Bitcoin Security with a Twisted Edwards Curve

Meryem Cherkaoui Semmouni, Abderrahmane Nitaj, Mostafa Belkasmi

Bitcoin Security with a Twisted Edwards Curve

Meryem Cherkaoui Semmouni[1], Abderrahmane Nitaj[2], and Mostafa Belkasmi[1]

[1] ENSIAS, Mohammed V University in Rabat, Morocco
meryem.semmouni@um5s.net.ma, m.belkasmi@um5s.net.ma
[2] LMNO, caen university, France
abderrahmane.nitaj@unicaen.fr
http://www.math.unicaen.fr/~nitaj

Abstract. The security of the Bitcoin cryptocurrency system depends on the Koblitz curve secp256k1 combined with the digital signature ECDSA and the hash function SHA-256. In this paper, we show that the security of Bitcoin with ECDSA and secp256k1 is not optimal and present a detailed study of the efficiency of Bitcoin with the digital signature algorithm Ed25519 combined with the twisted Edwards curve CurveEd25519 and the hash function SHA-512. We show that Bitcoin is more secure and more efficient with the digital signature algorithm Ed25519 and the twisted Edwards curve CurveEd25519.
Subject Classifications: 94A60

Keywords: Cryptography · cryptocurrency · Bitcoin · Security · Twisted Edwards curves · Signature

1 Introduction

The progress of the new technology of information is changing the way of our individual transfer cash, from paper to digital cash or electronic money (e-money). Electronic money is a substitute for cash. It is stored in electronic devices on remote servers. The use of e-money is highly encouraged in several countries and aims to create new, safe and practical development services. The transactions are becoming easier and cheaper, online payments and operations on our accounts are possible at anytime and anywhere. Meanwhile, the security of electronic systems becomes a serious concern. The amount of frauds, the attacks launched by various hackers, the problems of confidentiality and authentication, are of great danger for electronic systems. To overcome these problems, cryptography offers many solutions. Cryptography is used to secure e-commerce, the cloud, internet communications, and to protect sensitive banking, military information and information systems.

Another important application of cryptography is to secure Bitcoin system. Bitcoin is a peer-to-peer network without any central authority such as banks or governments. It was presented in 2008 by Satoshi Nakomoto [26] and launched in 2009. To authorize payments or transfers, Bitcoin uses the Elliptic Curve Digital

Signature Algorithm (ECDSA) [17] with the hash function SHA-256 [18], and the Koblitz curve secp256k1 with the equation

$$secp256k1: \ y^2 = x^3 + 7 \pmod{p_1}, \quad p_1 = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1.$$

The Koblitz curve secp256k1 was proposed in 2000 by the Standards for Efficient Cryptography Group of Certicom in the standards for efficient cryptography SEC2 [10] and used in the Bitcoin system since 2009. The Koblitz curve secp256k1 seems having many advantages when used in industrial applications, especially efficiency, security and shortness of the key.

In this paper, we study the possibility of using the digital signature Ed25519 [4] based on the twisted Edwards curve CurveEd25519 with the equation

$$CurveEd25519: \ -x^2 + y^2 = 1 - \frac{121665}{121666} x^2 y^2 \pmod{p_2}, \quad p_2 = 2^{255} - 19,$$

to secure Bitcoin instead of ECDSA with the Koblitz curve secp256k1. We compare the security and the efficiency of operations on the curves secp256k1 and CurveEd25519, and then the security and the efficiency of the digital signatures ECDSA with secp256k1 and SHA-256 and Ed25519 with CurveEd25519 and SHA-512.

Our comparison of the security of secp256k1 and CurveEd25519 is based on the study of the resistance of both curves to the attacks on the elliptic curve discrete logarithm ECDLP. Our study shows that secp256k1 presents some vulnerabilities to the complex-multiplication field discriminant as well as to Pollard's rho attack while CurveEd25519 is safe.

Similarly, we study the efficiency of the arithmetical operations on the curves secp256k1 and CurveEd25519 over their finite fields. We compare the cost of adding two points or doubling a point on both curves. We find that the arithmetic of the twisted Edwards curve CurveEd25519 is more efficient than the arithmetic of the Koblitz curve secp256k1.

Moreover, the digital signature Ed25519 uses the hash function SHA-512 which presents more security and is more sustainable than the hash function SHA-256 used in ECDSA for the Bitcoin system.

The former comparison suggests that the digital signature Ed25519 is more suitable for use in the Bitcoin system than ECDSA.

The rest of this paper is organized as follows. In Section 2, we recall some facts on Bitcoin, secp256k1, CurveEd25519, and Ed25519. In Section 3, we study and compare the resistance of secp256k1, CurveEd25519 to cryptanalytical attacks on the elliptic curve logarithm problem ECDLP. In Section 4, we study the efficiency of the arithmetic operations on the curve CurveEd25519. In section 5, we resume the comparison of the digital signatures ECDSA and Ed25519. We conclude the paper in Section 6.

