A Digital Signature on Cubic Pell Cryptosystem CP256-1299

Nur Azman Abu[1,*], Abderrahmane Nitaj[2], and Muhammad Rezal Kamel Ariffin[3]

[1]Department of Computer System and Communication, Faculty of ICT, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
[2]Département de Mathématiques, Université de Caen, Caen, France
[3] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Serdang, Malaysia
Email: nura@utem.edu.my (N.A.A.); abderrahmane.nitaj@unicaen.fr (A.N.); rezal@upm.edu.my (M.R.K.A.)
*Corresponding author

Manuscript received August 15, 2024; revised November 23, 2024; accepted February 21, 2025; published May 8, 2025

***Abstract*—Elliptic curves have proven to be a suitable foundation for cryptosystems, with Elliptic Curve Cryptosystems (ECC) offering strong security with smaller key sizes. Recent advancements in ECC design aim to create more efficient and secure curves. In this paper, we introduce a new digital signature scheme, named CP256-1299. It is a 256-bit scheme based on a cubic Pell curve where the arithmetic operations are efficient and straightforward. In previous works, cubic Pell curves have been used to design public key cryptosystems. Our main motivation in proposing the new digital signature algorithm is to exploit the effectiveness of the arithmetic of cubic Pell curves, while maintaining reasonable keys and high security. We compare our new scheme to three widely-used digital signature algorithms based on ECC, namely ED25519, SECP256K1 and SECP256R1. It turns out that our cubic Pell curve based digital signature algorithm is designed to operate with a larger periodic order while maintaining at least similar computational requirements to most popular elliptic curve cryptosystems. Our new scheme is also suitable to support a central bank digital currency.**

***Keywords*—cubic pell equation, elliptic curve cryptosystem, digital signature**

I. Introduction

Rivest-Shamir-Adleman (RSA) is the most widely used public key cryptosystems, with its security relying on the difficulty of factoring its modulus, $N = pq$, the product of two large primes. Traditionally, strong RSA [1] primes were considered highly secure, but the development of the Elliptic Curve Method (ECM) [2, 3] for factorization challenged this assumption. Since the early 2000s, the security of RSA has increasingly depended on the sheer size of its modulus. A need for larger modulus sizes has led many developers to shift from RSA to an Elliptic Curve Cryptosystem (ECC). ECC is favored for its compact key sizes and efficient operations, making it a preferred choice among technical developers. A variety of schemes based on ECC have been proposed in the last two decades, and even more, some of them are considered resistant to quantum computers, which makes them candidates for post quantum cryptography. ECC is employed for encryption such as EC-ElGamal, signatures such as EC-DSA, and key exchange such as EC-DH. Several international agencies recommend to use ECC for present and near future applications. Most ECC based schemes rely on an arithmetic of elliptic curves, and on the hardness of an elliptic curve discrete logarithm problem. In parallel, several attacks have been proposed against ECC based schemes. This development pushes the cryptography community to investigate different arithmetic operations, and different hard problems for ECC security.

Digital signatures have numerous advantages and benefits [4–6]. They are used to reduce bank fraud losses, to verify an authenticity and integrity of digital data, to provide authentication and non repudiation, to ensure secure communications and transactions. They are used in encrypted emails, banking, healthcare, signed contacts and agreements, and government communications.

II. Motivation

This paper proposes a new 256-bit digital signature scheme based on a new family of cubic curves that are different from elliptic curves, but have similar arithmetic and properties. The goals of this new digital signature algorithm are numerous:

1) The first goal is to present a technical support for a Central Bank Digital Currency (CBDC) implementation. We particularly focus on the digital ringgit for Malaysian central bank, namely, Bank Negara Malaysia (BNM).
2) The second goal is to use the arithmetic of a cubic Pell curve, which presents an alternative solution to ECC.
3) The third goal is to show that the proposed scheme has efficient implementation, reasonable keys, and security strength.

It is known that ECC is more secure and more efficient than RSA. Typically, a 256-bit ECC provides a 128-bit security level, whereas RSA requires a modulus of at least 2048 bits to achieve the same level of security. Various types of curves can be employed in ECC, each with unique properties and advantages. Due to an efficiency of ECC and its variants, we have decided to use cubic Pell curves as the underlying arithmetic of our scheme. A cubic Pell curve has previously been proposed as an alternative to RSA cryptosystem [7] and cryptanalized in [8, 9]. Our new 256-bit ECC variant is based on a cubic Pell curve, named CP256-1299 which will double its periodic cycle from traditional ECC 256-bit to 512-bit.

Several well-known elliptic curves are commonly referred to by their nicknames, such as ED25519, SECP256K1, and SECP256R1. A comparison between SECP256K1 and SECP256R1 is presented in [10]. SECP256K1 is a Koblitz curve defined over a finite field of characteristic 2, whereas SECP256R1 is a prime field curve. The "K" in SECP256K1 refers to Koblitz, while the "R" in SECP256R1 stands for random. Before the rise of Bitcoin, SECP256K1 was rarely used but has since gained popularity due to its unique properties. This curve was generated by Certicom [11], while the SECP256R1 curve was created by the National Institute of Standards and Technology (NIST) [12].

Although these curves are part of the standard set, Certicom is known to hold extensive patents on many elliptic curve algorithmic properties. There has been speculation that the National Security Agency (NSA) previously leveraged its

influence over NIST to introduce a backdoor into a random number generator used in elliptic curve cryptography standards.

