. There are a lot of obvious variants in classic and elliptic signature systems, ElGamal, Schnorr`s, ECDSA, etc, which can be used in transitional to post quantum period.

This paper introduces one of possible modifications for signature schemes based on The Edwards-curve Digital Signature Algorithm (EdDSA) proposed in [4] and specified in IETF RFC 8032 [5], which is a variant of Schnorr's signature system with (possibly twisted) Edwards curves. The main advantages of the modification proposed in this work are:

1) secure even in case of generator faults [6];
2) signature performance doesn't depend on message length;
3) security against related-key attacks.

Related work

The equation of the elliptic curve in the form, which later took the name “Edwards form”, was suggested in paper [11].The isomorphism (under certain conditions) between the curves in the Weierstrass form and in the Edwards form was proved. Howewer the curves, proposed in [11] were weak from the cryptographic point of view. And paper [11] was quickly followed by paper [12], where the Edwards curves were modified by the introduction of the certain parameter.

Hereafter, for simplification, we will consider curve E, assigned over finite field F_p, for an odd prime number p and $a,d \in F_p^* \; a \neq d \,, d \neq 1$:

$$E : ax^2 + y^2 = 1 + dx^2 y^2 , \tag{1}$$

The main differences (almost all of which are advantages) of the Edwards curve compared with the Weierstrass curves are the following.

1. Universality of the addition law. Indeed, the operations of different points addition and doubling a point are assigned by he same formulas:

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1 y_2 + x_2 y_1}{1 + dx_1 x_2 y_1 y_2}, \frac{x_1 y_2 - ax_2 y_1}{1 - dx_1 x_2 y_1 y_2} \right) \tag{2}$$

2. The absence of “the point on infinity”. Thus, the neutral element is a usual point of the Edwards curve with coordinates $O = (0,1)$, which obviously fulfill equation (1).

3. Group E_p of points of elliptic curve (1) with Legendre symbol $\left(\frac{ad}{p} \right) = -1$ is always cyclic.

4. The order of the group E_p is always divided by 4. The property of the Edwards curve can be considered an insignificant disadvantage due to the fact that its subgroup of the large prime order, on the basis of which cryptosystems are constructed, has at least 4 times less points than E_p, in other words at least three-quarters of the points of the group are “extra”.

5. The high performance of points addition. This property is one of the most important advantages of the Edwads curve. Thus, approximately 1.5 bit operations less required for two (different) points addition of the Edwads curve than for the same operation for the Weierstrass curve. Meanwhile for the doubling of the points the bit operation number is even less.

6. Uniformity of the addition law. The formulas for the addition and doubling of points are the same for the Edwards curve. This increases security of cryptosystems on the Edwards curves to timing and capacitive attacks, aimed to determine the number of such operations.

In this part we briefly specify EdDSA signature system according to [4], [5], but in more formal way. As the authors of algorithm pointed, the advantages with EdDSA are as follows:

- EdDSA provides high performance on a variety of platforms and the use of a unique random number for each signature is not required;
- specified in RFC 8032 EdDSA uses relatively small (in comparison with postquantum algorithms) public keys (32 or 57 bytes) and signatures (64 or 114 bytes) for Ed25519, providing approximately 128 bits of security (uses Edwards version of Curve25519) and Ed448, which provides approximately 224 bits of security, respectively;
- the formulas are "complete", i.e., they are valid for all points on the curve, with no exceptions. This obviates the need for EdDSA to perform expensive point validation on untrusted public values;
- EdDSA provides collision resilience, meaning that hash-function collisions do not break this system (only holds for PureEdDSA) and it is more resilient to side-channel attacks.

EdDSA needs to be instantiated with certain parameters. The scheme has next parameters:

- an odd prime number p;
- an integer b with $2^{b-1} > p$,
- a $(b-1)$-bit encoding of elements of finite field F_p,
- a cryptographic hash function H producing $2b$-bit output;
- a non-square element d of F_p and a non-zero square element a of F_p (the usual recommendation for best performance is

$$a = \begin{cases} -1, \text{if } p \equiv 1 \bmod 4, \\ 1, \text{if } p \equiv 3 \bmod 4; \end{cases} \tag{3}$$

- a prime n between 2^{b-4} and 2^{b-3} under special condition and the point $P \neq (0,1)$.

Note, that the set (1) defines complete Edwards curve (according to classification [7]), because $a \in Q_p$, according to (3), where Q_p is the set of quadratic residues modulo p. Due to the condition $d \notin \{0,-1\}$ and $d \notin Q_p$, the set (3) forms a group of points with affine coordinates with neutral element $O = (0,1)$ under the addition law (2) [7], [8].