2 Preliminaries

2.1 Description of Bitcoin

Bitcoin is a digital currency and a peer-to-peer payment system developed by an anonymous individual or group with the pseudonym Satoshi Nakamoto [26] in 2008. Bitcoin users communicate with each other using a secure collection of open source technologies. As a peer-to-peer system, there is no central authority or central server. A public distributed ledger blockchain is available to everyone, where the verified transaction is registered, the verification is done on network nodes. Bitcoins are created by a process called mining, and any participant in the bitcoin network may operate as a miner depending on its computer's ability to process operations on bitcoins. The transfer of bitcoins between users requires to use cryptographic algorithms to prove ownership of the bitcoins being transferred. The Bitcoin network security is based on the digital signature scheme known as the Elliptic Curve Digital Signature Algorithm (ECDSA) with the Koblitz curve secp256k1 to verify ownership transactions on the network, combined with the hash function SHA-256.

2.2 Description of the Koblitz curve secp256k1

In Bitcoin system, the Elliptic Curve Digital Signature Algorithm (ECDSA) is used to verify bitcoin transactions. ECDSA is an adaptation of the Digital Signature Algorithm (DSA) using a Koblitz elliptic curve [17]. The elliptic curve used for ECDSA in Bitcoin system is the elliptic curve secp256k1, defined by the Standards for Efficient Cryptography Group (SECG) [10], with the following parameters:

- the prime number: $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$,
- the equation: $y^2 \equiv x^3 + 7 \pmod{p}$,
- the base point: $P = (55066263022277343669578718895168534326250603453$
  $77759417550018736038911672924 0,$
  $32670510020758816978083085130507043184471273380659243$
  $27593890433575733748242 4),$
- the order n of P: $n = 2^{256} - 432420386565659656852420866394968145599$.

Adjoining the point at infinity $\mathcal{O}$, the curve secp256k1 has n solutions. This curve is also used as standard by other blockchain systems such as Ethereum and Zcash.

2.3 Description of ECDSA

For Bitcoin system, ECDSA is based on the Koblitz curve secp256k1 and on the cryptographic hash function SHA-256. The implementation of ECDSA in Bitcoin system is composed by three algorithms, key generation, signing and verification.

1. **ECDSA Key generation algorithm.**
   - Choose a random integer $d \in [1, n-1]$.
   - Compute $Q = (x_Q, y_Q) = dP$ on the curve secp256k1.
   - The public key is Q and the private key is d.
2. **ECDSA Signing.** Given a message m to be signed, the private key d and a hash function H,
   - Choose a random integer $k \in [1, n-1]$.
   - Compute $G = (x_G, y_G) = kP$ on the curve secp256k1.
   - Compute $r \equiv x_G \pmod{n}$. If $r = 0$, choose another k and recompute G and r.
   - Compute $s \equiv k^{-1}(H(m) + dr) \pmod{n}$.
   - The signature is the pair (r, s).
3. **ECDSA Verification.** Given a signature (r, s) and a hash function H,
   - Compute $w \equiv s^{-1} \pmod{n}$.
   - Compute $u_1 \equiv wH(m) \pmod{n}$ and $u_2 \equiv wr \pmod{n}$.
   - Compute $(x_0, y_0) = u_1 P + u_2 Q$ on the curve secp256k1.
   - Accept the signature if $x_0 \equiv r \pmod{n}$.

2.4 Description of the twisted Edwards Curve CurveEd25519

In 2007, Edwards [14], introduced a new normal form for elliptic curves. In a series of papers, Bernstein et al. [3,4] generalized the Edwards form to twisted Edwards curves with the equation

$$ax^2 + y^2 = 1 + dx^2y^2,\ a \neq d,\ ad \neq 0,$$

with a unique formula for both addition and doubling laws. Indeed, the sum of two points (x_1, y_1) and (x_2, y_2) on a twisted Edwards curve is :

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2}\right).$$

The point (0,1) is the neutral element of the addition law, and the inverse of a point (x_1, y_1) on E is simply $(-x_1, y_1)$.

In 2009, Bernstein [5] proposed Curve25519 to speed the computation of the Diffie-Hellman key exchange. Curve25519 is a Montgomery elliptic curve at the 128-bit security level with the equation

$$Curve25519 :\ v^2 = u^3 + 486662u^2 + u \pmod{p},\ p = 2^{255} - 19.$$

The security of the curve Curve25519 was studied by Bernstein in [5] who concluded that the arithmetic of this curve is fast and the security is optimal. Using a birational equivalence, Curve25519 can be represented in a twisted Edwards form. Let $Bv^2 = u^3 + Au^2 + u$ be the equation of a Montgomery elliptic curve. For $v(u+1) \neq 0$, define

$$X = \frac{u}{v},\ Y = \frac{u-1}{u+1},\ a = \frac{A+2}{B},\ d := \frac{A-2}{B}.$$

Then $aX^2 + Y^2 = 1 + dX^2Y^2$ represents the equation of a twisted Edwards curve. For $A = 486662$ and $B = 1$ as in Curve25519, we get the following equation $486664X^2 + Y^2 = 1 + 486660X^2Y^2$, or equivalently

$$-(-486664)X^2 + Y^2 = 1 - \frac{486660}{486664}(-486664)X^2Y^2.$$

Since -486664 is a square in $\mathbb{F}_p$, then $-486664 \equiv s^2 \pmod{p}$ with

$$\begin{aligned} s =& 5104256939916053613020613523314632928415220225303463182268 \\ & 183378666877215207. \end{aligned}$$

Hence, the former equation can be rewritten as

$$-(sX)^2 + Y^2 = 1 - \frac{486660}{486664}(sX)^2X^2.$$

Using the birational transformation $(x, y) = (sX, Y)$, the equation can be rewritten as the equation of the curve CurveEd25519:

$$CurveEd25519: \ -x^2 + y^2 = 1 - \frac{486660}{486664}x^2y^2. \tag{1}$$

This is the equation of the twisted Edwards curve used in [6] to construct the digital signature Ed25519. The corresponding parameters are as follows.