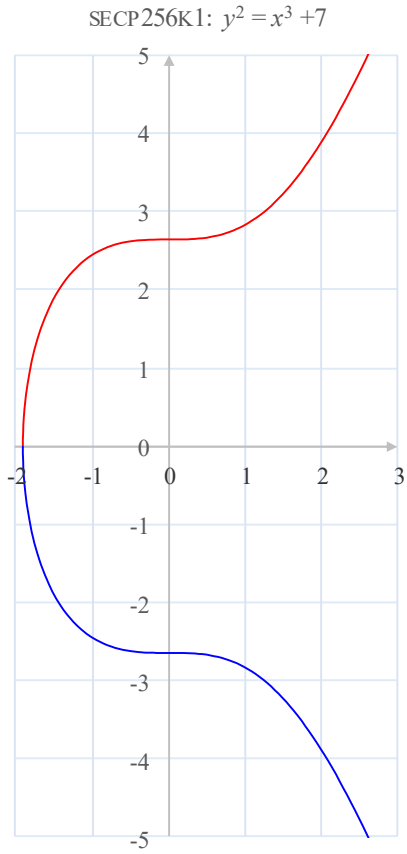

Fig. 1. A visual elliptic curve $y^2 = x^3 + 7$.

III. Literature Review

A. The Elliptic Curve SECP256K1

In designing a cryptosystem, an algebraic curve is chosen for its group computational efficiency without sacrificing its security. For instance, the elliptic curve SECP256K1: $y^2 = x^3 + 7$ with fixed $a = 0$, as visually depicted in Fig. 1, is defined over a finite field $\mathbb{F}_p$ with a generalized Mersenne prime p for faster field arithmetic.

This elliptic curve domain is recommended by the Standards for Efficient Cryptography Group. It is designed for a 256-bit prime. The elliptic curve SECP256K1 is defined over the finite field $\mathbb{F}_p$ with

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$
$$= 115792089237316195423570985008687907853269984665640564039457584007908834671663,$$

a generalized Mersenne prime number for faster field arithmetic. The defining equation of this elliptic curve is E: $y^2 = x^3 + 7$ as shown in Fig. 1 with an order $^{\#}E(\mathbb{F}_p) = \phi$ where

$$\phi = 115792089237316195423570985008687907852837564279074904382605163141518161494337.$$

A base point is G(x, y) =
(55066263022277343669578718895168534326250603453777594175500187360389116729240,
32670510020758816978083085130507043184471273380659243275938904335757337482424).

SECP256K1 is practically preferred to SECP256R1 due to its 30% faster execution time in optimized fashion. SECP256R1 uses a very suspicious seed C49D360886E704936A6678E1139D26B7819F7E90 without an explanation on why this seed is chosen which is strangely analogous to the case of a backdoor in Dual_EC_DRBG [13].

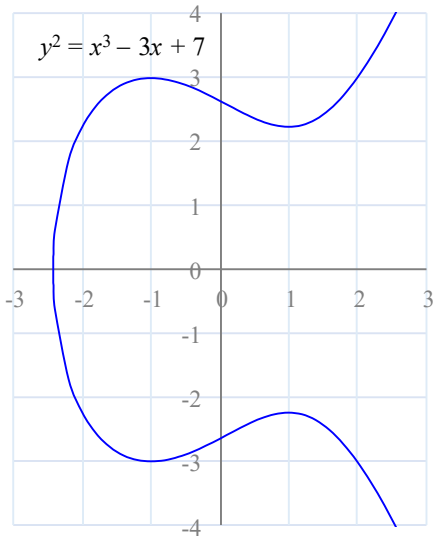

Fig. 2. A sample elliptic curve for $b = 7$.

B. The Elliptic Curve SECP256R1

An elliptic curve SECP256R1: $y^2 = x^3 - 3x + b$ with fixed $a = -3$, as visually depicted in Fig. 2, is defined over a finite field $\mathbb{F}_p$ with a generalized Mersenne prime p for faster field arithmetic. An elliptic curve SECP256R1 is defined over the finite field $\mathbb{F}_p$ with a generalized Mersenne prime p. Their powers are all multiples of 32. These properties give reduction algorithms that are particularly rapid on unsigned long 32-bit processing. $2^{256} \equiv 2^{128} + 2^{64} \pmod{p}$.

$$p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1 =$$
$$115792089210356248762697446949407573530086143415290314195533631308867097853951,$$

a generalized Mersenne prime number for faster field arithmetic operation. The defining equation of this elliptic curve is E: $y^2 = x^3 - 3x + b$ as shown in Fig. 2 with an order

$$^{\#}E(\mathbb{F}_p) = \phi =$$
$$115792089210356248762697446949407573529996955224135760342422259061068512044369.$$

A parameter b =
$$41058363725152142129326129780047268409114441015993725554835256314039467401291.$$

A starting base point is G(x, y) =
(48439561293906451759052585252797914202762949526041747995844080717082404635286,
36134250956749795798585127919587881956611106672985015071877198253568414405109).

An ECC carry a preferably fixed base point G. This base point has an order ϕ. To gain an impression on how this curve might looks like a sample elliptic curve is presented in Fig. 2 for a small integer $b = 7$.

The SECP256R1 curve, recommended by NIST, is considered similarly unsafe as SECP256K1 in several categories. While NIST's P-256 curve passes the CM field discriminant check, it fails the rigidity test because the seeds used for curve generation are not fully explained, raising concerns about potential backdoors. In 2013, rumors suggested that the NSA may have been involved in generating weak curves. A significant concern is the possibility that an unverifiable random base point could provide a shortcut for point projection on the elliptic curve, compromising its security.

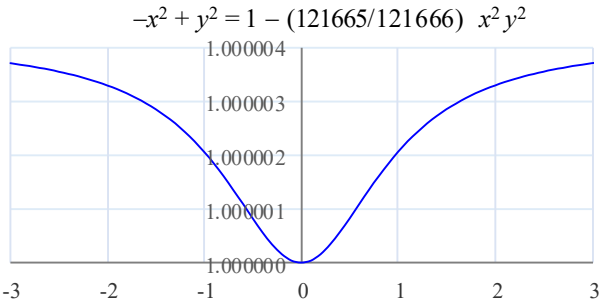

Fig. 3. An upper half of a twisted Edwards curve.