There is also extra constraint for the base point P of elliptic curve E, with order n of the point P, i.e. $nP = 0$, where n is a large prime. So the number of points on the curve is $|E| = 2^c n$ and a cofactor is 2^c with integer c, $c \in \{2,3\}$ (according to choice of elliptic curve).

An EdDSA secret key is a b-bit string k. The hash $H(k) = (h_0, h_1, ..., h_{2b-1})$ determines an integer $e = 2^{b-2} + \sum_{c \le i < (b-c)} 2^i h_i$. The knowledge of e is sufficient for producing valid signatures, which justifies considering e as the signing key. The public key $Q = eP$ is a point on the curve (1).

An element $\overline{(x,y)} \in E$ is encoded as a b-bit string (x,y), namely y is encoded by $(b-1)$-bit string and concatenated with one bit that is 1 if x is negative and 0 if x is not negative [4].

Note that the encoding of elements of the finite field F_p is defined specifically as, x is "negative" if the (b-1)-bit encoding of x is lexicographically larger than the (b-1)-bit encoding of $-x$ (in little-endian form).

The signature of a message M under this secret key k is defined as follows.

Signature algorithm:

1) compute $r = H(h_b, ..., h_{2b-1}, M) \in \{0, ..., 2^{2b-1}\}$;
2) compute $R = rP$;
3) compute $h = H(\overline{R}, \overline{Q}, M)$ and convert it into integer;
4) compute $s = (r + eh) \bmod n$.

The signature of M is $DS = (\overline{R}, s)$

Verification algorithm:

1) compute $h' = H(\overline{R}, \overline{Q}, M)$ and convert into integer;
2) check $sP = R + h'Q$.

The verifier rejects the alleged signature if the parsing fails or if the group equation does not hold.

EdDSA is based on digital signature scheme that was first designed by Schnorr [9]. A main demand when using this kind of signatures is that r has to be chosen unpredictably [4]. Indeed, if r can be guessed correctly for an existing signature, then the signing key a_k can be simply computed as $e=(s-r)h^{-1} \bmod n$ using the extended Euclidean algorithm.

Legitimate users choose $Q=eP$, where e is a random secret; the derivation of e from $H(k)$ ensures adequate randomness. These users have negligible chance of generating any particular multiple of P targeted by the attacker. The chance of the attacker randomly guessing e is much smaller than the chance of the attacker computing e by known discrete-logarithm algorithms; standard elliptic-curve security criteria are designed so that the latter algorithms have negligible chance of succeeding in any reasonable amount of time.

Furthermore, if the same nonce value k (with unknown hash $H(k)=(h_0,h_1,...,h_{2b-1})$) has been used for generating signatures of different messages M_1, M_2, and $h_1=H(\overline{R},\overline{Q},M_1)$, $h_2=H(\overline{R},\overline{Q},M_2)$ the signing key e can be tried to recover by eavesdropper as

$$e=((s_1-s_2)-(r_2-r_1))(h_1-h_2)^{-1} \bmod n,$$

where $r_1=H(h_b,...,h_{2b-1},M_1)$ and $r_2=H(h_b,...,h_{2b-1},M_2)$. But values r_1 and r_2 are still different and unknown. So the secret key cannot be found even in case of generator`s faults, when it produces the same nonce k for two different messages. That is one of the essential differences of EdDSA from ECDSA [10].

Modification of EdDSA

This work considers signature construction, with longtime signature key. In our notation private key is $e \in F_p$, $e \neq \pm 1$ and public key $Q=eP$ is a point on the elliptic curve E over F_p. Note that not only the curve (1), but any suitable Edwards elliptic curve with $|E|=2^c n$ and small cofactor 2^c can be used in this construction, i.e.

$$E=\{(x,y)\in F_p \times F_p : x^2+ay^2=1+dx^2y^2\}. \tag{4}$$

Under the condition $(ad)\notin Q_p$ the set (4) forms a group of points with affine coordinates with neutral element $O=(1,0)$ according the addition law [7]:

$$(x_1,y_1)+(x_2,y_2)=\left(\frac{x_1x_2-ay_2y_1}{1-dx_1x_2y_1y_2},\frac{x_1y_2+x_2y_1}{1+dx_1x_2y_1y_2}\right). \tag{5}$$

The modified scheme of EdDSA make use of next parameters:

- an odd prime number p, defining the underlying field;
- an integer b with $2^{b-1}>p$ and a $(b-1)$-bit encoding of elements of finite field F_p,
- a cryptographic hash function H producing bit strings of the length not less than b.

The signature and verification algorithm for a message M under this secret key e defined as follows.