- the prime number: $p = 2^{255} - 19$,
- the equation: $CurveEd25519: \ -x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2 \pmod{p}$,
- the base point: $B = (15112221349535400772501151409588531511454012693041857206046113283949847762202,$
  $46316835694926478169428394003475163141307993866256225615783033603165251855960)$,
- the order n of B: $n = 2^{252} + 27742317777372353535851937790883648493$.

2.5 Description of the Digital Signature Ed25519

In 2011, Bernstein et al. [6] proposed the digital signature scheme Ed25519, an instance of the Elliptic Curve signature scheme EdDSA. The arithmetical operations of Ed25519 are based on the fast twisted Edwards curve CurveEd25519 with the equation (1) modulo $p = 2^{255} - 19$. The digital signature Ed25519 uses several domain parameters:

- Finite field $\mathbb{F}_p$ with $p = 2^{255} - 19$ and bit-size $b = 256$.
- Twisted Edwards curve with the equation (1).
- Base point B given in(2.4) with order n.
- Hash function H that produces a $2b$-bits output such as SHA-512.

Ed25519 consists in applying three algorithms to generate the public and the private keys, to sign a message m and to verify the signature.

1. **Ed25519 Key generation algorithm.**
   - Choose a random integer $k \in [1, n-1]$.
   - Compute $H(k) = (h_0, h_1, \ldots, h_{2b-1})$ in binary representation.
   - Compute the integer $a = 2^{b-2} + \sum_{i=3}^{b-3} 2^i h_i$.
   - Compute the public key $A = aB$ on the curve CurveEd25519.
2. **Ed25519 Signing.** Given a message m to be signed and a hash function H,
   - Compute $r = H(h_b, \ldots, h_{2b-1}, m)$ as an integer modulo n.
   - Compute $R = rB$ on the curve CurveEd25519.
   - Compute $h = H(R, A, M)$ as an integer.
   - Compute $s = (r + ha) \pmod{n}$.
   - The signature is the pair (R, s).
3. **Ed25519 Verification.** Given a signature (R, s) and a hash function H,
   - Compute $h = H(R, A, M)$ as an integer.
   - Compute $U = 8sB$ on the curve CurveEd25519.
   - Compute $V = 8R + 8hA$ on the curve CurveEd25519.
   - Accept the signature if $U = V$.

3 Resistance of secp256k1 and CurveEd25519 to cryptanalytical attacks

The Koblitz curve secp256k1 is defined by the equation

$$secp256k1: \ y^2 = x^3 + 7 \pmod{p_1}, \quad p_1 = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1.$$

The order of its base point and the order of the curve secp256k1 are

$$n_1 = 2^{256} - 432420386565659656852420866394968145599,$$
$$\#secp256k1(\mathbb{F}_{p_1}) = n_1.$$

The twisted Edwards curve CurveEd25519 is defined by the equation

$$CurveEd25519: \ -x^2 + y^2 = 1 - \frac{121665}{121666} x^2 y^2 \pmod{p_2}, \quad p_2 = 2^{255} - 19.$$

The order of its base point and the order of the curve CurveEd25519 are

$$n_2 = 2^{252} + 27742317777372353535851937790883648493,$$
$$\#CurveEd25519(\mathbb{F}_{p_2}) = 8n_2.$$

The security of elliptic curve cryptosystems is based on the computational intractability of the Elliptic Curve Discrete Logarithm Problem(ECDLP): Given an elliptic curve E and two points P and Q on E such that $Q = kP$, find k. The hardness of the ECDLP depends on certain properties of the elliptic curve E and the base point $P \in E(\mathbb{F}_p)$. In the rest of this section, we give a detailed study of resistance of the Koblitz curve secp256k1 and the twisted Edwards Curve CurveEd25519 to various cryptanalytic attacks.

3.1 Complex-multiplication field discriminants

If E is an elliptic curve over a finite field $\mathbb{F}_p$, then the number of rational point is $\#E(\mathbb{F}_p) = p + 1 - t$ where t is the trace of the Frobenius endomorphism, which by Hasse's Theorem satisfies $-2\sqrt{p} < t < 2\sqrt{p}$. Then $t^2 - 4p < 0$ and we can write

$$t^2 - 4p = -s^2 d,$$

where d is square-free. Then s is the largest integer such that s^2 divides $t^2 - 4p$. The complex-multiplication field discriminant of the elliptic curve E is the integer D with

$$D = \begin{cases} \frac{t^2-4p}{s^2} & \text{if } \frac{t^2-4p}{s^2} \equiv 1 \pmod 4 \\ \frac{4\left(t^2-4p\right)}{s^2} & \text{if } \frac{t^2-4p}{s^2} \not\equiv 1 \pmod 4. \end{cases}$$

The complex-multiplication field discriminant D is considered as a security parameter by the standard Brainpool [13] and by the SafeCurves web page [9]. It is required that $|D|$ should be large, typically $|D| > 2^{100}$.