C. The Elliptic Curve ED25519

There has been a substantial recent progress on efficient implementation of elliptic curves. An elliptic curve cryptosystems based on Montgomery-form,

$$E^M: By^2 = x^3 + Ax^2 + x$$

has become more popular than a standard Weierstrass-form,

$$E^W: y^2 = x^3 + ax + b.$$

Montgomery-form elliptic curves have been getting useful for public-key cryptosystems from the point of view of not only efficient implementation but also protection against timing-attacks [14].

Let $p = 2^{255} - 19$, then Curve25519 in $\mathbb{F}_p{}^2$ is an elliptic curve in Montgomery form,

$$y^2 = x^3 + 486662x^2 + x.$$

This curve is birationally equivalent to a twisted Edwards curve ED25519:

$$-x^2 + y^2 = 1 - \frac{121665}{121666}x^2 y^2.$$

The upper half of the twisted Edwards curve ED25519 is depicted in Fig. 3. The transformation is easy: simply define $x = \sqrt{486664}\ \frac{u}{v}$ and $y = \frac{u-1}{u+1}$. Note that 486664 is a square modulo p. An inverse transformation is just as easy: simply define $u = \frac{1+y}{1-y}$ and $v = \sqrt{486664}\ \frac{u}{x}$ [15]. It should be noted that 486664 is a square modulo p and $d = \frac{121665}{121666}$ is not a square modulo p.

A cryptocurrency Monero uses a particular Twisted Edwards elliptic curve for cryptographic operations, ED25519, the birational equivalent of the Montgomery curve Curve25519. Curve ED25519 is not subject to any known patents.

This Twisted Edwards curve has order

$$\#E = n = 2^3 \cdot 7237005577332262213973186563042994240857116359379907606001950938285454250989,$$

where

$$n = 2^{252} + 27742317777372353535851937790883648493.$$

Implementations of ED25519 (such as Monero) typically use the generator G = $(x, \frac{4}{5})$, where x is even, or parity b = 0, variant based on point decompression of $y = \frac{4}{5} \pmod q$.

A starting base point is G(x, y) = (15112221349535400772501151409588531511454012693041857206046113283949847762202, 46316835694926478169428394003475163141307993866256225615783033603165251855960).

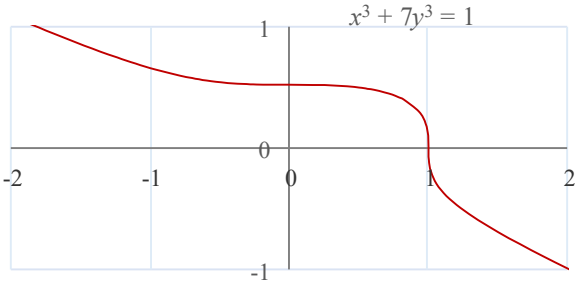

Fig. 4. A visual instance on a cubic Pell curve for $c = 7$ at the level $z = 0$.

III. BACKGROUND OVERVIEW ON A CUBIC PELL EQUATION

In 1659, John Pell and Johann Rahn have written an algebra text on finding infinitely many positive integer solutions to a quadratic equation $u^2 - dv^2 = 1$. In 1909, Axel Thue has shown that a cubic equation $u^3 - dv^3 = 1$ has finitely many integer solutions. A sequence of solution points (u_n, v_n) modulo a prime can be generated without bound as n increases without bound. This basic cubic equation is birationally equivalent to an elliptic curves of the form $y^2 = x^3 - D$ [16].

Let p be a prime number, and c a cubic non-residue in $\mathbb{F}^*_p$. A necessary condition for an existence of a cubic non-residue in $\mathbb{F}^*_p$ is $p \equiv 1 \pmod 3$. For such moduli, a set of cubes is in the form;

$$E_3 = \{g^3, g^6, \ldots, (g^3)^{\frac{p-1}{3}} \equiv 1\} \pmod p;$$

where g is a primitive root of $\mathbb{F}^*_p$. For each $a \in E_3$, the equation $t^3 \equiv a \pmod p$ has three solutions, and no solution if $a \notin E_3$. As a consequence, an integer $c \in \mathbb{F}_p{}^*$ is a cubic residue if and only if $c^{\frac{p-1}{3}} \equiv 1 \pmod p$.

A cubic Pell equation in the finite field $\mathbb{F}_p$ is given by the equation:

$$x^3 + cy^3 + c^2z^3 - 3cxyz \equiv 1 \pmod p.$$

An equivalent elliptic curve is depicted in Fig. 4 at the level $z = 0$. A group of points (x, y, z) can be formed by points which satisfy the cubic Pell equation modulo p. The set C of all solutions of the cubic Pell equation form a finite group with an addition law $\oplus$ with following properties:

i. The sum of two solutions (x_1, y_1, z_1) and (x_2, y_2, z_2) of

the Pell equation is defined by

$$(x_1, y_1, z_1) \oplus (x_2, y_2, z_2) = (x_3, y_3, z_3)$$

where

$$x_3 = cy_1z_2 + cy_2z_1 + x_1x_2,$$
$$y_3 = cz_1z_2 + x_1y_2 + x_2y_1,$$
$$z_3 = x_1z_2 + x_2z_1 + y_1y_2.$$

ii. A neutral identity element is (1, 0, 0).

iii. An inverse of (x, y, z) is $(x^2 - cyz, cz^2 - xy, y^2 - xz)$.

iv. The order of C is $p^2 + p + 1$.