Signature algorithm:

1) generate random k: $1<k<n$;

2) compute $r=H(k \,||\, H(M))$;

3) compute point $R=rP$;

4) encode $\overline{R},\overline{Q}$ and compute $h=H(\overline{R},\overline{Q},H(M))$;

5) compute $s = (r + eh) \bmod n$.

The signature of M is $DS = (\overline{R}, s)$.

Verification algorithm:

1) compute $h' = H(\overline{R}, \overline{Q}, H(M))$ and convert into integer;

2) check $sP = R + h'Q$.

Correctness of proposed signature can be proved with next equality

$$R + H(\overline{R}, \overline{Q}, H(M))Q = rP + H(\overline{R}, \overline{Q}, H(M))dP = (r + dH(\overline{R}, \overline{Q}, H(M)))P = sP.$$

Reducing the secret key search to solution of DLP for Modification of EdDSA

In this section, we show that the task of the secret key recovery in the proposed digital signature algorithm is polynomially reduced to the DLP problem./

Theorem: the problem of obtaining secret key from given pair message M and its signature DS in Modification of EdDSA is not easier (not more efficient) than solution of discrete logarithm problem.

Proof: let we have an Oracle O, which for given message M and its signature $DS = (\overline{R}, s)$ returns secret key e. Let we have some point $R = rP$ for some unknown r for elliptic curve base point P, with $ordP = n$, i.e. $nP = 0$. Then we can construct the next algorithm, which finds r in polynomial time, using an Oracle O.

Algorithm input: P, R.

Algorithm output: r (where $R = rP$).

Algorithm.

1) generate random $s : 1 < s < n$ and $h : 1 < h < n$;

2) compute $h^{-1} \bmod n$;

3) compute $Q = h^{-1}sP - h^{-1}P$;

4) query the oracle O and gets its answer $O(P, R, s, h, Q) = a_k$;

5) compute $r = s - a_k h$.

It should be marked, that according to the Algorithm it follows that $Q = a_k P$, where $s = r + eh$. Then $sP = rP - hQ$ and $Q = h^{-1}(s - r)P$.◄

Conclusion

The signature algorithm, considered in this work, is promising and provable secure against leakage of secret key. The main advantage over the original signature EdDSA is to reduce the time of its work, as well as that the time of work does not depend on the length of the message.

Besides it is still secure even in case of PRNG faults. In further analysis, it is desirable to show that the algorithm is resistant to a keyless subscription.

References:

1. ETSI GR QSC 001 V.1.1.1 (2016-07). Quntum-Safe Cryptography (QSC); Quantum-safe algorithmic framework. Access mode: https://portal.etsi.org/webapp/workProgram/Report_WorkItem.asp?wki_id=46690 30.10.2016.

2. Quantum Safe Cryptography and Security. An introduction, benefits, enablers and challenges. ETSI White Paper No. 8, 2015. Access mode: http://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf 30.10.2016.

3. DSTU 4145-2002. Information Technology. Cryptographic protection of information. Digital signature based on elliptic curves.

4. D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. High-speed high-security signatures // Proc. of the 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'11), Nara, Japan, ser. Lecture Notes in Computer Science, vol. 6917. Springer-Verlag, September 2011, pp.124–142.

5. S. Josefsson, I. LiusvaaraRFC 8032 Edwards-Curve Digital Signature Algorithm (EdDSA). January 2017 DOI: 10.17487/RFC8032

6. Ambrose, Christopher & Bos, Joppe & Fay, Björn & Joye, Marc & Lochter, Manfred & Murray, Bruce. (2018). Differential Attacks on Deterministic Signatures.

7. Bessalov A.V. (2017). Ellipticheskie krivyie v forme Edvardsa i kriptografiia: monografiya. Kyiv : KPI im. Igoria Sikorskogo ; Politekhnik». 272.

8. Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, Christiane Peters. Twisted Edwards curves // Africacrypt 2008, 389–405. http://eprint.iacr.org/2008/013

9. Claus P. Schnorr. Efficient Identification and Signatures forSmart Cards // Advances in Cryptology. CRYPTO ’89. NewYork: Springer, 1990, pp. 239–252.

10. Hartl Alexander & Annessi Robert & Zseby Tanja. (2017). A Subliminal Channel in EdDSA: Information Leakage with High-Speed Signatures. 67-78.

11. Edwards H.M. (2007). A normal form for elliptic curves. Bulletin of the American Mathematical Society, V. 44, 393-422.

12. Bernstein D.J., Lange T. (2007) Faster Addition and Doubling on Elliptic Curves // Kurosawa K. (eds) Advances in Cryptology – ASIACRYPT 2007. Lecture Notes in Computer Science, vol 4833. Springer, Berlin, Heidelberg.

Institute of Physics and Technology NTUU
“Igor Sikorsky Kyiv polytechnic institute”

Received 11.09.2019