For the curve secp256k1, we have

$$t_1 = p_1 + 1 - \#secp256k1(\mathbb{F}_{p_1}) = 432420386565659656852420866390673177327,$$

and

$$t_1^2 - 4p_1 = -(79 \cdot 349 \cdot 2698097 \cdot 1359580455984873519493666411)^2 \cdot 3.$$

It follows that the complex-multiplication field discriminant is $D_1 = -3$ which is much smaller than the required lower bound 2^{100}.

The twisted Edwards curve CurveEd25519 is birrationally equivalente to the Montgomery curve with the equation

$$Curve25519: \ v^2 = u^3 + 486662u^2 + u \pmod{p_2}, \ p_2 = 2^{255} - 19.$$

For Curve25519, we have

$$t_2 = p_2 + 1 - \#Curve25519(\mathbb{F}_{p_2}) = -221938542218978828286815502327069187962,$$

and

$$\begin{aligned} t_2^2 - 4p_2 = &- 2^4 \cdot 16451 \cdot 8312956054562778877481 \\ &\cdot 8332672572899929670107862883852213333365522455698 7. \end{aligned}$$

Then, the complex-multiplication field discriminant is $D_2 = \frac{4\left(t_2^2-4p_2\right)}{2^4}$ and satisfies $|D_2| > 2^{254}$ which is much larger than the required bound 2^{100}.

As a consequence of the former study, the curve CurveEd25519 is much stronger than the curve secp256k1 to the complex-multiplication field discriminant criterion.

3.2 Pohlig-Hellman attack

The Pohlig-Hellman algorithm [28] is an algorithm devoted to solve the discrete logarithm problem on finite fields or elliptic curves. For an elliptic curve with base point P of order n, the attack reduces the problem of finding the discrete logarithm k satisfying $Q = kP$ first in recovering k modulo each of the prime factors of the order n of P, and second in applying the Chinese Remainder Theorem to recover k entirely modulo n. The expected running time of Pohlig-Hellman algorithm is $O\left(\sqrt{n'}\right)$ where n' is the largest prime factor of n. In order to maximize resistance to the Pohlig-Hellman attack, the elliptic curve parameters should be selected so that the order n of the base point P is divisible by a large prime. For the curves secp256k1 and CurveEd25519, the orders n_1 and n_2 of the base points are prime numbers. This increase the resistance of both curves to the Pohlig-Hellman attack.

3.3 Pollard's rho attack

This algorithm was presented by Pollard [29] in 1978 to attack the discrete logarithm problem in finite fields. Since then, it was adapted to attack the elliptic curve discrete logarithm problem. The main idea behind Pollard's rho algorithm is to find distinct pairs (u, v) and (u', v') of integers such that $uP + vQ = u'P + v'Q$ from which we deduce $k = (v' - v)(u - u')^{-1} \pmod{n}$ when $\gcd(n, u - u') = 1$. Such an occurrence is called a collision and can be applied to the curves secp256k1 and CurveEd25519 since the order of their base points is a prime number in both cases. The expected number of iterations before a collision is obtained is approximately $O\left(\sqrt{\frac{\pi n}{2}}\right)$ [20] and requires approximately $O\left(\sqrt{\frac{\pi n}{2}}\right)$ amount of storage. For the curve secp256k1, we have $\sqrt{\frac{\pi n_1}{2}} \approx 2^{128}$, and for CurveEd25519, we have $\sqrt{\frac{\pi n_2}{2}} \approx 2^{126}$. Hence, both curves have high level bit-security and seem resistant to Pollard's rho method. However, the curve secp256k1 has j-invariant 0 and has specific properties such as efficient computation of endomorphisms of certain multiples of points. This can be turned out to a vulnerability by speeding Pollard's rho algorithm (see [2] for more details and discussions). For the curve CurveEd25519, via the Montgomery curve Curve25519, the j-invariant is not 0 so that the speed up of Pollard's rho algorithm is not possible.

Summarising the former comparison, the curve secp256k1 is more sensitive to Pollard's rho algorithm than the curve CurveEd25519.

3.4 Anomalous attack

An elliptic curve E over a prime field $\mathbb{F}_p$ is anomalous if $\#E(\mathbb{F}_p) = p$. For anomalous curves, the group $E(\mathbb{F}_p)$ is cyclic since it has prime order, and hence $E(\mathbb{F}_p)$ is isomorphic to the additive group $\mathbb{F}_p^+$ of integers modulo p. Semaev [33], Smart [35], and Satoh and Araki[31] independently proposed an efficient attack for the ECDLP in the anomalous case which reduces the ECDLP in an elliptic

curve to addition in the additive group $\mathbb{F}_p^+$ by a lifting modulo p^2. The curves secp256k1 and CurveEd25519 are resistant to the anomalous attack since the prime moduli p_1, p_2 are different from the number of points of both curves, more specifically, $\#secp256k1(\mathbb{F}_{p_1}) = n_1 \neq p_1$, and $\#CurveEd25519(\mathbb{F}_{p_2}) = 8n_2 \neq p_2$.