The following result gives an explicit formula for the order of the group C.

Lemma 1. Let p be a prime number such that $\gcd(p - 1, 3) = 3$. Let c be a cubic non-residue in F^*_p. Then

$$^{\#}\{(x, y, z) \in \mathrm{Z}/p\mathrm{Z}^3 : x^3 + cy^3 + c^2z^3 - 3cxyz \equiv 1 \pmod p\} = p^2 + p + 1.$$

Proof. Let p be a prime number such that $\gcd(p - 1, 3) = 3$. Let c be a cubic non-residue in $\mathbb{F}^*_p$. Let θ be a real number such that $\theta^3 = c$. Consider the group

$$\mathrm{C} = \{(x, y, z) \in \mathrm{Z}/p\mathrm{Z}^3: x^3 + cy^3 + c^2z^3 - 3cxyz \equiv 1 \pmod p\}.$$

Also, consider the set

$$\mathrm{G} = \{x + y\theta + z\theta^2: (x, y, z) \in \mathbb{F}^3_p \setminus \{(0, 0, 0)\}\}.$$

If $x_1 + y_1\theta + z_1\theta^2 \in G$ and $x_2 + y_2\theta + z_2\theta^2 \in G$ then an addition operation is given as

$$(x_1 + y_1\theta + z_1\theta^2) \oplus (x_2 + y_2\theta + z_2\theta^2) = x_3 + y_3\theta + z_3\theta^2$$

with

$$x_3 = cy_1z_2 + cy_2z_1 + x_1x_2,$$
$$y_3 = cz_1z_2 + x_1y_2 + x_2y_1,$$
$$z_3 = x_1z_2 + x_2z_1 + y_1y_2.$$

A straight forward calculation shows that $x_1 + y_1\theta + z_1\theta^2 \in G$. Also (1, 0, 0) is an identity element in G, and an inverse of $x + y\theta + z\theta^2 \in G$ is the element $x^2 - cyz + (cz^2 - xy)\,\theta + (y^2 - xz)\,\theta^2 \in G$. This makes G a subgroup of $\mathbb{F}^*_p{}^3$. Since the order of $\mathbb{F}^*_p{}^3$ is $p^3 - 1 = (p - 1)(p^2 + p + 1)$, the order of G is a divisor of $p^3 - 1$. Let $x + y\theta + z\theta^2 \in G$. Then since $a^p \equiv a \pmod p$ for all $a \in \mathbb{F}_p$, we have

$$\begin{aligned}(x + y\theta + z\theta^2)^{p^2+p+1} &= (x + y\theta + z\theta^2)^{p^2} \\ &\quad .(x + y\theta + z\theta^2)^p \\ &\quad .(x + y\theta + z\theta^2).\end{aligned}$$

We use

$$\theta^p = (\theta^3)^{\frac{p-1}{3}}\theta = c^{\frac{p-1}{3}}\theta$$

and

$$\theta^{p^2} = (\theta^3)c^{\frac{(p-1)p}{3}}\theta^p = c^{\frac{p-1}{3}}c^{\frac{p-1}{3}}\theta = c^{\frac{2(p-1)}{3}}\theta.$$

Then

$$\begin{aligned}(x + y\theta + z\theta^2)^{p^2+p+1} &= (x + y\theta + z\theta^2)^{p^2} \\ &\quad .(x + y\theta + z\theta^2)^p \\ &\quad .(x + y\theta + z\theta^2) \\ &= (x + yc^{\frac{2(p-1)}{3}}\theta + zc^{\frac{4(p-1)}{3}}\theta^2) \\ &\quad .(x + yc^{\frac{p-1}{3}}\theta + zc^{\frac{2(p-1)}{3}}\theta^2) \\ &\quad .(x + y\theta + z\theta^2).\end{aligned}$$

Observe that in $\mathbb{F}_p$, we have

$$c^{\frac{4(p-1)}{3}} = c^{\frac{p-1}{3}}c^{p-1} = c^{\frac{p-1}{3}}.$$

Hence,

$$\begin{aligned}(x + y\theta + z\theta^2)^{p^2+p+1} &= (x + yc^{\frac{2(p-1)}{3}}\theta + zc^{\frac{p-1}{3}}\theta^2) \\ &\quad .(x + yc^{\frac{p-1}{3}}\theta + zc^{\frac{2(p-1)}{3}}\theta^2) \\ &\quad .(x + y\theta + z\theta^2).\end{aligned}$$

Using $\theta^3 = c$, $c^{p-1} = 1$, and $1 + c^{\frac{p-1}{3}} + c^{\frac{42(p-1)}{3}} = 0$ in $\mathbb{F}_p$, we get

$$(x + y\theta + z\theta^2)^{p^2+p+1} = x^3 + cy^3 + c^2z^3 - 3cxyz \pmod p.$$

This implies that $x + y\theta + z\theta^2$ is of order $p^2 + p + 1$ if and only if $(x, y, z) \in C$. From this we deduce $^{\#}C = p^2 + p + 1$, and terminates the proof.

IV. A Cubic Pell Discrete Logarithm Problem

An addition law $\oplus$ in C can be used to define a scalar multiplication of a point $P \in C$ by an integer m. This multiplication is processed by adding P to itself m times;

$$m \otimes P = P \oplus P \oplus \ldots \oplus P.$$

The set of multiples of P is denoted as $\langle P \rangle$. To compute $m \otimes P$, various algorithms can be used such as double and add point algorithm [17]. The scalar multiplication gives rise to a Cubic Pell Discrete Logarithm Problem (CPDLP) as follows.

Definition 1. Let P and Q be two points of C such that $Q \in \langle P \rangle$. A difficult mathematical problem, CPDLP is to find an integer λ such that $Q = \lambda \otimes P$.

This intractable problem is a fundamental building block for elliptic curve cryptography. A point multiplication shall be defined as $\mathrm{P}_\lambda = (x_\lambda, y_\lambda, z_\lambda) = \lambda \otimes \mathrm{P}_1 = \lambda \otimes (x_1, y_1, z_1)$. A discrete logarithm problem in cubic Pell field is defined as solving for an integer λ from base point $\mathrm{P}_1 = (x_1, y_1, z_1)$ to a projection point $\mathrm{P}_\lambda(x_\lambda, y_\lambda, z_\lambda)$. CPDLP is presumably safe as intractable as an Elliptic Curve Discrete Logarithm Problem (ECDLP).

An elliptic curve E over a finite field $\mathbb{F}_p$ has an order of the form $^{\#}E = p + 1 - t_p$, where, according to Hasse's Theorem, $0 \le |t_p| \le 2\sqrt{p}$. As consequence, a running time of the generic algorithms to ECDLP is $O(\sqrt{p})$.

V. An Elliptic Curve Cryptosystem Design

A choice of an elliptic curve and its parameter(s) plays an important role in both security and efficiency of ECC. A primary objective is to choose a curve with a reasonably high security level and at reasonably low computational cost. Constructive steps in designing a new ECC are as follows:

i. Select an elliptic curve

ii. Choose a modulus prime
iii. Count on arithmetic operation
iv. Count a periodic cycle of points on the curve.
v. Generate a system parameter and a base point
vi. Select digital signature and verification scheme

Selection of elliptic curves, prime modulus and potential adoption of the system have been listed in Table 1. This section will go through constructive steps in designing a new ECC nicknamed as CP256-1299.

Table 1. A comparison on elliptic curves, prime modulus and adoption system among popular ECCs

ECC	Curve	p	Adoption
SECP256K1	$y^2 = x^3 + 7$	$2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$	Bitcoin Ethereum
SECP256R1	$y^2 = x^3 - 3x + b$	$2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$	Hyperledger Fabric
ED25519	$-x^2 + y^2 = 1 - \frac{121665}{121666} x^2 y^2$	$2^{255} - 2^4 - 2^1 - 1$	Monero
CP256-1299	$x^3 + cy^3 + c^2z^3 - 3cxyz = 1$	$2^{256} - 2^{10} - 2^8 - 2^4 - 2^1 - 1$	Digital Ringgit

A. Selection of an Elliptic Curve CP256-1299

Elliptic curves can be expressed in many forms, and elliptic-curve computations can be carried out in many ways. Two popular options reigned supreme for 50 years of elliptic-curve Elliptic Curve Cryptosystem (ECC).

i. Short Weierstrass curves $y^2 = x^3 + ax + b$, with Jacobian coordinates (X:Y:Z) representing (X/Z^2, Y/Z^3), were the representation of choice for most computations.
ii. Montgomery curves $By^2 = x^3 + Ax^2 + x$, with Montgomery coordinates (X:Z) representing two points (X/Z, ±⋯), were the representation of choice for single-scalar multiplication.

This trend has changed starting from an advent of Edwards curves in 2007. Edwards curves involve significantly fewer multiplications than short Weierstrass curves in Jacobian coordinates, and for sufficiently large scalar, fewer multiplications than Montgomery curves in Montgomery coordinates. Note that larger scalars benefit from larger windows, reducing the number of additions per bit for Edwards coordinates but not for Montgomery coordinates [18].

Selection of an elliptic curve in this paper concentrates on not only its simplicity but also its efficiency. Murru and Saettone, proposed a cubic Pell RSA variant [7]. It is intended to be more secure than RSA in broadcast applications. Working on a cubic field related to a cubic Pell equation, a group field can be constructed to ride on a much larger periodic cycle.

B. Generating an Efficient Strong 256-bit Prime

An RSA cryptosystem is the most popular and extensively analysed public key cryptosystem since its inception in 1978. Recent attack on a partial prime exposure should taken into consideration [19]. An attack on RSA starts on its periodic cycle which divides a prime factor of $p - 1$. One such popular classic attack was via Pollard's $p - 1$ algorithm [20] which initially presented in 1974. Then comes another field arena which carry a periodic cycle $p + 1$. Williams's $p + 1$ integer factorization algorithm via Lucas sequences has been presented [21].

A strong prime must consist of at least two large prime factors on both side of $p - 1$ and $p + 1$ respectively. A periodic cycle in a field modulo p has always been given by a totient function $\phi(p) = p - 1$ back then. ECM is a generalization of Pollard's $p-1$ algorithm. The complexity of Pollard's $p - 1$ algorithm is dominated by the largest prime factor of periodic cycle $p - 1$ over $\mathbb{F}_p$.

Then comes another field arena which carry a periodic cycle $p + 1$. A priority has always been given a slightly larger prime factor on $p - 1$ than on $p + 1$. This right-hand-side of strong prime criterion on $p + 1$ has been hardly taken into serious consideration yet in an elliptic curve cryptography. There is hardly new arena yet to make use of smaller a periodic cycle $p + 1$ in solving an intractable elliptic curve discrete logarithm problem yet. Even though there is no such need for a moment, this first author still think that a precaution should be taken here in designing a new elliptic curve cryptosystem.

It is well-known that to avoid successful relevant attacks against an ECC system, the number of points on the chosen curve, called order of the curve n, must carry at least one very large prime factor. Let us take an overview on the ED25519 prime modulus $p = 2^{255} - 19$. Its largest prime factor on $p - 1$ is a 236-bit prime,

74058212732561358302231226437062788676166966415465897661863160754340907.