3.5 The Frey-Rück attack

Frey and Rück [19] described a method based on the Tate-Lichtenbaum pairing to reduce ECDLP on the elliptic curve E over $\mathbb{F}_p$ to the discrete logarithm problem into the multiplicative group $\mathbb{F}_{p^k}^*$ for some extension of the base field $\mathbb{F}_p$. For $k \leq 30$, the index calculus method can solve the DLP in subexponential time in the multiplicative group $\mathbb{F}_{p^k}^*$. In general, the embedding degree is usually enormous, and the criterion to avoid the attack is that the order n of the base point of the elliptic curve satisfies $n| \left(p^k - 1\right)$ only for large values of k. The curves secp256k1 and CurveEd25519 are such that $n_1 \nmid \left(p_1^k - 1\right)$ and $n_2 \nmid \left(p_2^k - 1\right)$ for $k \leq 10^6$. As a consequence, both curves are resistant to the Frey-Rück attack.

3.6 MOV supersingular attack

An elliptic curve E over a finite field $\mathbb{F}_p$ is called supersingular if $\#E(\mathbb{F}_p) = p + 1$. Menezes, Okamoto and Vanstone [24] described how the Weil pairing can be used to reduce ECDLP on the elliptic curve E over $\mathbb{F}_p$ to the discrete logarithm problem into the multiplicative group $\mathbb{F}_{p^k}^*$ for $k \leq 6$, where the index calculus method can solve the DLP in subexponential time. This implies that supersingular elliptic curves are too weak to be used in cryptography. The curves secp256k1 and CurveEd25519 are not supersingular since $\#secp256k1(\mathbb{F}_{p_1}) = n_1 \neq p_1 + 1$, and $\#CurveEd25519(\mathbb{F}_{p_2}) = 8n_2 \neq p_2 + 1$. As a consequence, both curves are resistant to the MOV supersingular attack.

3.7 Comparison of the security

The following table 1 resumes the former cryptanalytical study.

Attack	Attack condition	Resistance of secp256k1	Resistance of CurveEd25519
CM field discriminants	$\|D\| > 2^{100}$	$\|D_1\| < 2^2$	$\|D_2\| > 2^{254}$
Pohlig-Hellman	n with small factors	n_1 is prime	n_2 is prime
Pollard's rho	small $\sqrt{\frac{\pi n}{2}}$	$n_1 \geq 2^{255}$ is large	$n_2 \geq 2^{254}$ is large
j-invariant	$j = 0$	$j = 0$	$j \neq 0$
Anomalous	$n = p$	$n_1 \neq p_1$	$n_2 \neq p_2$
Frey-Rück	$n\|(p^k - 1)$ for $k \leq 30$	$n_1 \not\|(p_1^k - 1)$	$n_2 \not\|(p_2^k - 1)$
MOV	$n = p + 1$	$n_1 \neq p_1 + 1$	$n_2 \neq p_2 + 1$

Table 1: Resistance of secp256k1 and CurveEd25519 to cryptanalytical attacks

Table 1 shows that the CurveEd25519 is more resistant than the curve secp256k1 to at least two attacks. As a consequence, the CurveEd25519 can be used for industrial applications, such as in a Bitcoin system.

4 Comparison of the Efficiency of secp256k1 and CurveEd25519

In this section, we give a comparison of the efficiency of the arithmetical operations of the curves secp256k1 and CurveEd25519.

4.1 Efficiency of CurveEd25519

CurveEd25519 is a particular case of the a twisted Edwards curve $E_{a,d}$ defined over the finite field $\mathbb{F}_p$ by the equation

$$E_{a,d}: \ ax^2 + y^2 = 1 + dx^2y^2, \ a \neq d, ad \neq 0.$$

In [21], Hisil et al. presented a technique to perform operations on $E_{a,d}$ based on the representation of a point $P = (x, y)$ by the quadruple $(X : Y : T : Z)$ where $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$, $xy = \frac{T}{Z}$, and $Z \neq 0$. With this notation, the twisted Edwards curve equation transforms into an extended one, namely

$$E_{a,d}^e: \ \left(aX^2 + Y^2\right) Z^2 = Z^4 + dT^2, \ a \neq d, ad \neq 0.$$

The negative of a point $(X : Y : Z : T) \in E_{a,d}^e$ is the point $(-X : Y : -T : Z)$ and the point at infinity $\mathcal{O}$ is represented by $(0 : 1 : 0 : 1)$.

When $a = -1$ as in CurveEd25519, the addition of two distinct points $(X_1 : Y_1 : T_1 : Z_1)$ and $(X_2 : Y_2 : T_2 : Z_2)$ can be performed with the following operations

$A = (Y_1 - X_1) \cdot (Y_2 + X2)$,	$B = (Y_1 + X_1) \cdot (Y_2 - X2)$,	$C = 2Z_1 \cdot T_2$,	$D = 2T_1 \cdot Z_2$,
$E = D + C$,	$F = B - A$,	$G = B + A$,	$H = D - C$,
$X_3 = E \cdot F$,	$Y_3 = G \cdot H$,	$T_3 = E \cdot H$,	$Z_3 = F \cdot G$.