This 236-bit prime is admirably large. However, when it comes to its largest prime factors on $p + 1$, there are two 95-bit primes,

31927947500766558008599290859

and

35408198551781170063534027037.

Such periodic cycles are certainly within reach of current computing prowess. An invention of a ring which can attack an elliptic curve discrete logarithm from $p+1$ periodic cycle will have a clear direction to break many such elliptic curve ciphers.

First, let us take an overview on an SECP256K1 prime modulus $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$. Its largest prime factor on $p-1$ is a 237-bit prime, namely,

205115282021455665897114700593932402728804164701536103180137503955397371.

This prime is admirably large. However, when it comes to its largest prime factors on $p+1$ is the 184-bit prime, namely,

21759506893163426790183529804034058295931507131047955271.

This prime is practically large.

Second, let us take an overview on SECP256R1 prime modulus $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$. The largest prime factor of $p - 1$ is a 160-bit prime,

835945042244614951780389953367877943453916927241.

This prime is reasonably large. However, when it comes to its largest prime factors in $p+1$, it is merely 94-bit prime,

113183089279739419314049141 03.

A periodic cycle $p+1$ of this size will give a clear direction on to break this elliptic curve within reach of super computing

power.

A choice of prime p will determine ECC algebraic efficiency in computing modulo p due to its friendly form to CPU word processing. Since the target here is 256-bit ECC, a prime modulus is preferably chosen as to a power of 256 for efficient modular reduction. The best candidate is in a tight form $2^{256} - k$ for some small integer k. An integer $k = 1299$ is chosen as the smallest positive integer for which $2^{256} - k$ satisfies strong prime criteria. It should be noted that $k = 1299 = 2^{10} + 2^8 + 2^4 + 2^1 + 2^0 < 2^{16}$. This prime is also well protected on the left and the right.

$p = 2^{256} - 1299 =$
115792089237316195423570985008687907853269984665640564039457584007913129638637.
$p - 1 =$
$2^2 \cdot 3 \cdot 79 \cdot 116121953 \cdot 4944856253 \cdot 21271717174740349$
$246610256385913263328140464110678736952 3.$
$p + 1 =$
$2 \cdot 107 \cdot 3113857811 \cdot 173766614273427837579699871$
$663288827015055936164654544261019401447.$

On the left, its largest prime factor on $p - 1$ is a 188-bit prime while on the right its largest prime factor on $p + 1$ is a 217-bit prime.

An integer $k = 2063$ is another good candidate for which $2^{256} - k$ satisfies strong prime criteria. It should be noted that $k = 2063 = 2^{11} + 2^3 + 2^2 + 2^1 + 1 < 2^{16}$. Thus,

$p = 2^{256} - 2063 =$
157920892373161954235709850086879078532699846656405640394575840079131296378 73.
$p - 1 =$
$2^4 \cdot 72370055773322622139731865630429942408293$
$740416025352524660990004945706023 67.$
$p + 1 =$
$2 \cdot 3 \cdot 103 \cdot 70030055387 \cdot 72370055773322622139731865$
630429942408293740416025
$3525246609900049457060236 7.$

On the left, the largest prime factor of $p - 1$ is a 252-bit prime while on the right its largest prime factor on $p + 1$ is a 211-bit prime.

A comparison on bit sizes of largest prime factor $p - 1$, on the left and largest prime factor $p + 1$, on the right of prime modulus has been listed in Table 2.

Table 2. A comparison on bit size on largest prime factor from prime modulus p, accompanying largest prime factors $p-1$ on the left and $p+1$ on the right together with point periodic order among popular ECCs

ECC	Bitsize of $(p-1)$	Bitsize of p	Bitsize of $(p+1)$	Bitsize of $^{\#}E$
ED25519	236	255	95	253
SECP256K1	237	256	184	256
SECP256R1	160	256	94	256
CP256-1299	188	256	217	512
EC256-2063	252	256	211	509

There are 2 strong prime candidates in this proposed cryptosystem, namely, $p_1 = 2^{256} - 1299$ and $p_2 = 2^{256} - 2063$. We have $p_1 \equiv 1 \pmod 3$ and have $p_2 \equiv 2 \pmod 3$. An ideal choice on p_2 will have carry the largest 252-bit prime factor on the left and a modestly large 211-bit prime factor on the right. Unfortunately, this prime modulus is not suitable for our system since every element in F_{p^2} is a cube, and so is c.

Moreover, it will ride on $p_2^2 - 1$ periodic cycle within a cubic Pell norm. A more practical choice on $2^{256} - 1299$ will have carry a modest 188-bit prime factor on the left and the practically largest 217-bit prime factor on the right. At the same time, this prime modulus p will ride on $p^2 + p + 1$ periodic cycle within a cubic Pell norm. There are 3 more candidates have been found in this project namely, $2^{256} - 5093$, $2^{256} - 5097$ and $2^{256} - 8939$. Naturally, the smallest $k = 1299$ is easier to gain popularity.

Algorithm 1: Reduction modulo $p = 2^{256} - k$, $k < 2^{16}$.
Input: An integer $M = [M_{15}, ..., M_2, M_1, M_0]$ in base 2^{32} with $0 \le M < p^2$.
Output: $C = M \bmod p$

define an integer array $C = [C_7, ..., C_2, C_1, C_0]$ in base 2^{64}
redefine an integer array $M = [M_{15}, ..., M_2, M_1, M_0]$ in base 2^{64}
for i=0, …, 7 do
$C_i = M_i + k \cdot M_{i+8}$.

Set an initial carry = 0.
for i=0, …, 7 do
$M_i = C_i$ + carry,
$C_i = M_i \bmod 2^{32}$,
carry = $M_i >> 32$.

If carry > 0 then
$M_0 = C_0 + k \cdot$carry,
$C_0 = M_0 \bmod 2^{32}$,
Carry = $M_0 >> 32$.

If carry > 0 then
$M_1 = C_1 + k \cdot$carry,
$C_1 = M_1 \bmod 2^{32}$,
Carry = $M_1 >> 32$.

redefine an integer array $C = [C_7, ..., C_2, C_1, C_0]$ in base 2^{32}
return C.

A prime modulo is typically selected for a finite field to have a very special form facilitating efficient implementation. This particular prime p is chosen for its efficiency on modular operation. Algorithm 1 a pseudocode on a fast modulo reduction by this prime $p = 2^{256} - k$ which cut down a modulo operation into half of modulo reduction by a random prime p. At the same time, this reduction modulo p can be easily adopted for a 32-bit processor down to a 16-bit processor since $k = 1299 = 2^{10} + 2^8 + 2^4 + 2^1 + 2^0 < 2^{16}$.

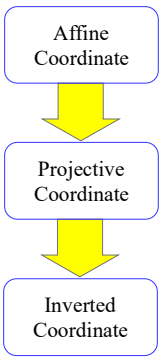

Fig. 5. Searching for an efficient count on arithmetic operations.

C. Count on Arithmetic Operations

Primary computational cost in a cryptosystem on each iteration are an inversion (**I**), multiplication (**M**), squaring (**S**), Greatest Common Divisor (GCD), multiplication by a small

constant (**m**) and addition (**A**) listed in descending order. In searching for an efficient count on arithmetic operations, a transition from Affine coordinates to inverted coordinates is called for here as depicted in Fig. 5.

Traditionally, an inversion cost six times the cost of multiplication on a small arithmetic field. To avoid an inversion, for instance, an efficient implementation has been proposed in [15] on a projective Edwards curve.

Defining two points on an elliptic curve in a standard Weierstrass-form with an equation $y^2 = x^3 + ax + b$, as $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. Then $P_3 = P_1 \oplus P_2 = (x_3, y_3)$ is algebraically given by $x_3 = m^2 - (x_1 + x_2)$ and $y_3 = m(x_1 - x_3) - y_1$ where a slope of a secant line intersecting the two points, $m = \frac{y_2 - y_1}{x_2 - x_1}$ for distinct point addition when $P_1 \neq P_2$, and a slope of tangent line $m = \frac{3x_1^2 + a}{2y_1}$ for point doubling when $P_1 = P_2$.

In this paper, we concentrate on a point addition only for simplicity. This classic addition $P_3 = P_1 \oplus P_2$ requires 1**I**, 2**M**, 1**S** and 6**A**. This computation is considered already efficient. However, there is still an inversion. A prime field inversion is the most expensive operation here.

The elliptic curve ED25519 proposed by D. J. Bernstein in 2006 is the most discussed alternative curve. It is convenient to compare computational cost in this new CP256-1299 against ED25519. The Edwards curve ED25519 is defined over a pseudo-Mersenne prime field $\mathbb{F}_p$ with $p = 2^{255} - 19$ by means of the following general equation:

$$E: ax^2 + y^2 = 1 + d\,x^2y^2$$

where $a = -1$ and $d = -\frac{121665}{121666} \pmod p$. A unified complete addition operation is given in 3 coordinates in [22] as follows.

Affine coordinates: Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points belonging to a Twisted Edwards curve. An addition operation on these 2 points is defined $P_1 + P_2$ as another point $P_3 = (x_3, y_3) = (x_1, y_1) \oplus (x_2, y_2)$ on the curve where

$$x_3 = \frac{x_1 \cdot y_2 + x_2 \cdot y_1}{1 + d \cdot x_1 \cdot x_2 \cdot y_1 \cdot y_2}, y_3 = \frac{y_1 \cdot y_2 - a \cdot x_1 \cdot x_1}{1 - d \cdot x_1 \cdot x_2 \cdot y_1 \cdot y_2}$$

These formulas for addition also apply for point doubling; that is, when $P_1 = P_2$. To subtract a point, invert its coordinates over the y-axis, $(x, y) \rightarrow (-x, y)$ and use a point addition. Recall that 'negative' elements $-x$ of $\mathbb{F}_q$ are really $-x \pmod q$. A point addition requires 2**I**, 5**M**, 2**m** and 4**A**. This computation appears to be compactly efficient. However, there are still two costly inversions.

Projective coordinates: Given a set of projective points (X:Y:Z) where $Z \neq 0$ corresponds to a set of affine points $(x, y) = (X/Z, Y/Z)$, an equation $E: ax^2 + y^2 = 1 + d\,x^2y^2$ will be transformed into

$$E_z: a(X/Z)^2 + (Y/Z)^2 = 1 + d\,(X/Z)^2\,(Y/Z)^2$$

Multiply on both sides by nonzero Z^4:

$$Z^2\,(aX^2 + Y^2) = Z^4 + d\,X^2Y^2$$

A neutral identity element is (0:1:1) and an inverse of (X:Y:Z) is (–X:Y:Z). An addition of point $(X_1:Y_1:Z_1)$ and $(X_2:Y_2:Z_2)$ on the curve E_z is

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

where

$A = Z_1Z_2$,
$B = A^2$,
$C = X_1X_2$,
$D = Y_1Y_2$,
$E = dCD$,
$F = B - E$,
$G = B + E$,
$X_3 = AF[(X_1 + Y_1)(X_2 + Y_2) - (C + D)]$,
$Y_3 = AG(D - aC)$,
$Z_3 = GF$.

A point addition requires 10**M**, 1**S**, 2**m** and 7**A**. In this projective coordinate, an inversion has been avoided at an expense of doubling multiplication.