Table 2: Addition in CurveEd25519

The computational cost of the addition on $E^e_{-1,d}$ is then eight multiplications ($8M$), two doublings ($2D$), and eight additions ($8Add$) in the field $\mathbb{F}_p$. This be reduced to $7M + 2D + 7Add$ when $Z_2 = 1$.

Similarly, for $a = -1$ as in CurveEd25519, the doubling of a point $(X_1 : Y_1 : T_1 : Z_1)$ can be performed with the following operations

$A = X_1^2$,	$B = Y_1^2$,	$C = 2Z_1^2$,	$D = -A$,
$E = (X_1 + Y_1)^2 - A - B$,	$G = D + B$,	$F = G - C$,	$H = D - B$,
$X_3 = E \cdot F$,	$Y_3 = G \cdot H$,	$T_3 = E \cdot H$,	$Z_3 = F \cdot G$.

Table 3: Doubling in CurveEd25519

The computational cost of the doubling on $E^e_{-1,d}$ is then four multiplications ($4M$), four squarings ($4S$), one doubling (1D) and six addition ($6Add$) in the field $\mathbb{F}_p$. This can be reduced to $3M + 4S + 1D + 6Add$ by performing a parallel doubling process (see [21], Section 4.4). There are other ways to perform addition and doubling on twisted Edwards curves as shown in [6]. The advantage of the methods presented above do not use the curve parameter d as input.

4.2 Efficiency of secp256k1

The Koblitz curve secp256k1 with the equation $y^2 = x^3 + 7 \pmod{p_1}$ belongs to the family of curves with a short Weierstrass equation of the form $y^2 = x^3 + ax + b$. Any point (x, y) on this curve can be represented by the projective point $(X : Y : Z)$ with $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$ for $Z \neq 0$ and $(0 : 1 : 0)$ for the point at infinity. Then, the Weierstrass equation transforms to the projective one $Y^2Z = X^3 + aXZ^2 + bZ^3$. The addition law in the projective case has many forms. To compute the sum

$$(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3 : Y_3 : Z_3),$$

the following formula for addition has an optimal efficiency (see [7], The "add-1998-cmo-2" addition formulas).

$Z1Z1 = Z_1^2$,	$Z2Z2 = Z_2^2$,	$U1 = X_1 \cdot Z2Z2$,
$U2 = X_2 \cdot Z1Z1$,	$S1 = Y_1 \cdot Z_2 \cdot Z2Z2$,	$S2 = Y_2 \cdot Z_1 \cdot Z1Z1$,
$H = U2 - U1$,	$HH = H^2$,	$HHH = H \cdot HH$,
$r = S2 - S1$,	$V = U1 \cdot HH$,	
$X_3 = r^2 - HHH - 2V$, $Y_3 = r \cdot (V - X3) - S1 \cdot HHH$, $Z_3 = Z1 \cdot Z2 \cdot H$.		

Table 4: Point addition in secp256k1

The computational cost of the point addition on secp256k1 is then twelve multiplications ($12M$), four squarings ($4S$), one doubling (1D) and six additions ($6Add$) in the field $\mathbb{F}_{p_1}$.

To compute the double point $2(X_1 : Y_1 : Z_1) = (X_3 : Y_3 : Z_3)$, the following formula has an optimal efficiency (see [7], The "dbl-1998-cmo-2" doubling formulas).

$XX = X_1^2$,	$YY = Y_1^2$,	$ZZ = Z_1^2$,
$S = 4X_1 \cdot YY$,	$M = 3XX + a \cdot ZZ^2$,	$T = M^2 - 2 \cdot S$,
$X_3 = T$,	$Y_3 = M \cdot (S - T) - 8YY^2$,	$Z_3 = 2Y1 \cdot Z1$.

Table 5: Point doubling in secp256k1

The computational cost of the point doubling on secp256k1 is then three multiplications ($3M$), six squarings ($6S$), eight doubling ($8D$) and five additions ($5Add$) in the field $\mathbb{F}_{p_1}$.

The following table 6 gives the cost of the point addition and point doubling on the curves secp256k1 and CurveEd25519 in terms of the field arithmetic multiplication (M), squaring (S), doubling (D) and addition (Add).

Curve	Addition	Doubling
secp256k1	$12M + 4S + 1D + 6Add$	$3M + 6S + 8D + 5Add$
CurveEd25519	$7M + 2D + 7Add$	$3M + 4S + 1D + 6Add$

Table 6: Arithmetic comparison of secp256k1 and CurveEd25519

In [6] There are various ways to speed up the computation on $\mathbb{F}_{p_2}$, so the arithmetic operations are efficient and speed in this field. For more detail see section 5.2.

Table 6 shows that the operations on CurveEd25519 are faster than the operations on secp256k1. Other forms of elliptic exist with explicit formula for the cost of the addition or doubling of points [8,30]. In all cases, the operation on Edwards curves are the fastest comparing to the other forms. As a consequence, for efficiency reasons, it is more convenient to use the curve CurveEd25519 for industrial applications such as in a Bitcoin system.

5 Comparison of the digital signatures ECDSA and Ed25519

In this section, we show that the digital signature Ed25519 based on the curve CurveEd25519 is more suitable for the Bitcoin system than the digital signature ECDSA which is used in practice.

5.1 The elliptic curves

The elliptic digital signature algorithm ECDSA is based on the Koblitz elliptic curve secp256k1 while the digital signature Ed25519 is based on the twisted Edwards curve CurveEd25519. In the past sections, we have showed that the curve secp256k1 is more vulnerable to Pollard's rho attack while the curve CurveEd25519 is safe. Moreover, as discussed in [2], secp256k1 is more vulnerable to specific attacks based on some of its twists. More vulnerabilities of secp256k1 are listed in [23]. As a consequence, CurveEd25519 is more secure than secp256k1 for industrial applications, especially for Bitcoin.

5.2 The finite fields

ECDSA uses the Koblitz elliptic curve secp256k1 over the finite field $\mathbb{F}_{p_1}$ where $p_1 = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$. The digital signature Ed25519 uses the twisted Edwards curve CurveEd25519 over the field $\mathbb{F}_{p_2}$ where $p_2 = 2^{255} - 19$. There are various ways to speed up the computation on $\mathbb{F}_{p_2}$. In [6], any integer a modulo p_2 is represented in base 2^{51} as

$$a = a_0 + 2^{51}a_1 + 2^{102}a_2 + 2^{153}a_3 + 2^{204}a_4,\ a_i \in \left\{0, \ldots, 2^{51} - 1\right\}.$$

This representation is then performed to process the multiplication and squaring in an efficient way to fit an 128-bit serial multiplier. Moreover, in [6], any integer b modulo p_2 is represented in base $2^{25.5}$ using the sequence $2^{\lceil 25.5i \rceil}$ for $i = 0, \ldots, 9$ as

$$\begin{aligned} b =& b_0 + 2^{26}b_1 + 2^{51}b_2 + 2^{77}b_3 + 2^{102}b_4 + 2^{128}b_5 + 2^{153}b_6 + 2^{179}b_7 + \\ & 2^{204}b_8 + 2^{230}b_9, \quad b_i \in \left\{-2^{25}, \ldots, 2^{25}\right\}. \end{aligned}$$

This representation is efficient in processing the multiplication and squaring on a 64-bit serial multiplier. As a consequence, the arithmetic operations are efficient in the field $\mathbb{F}_{p_2}$. This makes Ed25519 a good candidate for industrial applications, especially for Bitcoin.

5.3 The hash functions

In Bitcoin, the Koblitz curve secp256k1 is combined with the hash function SHA-256 in the ECDSA signature process. In a similar way, the digital signature Ed25519 combines the curve CurveEd25519 with the hash function SHA-512. SHA-256 and SHA-512 are parts of the SHA2 family, standardized in 2001 by the National Institute of Standards and Technology (NIST) [18]. The SHA-2 family will remain deployed in the future even in the presence of SHA3. SHA-256 and SHA-512 are closely related since they use very similar algorithms, based on the same byte operations. They differ only in the input bit lengths and produce outputs of lengths of 256 bits and 512 bits respectively. Nevertheless, SHA-256 and SHA-512 differs at the security level. SHA-512 is more secure than SHA-256 and is recommended by various cryptographic standards such as NIST [27], ENISA [15] and BlueKrypt [1] for use for more sensible data and for longest terms. This is an advantage for the digital signature Ed25519 over the digital signature ECDSA for long terms.

6 Conclusion

We have studied and compared the digital signature ECDSA with the Koblitz elliptic curve secp256k1 and the digital signature Ed25519 based on the twisted Edwards curve CurveEd25519 for use in Bitcoin. Our analysis of the security shows that the curve CurveEd25519 is more secure than secp256k1, especially against Pollard's rho attack on the elliptic discrete logarithm problem. Moreover, our study of the efficiency and implementation shows that Ed25519 is more efficient. We conclude that Ed25519 is more suitable for use in the Bitcoin system, especially for long term applications.

References

1. Bluekrypt. cryptographic key length recommendation.
`https://www.keylength.com/en/`
2. Bos J.W., Halderman J.A., Heninger N., Moore J., Naehrig M., Wustrow E.: Elliptic Curve Cryptography in Practice. In: Christin N., Safavi-Naini R. (eds) Financial Cryptography and Data Security. FC 2014. Lecture Notes in Computer Science, vol 8437. Springer, Berlin, Heidelberg, pp. 157–175 (2014)
`https://cryptome.org/2013/11/ecc-practice.pdf`
3. Bernstein D. J., and Lange T.: Faster addition and doubling on elliptic curves. In K. Kurosawa, editor, ASIACRYPT, volume 4833 of LNCS, pages 29-50. Springer, 2007.
4. Bernstein D. J., Birkner P., Joye M., Lange T., and Peters C.: Twisted Edwards curves. In S. Vaudenay, editor, AFRICACRYPT, volume 5023 of LNCS, pages 389-405. Springer, 2008.
5. Bernstein, D.J.: Curve25519: new Diffie-Hellman speed records, in PKC 2006 proceedings, Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, Tal Malkin Eds. Lecture Notes in Computer Science 3958, Springer, pp. 207–228 (2006)

6. Bernstein, D.J., Duif, N., Lange, T., Schwabe, P., Yang, B.Y.: High-speed high-security signatures, in CHES 2011, 124-142 (2011).
7. Bernstein, D.J., Lange, T.: Explicit-Formulas Database. https://www.hyperelliptic.org/EFD/g1p/auto-shortw-jacobian-0.html#addition-add-2001-b
8. Bernstein D.J., Lange T.: Faster Addition and Doubling on Elliptic Curves. In: Kurosawa K. (eds) Advances in Cryptology - ASIACRYPT 2007. ASIACRYPT 2007. Lecture Notes in Computer Science, vol 4833. Springer, Berlin, Heidelberg, pp. 29–50 (2007) https://eprint.iacr.org/2007/286.pdf
9. Bernstein D.J., Lange T.: SafeCurves: choosing safe curves for ellipticcurve cryptography, version 2017.01.22 (2017) http://safecurves.cr.yp.to/index.html
10. Certicom Research: Standards for efficient cryptography 2: Recommended elliptic curve domain parameters. Standard SEC2, Certicom (2000)
11. Cohen H., Frey G., Avanzi R., Doche C., Lange T., Nguyen K., and Vercauteren F.: Handbook of elliptic and hyperelliptic curve cryptogra- phy. CRC press, 2005.
12. Diffie, W., Hellman, M.E.: New directions in cryptography, IEEE Transactions on Information Theory, Vol. IT–22, 1976, pp. 644–654 (1976)
13. ECC Brainpool. ECC Brainpool standard curves and curve generation, v. 1.0 (2005) // www.ecc-brainpool.org/download/Domain-parameters.pdf
14. Edwards, H. M.: A normal form for elliptic curves. Bulletin of the American Mathematical Society, vol. 44, pp. 393–422, 2007.
15. ENISA: Algorithms, key size and parameters report (2014) https://www.enisa.europa.eu/publications
16. El Gamal, T.: A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Transactions on Information Theory IT-31, 496-473 (1985)
17. FIPS PUB 186-4, Digital Signature Standard (DSS). July 2013 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
18. FIPS PUB 180-4, Secure Hash Standard (SHS) https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.180-4.pdf
19. Frey, G., Rück, H.-G.: A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, Math. Comp., 62, No.206 (1994) pp. 865–874 (1994)
20. Hankerson, D., Vanstone, S., and Menezes, A.: Guide to elliptic curve cryptography. Springer Professional Computing. Springer-Verlag, New York, 2004.
21. Hisil, H., Wong, K.K.-H., Carter, G., Dawson, E.: Twisted Edwards curves revisited. In Advances in Cryptology - ASIACRYPT 2008, vol. 5350 of Lecture Notes in Computer Science, Springer Verlag, pp. 326–343 (2008)
22. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of Computation, 48: pp. 203–209, (1987)
23. Mayer, H.: ECDSA security in Bitcoin and Ethereum: a research survey https://blog.coinfabrik.com/wp-content/uploads/2016/06/ECDSA-Security-in-Bitcoin-and-Ethereum-a-Research-Survey.pdf
24. Menezes, A.J., Okamoto, T., Vanstone, S.A.: Reducing elliptic curve logarithms to logarithms in a finite field, IEEE Trans. Inf. Theory, 39, No. 5 (1993) pp. 1639–1646 (1993)
25. Miller, V.S.: Use of elliptic curves in cryptography. In H. C. Williams, editor, Advances in Cryptology - CRYPTO'85, Vol. 218 of Lecture Notes in Computer Science, Springer-Verlag, pp. 417–426 (1986)

26. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. 24 May 2009. https://bitcoin.org/bitcoin.pdf
27. NIST: Policy on Hash Functions, https://csrc.nist.gov/projects/hash-functions/nist-policy-on-hash-functions
28. Pohlig, S.C., Hellman, M.E.: An improved algorithm for computing logarithms over GF (p) and its cryptographic significance, IEEE Transactions on Information Theory, 24, 1978, pp. 106–110 (1978)
29. Pollard, J.: Monte Carlo methods for index computation mod p, Mathematics of Computation, 32 (1978), pp. 918–924 (1978)
30. Renes J., Costello C., Batina L.: Complete Addition Formulas for Prime Order Elliptic Curves. In: Fischlin M., Coron JS. (eds) Advances in Cryptology - EUROCRYPT 2016. EUROCRYPT 2016. Lecture Notes in Computer Science, vol 9665. Springer, Berlin, Heidelberg (2006) https://www.microsoft.com/en-us/research/wp-content/uploads/2016/06/complete-2.pdf
31. Satoh, T, Araki, K.: Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves, Commentarii Mathematici Universitatis Sancti Pauli, 47 (1998), pp. 81–92 (1998)
32. Schoof, R.: Counting points on elliptic curves over finite fields, Journal de Théorie des Nombres de Bordeaux, 7(1):219-254, (1995)
33. Semaev, I.: Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p, Mathematics of Computation, 67 (1998), pp. 353–356 (1998)
34. Silverman, J.H.: The Arithmetic of Elliptic Curves, Graduate Texts in Mathematics, Springer-Verlag, 106 (1986)
35. Smart, N.P.: The discrete logarithm problem on elliptic curves of trace one, Journal of Cryptology, 12 (1999), pp. 110–125 (1999)