Inverted coordinates: Given the set of projective points (X:Y:Z) where $XYZ \neq 0$ corresponds to the set of affine points (Z/X, Z/Y). Then take $X = Z/x$ and $Y = Z/y$ from an equation E becomes

$$E_z': Z^2\,(X^2 + Y^2) = dZ^4 + X^2Y^2$$

where

$A = Z_1Z_2$,
$B = dA^2$,
$C = X_1X_2$,
$D = Y_1Y_2$,
$E = CD$,
$F = C - aD$,
$G = (X_1 + Y_1)(X_2 + Y_2) - (C + D)$,
$X_3 = (E + B)F$,
$Y_3 = (E - B)G$,
$Z_3 = AFG$.

Note that the requirement $XYZ \neq 0$ means that we cannot represent an inverted coordinates points (x, y) such that $xy = 0$. There are 4 points satisfying this condition: A neutral identity element (0, 1), the point of order 2 (0, −1) and the points of order 4 (±1, 0). Additions that involve these points must be handled separately. A point addition requires 9**M**, 1**S**, 2**m** and 7**A**. In this inverted coordinate, 1**M** has been saved at an expense of a negligibly small order.

Edwards curves makes an impact on ECC efficiency, not just in multiplication counts but also in real-world software speeds especially in a cryptocurrency ledger. An Edwards addition law is strongly unified where the same formulas can also be used for generic doublings. An identity point (0, 1) is the neutral element of the addition law. A negative of a point (x, y) is indeed just $(-x, y)$ [18].

From the explicit formulas above, one can readily count on the number of arithmetic operation for point addition. They are listed in Table 3 for each coordinate system.

Table 3. Count of arithmetic operations for a point addition

Coordinate\Operation	I	M	S	m	A
Affine	2	5	0	2	4
Projective	0	10	1	2	7
Inverted	0	9	1	2	7
Cubic Pell	0	9	0	2	6

An addition of two point on a cubic Pell curve already

requires 9**M,** 2**m** and 6**A** which is already competitive among operational cost of Edwards curve in 3 coordinate systems. Montgomery endomorphism in a twisted μ_4-normal form isomorphic to twisted Edwards curves and ordinary elliptic curves brings good reduction and efficient arithmetic [23].

D. Counting an Order as the Number of Points on the Cubic Pell Curve

There are 3 cases in periodic CYCLES which satisfy a cubic Pell equation.

$$E_p: x^3 + cy^3 + c^2z^3 - 3cxyz = 1 \pmod p$$

Let p be a prime, a periodic cycle is mostly determined by a prime modulo p. These three CASES have been observed experimentally. A theoretical background on each cases is beyond the scope of this paper.

i. $^{\#}E_p = p - 1$.
ii. $^{\#}E_p = p^2 - 1$.
iii. $^{\#}E_p = p^2 + p + 1$.

Instances on these three cases are numerically given in Table 4–6 respectively.

Table 4. Take $p = 4099$ and $n = p - 1 = 4098 = 1000000000010_2$, then a projection point is on the left $n\otimes(4, 2, 1) = (x_n, y_n, z_n)$ goes back to an identity $(1, 0, 0) = (x_L, y_L, z_L)$

i	b_i	*L*	x_L	y_L	z_L	*R*	x_R	y_R	z_R
12	1	1	4	2	1	2	44	23	12
11	0	2	44	23	12	3	505	264	138
10	0	4	1701	3032	1585	5	1030	2031	1808
9	0	8	2700	2645	952	9	1653	2149	3600
8	0	16	3138	960	3785	17	2579	3819	3802
7	0	32	3056	970	280	33	2439	3754	2017
6	0	64	142	1020	207	65	2408	1714	3010
5	0	128	250	3466	656	129	1654	2560	1608
4	0	256	4024	2809	3166	257	2202	456	1811
3	0	512	517	3156	490	513	2327	692	591
2	0	1024	96	610	2249	1025	3348	1979	2114
1	1	2049	1366	3329	3966	2050	2311	2763	3393
0	0	4098	1	0	0	4099	3883	21	2901

Table 5. Take $p = 4133$ and $n = p^2 - 1 = 17081688 = 1000001001010010101011000_2$, then a projection point is on the left $n\otimes(4, 2, 1) = (x_n, y_n, z_n)$ goes back to an identity $(1, 0, 0)$

i	b_i	*L*	x_L	y_L	z_L	*R*	x_R	y_R	z_R
24	1	1	4	2	1	2	44	23	12
23	0	2	44	23	12	3	505	264	138
22	0	4	1667	3032	1585	5	486	1759	1672
21	0	8	486	3163	3648	9	763	1963	739
20	0	16	3046	1125	2451	17	644	2951	2701
19	0	32	561	3651	4023	33	1463	2557	3290
18	1	65	1605	2132	1789	66	927	3596	626
17	0	130	878	2259	1077	131	1339	1799	1438
16	0	260	3295	1415	1257	261	3486	384	2887
15	1	521	2426	3898	1358	522	2273	1019	3255
14	0	1042	37	2277	2510	1043	1631	1954	2232
13	1	2085	418	2593	81	2086	292	3509	1795
12	0	4170	3037	2520	846	4171	302	1411	3195
11	0	8340	1000	2797	3397	8341	876	3903	3650
10	1	16681	2373	1101	2011	16682	4023	2562	220
9	0	33362	2037	3164	2341	33363	1075	53	1197
8	1	66725	1960	365	1769	66726	2097	1231	1500
7	0	133450	2762	1409	275	133451	4096	686	2547
6	1	266901	3941	518	2124	266902	3663	24	1074
5	0	533802	3357	2984	2397	533803	1746	2365	2381
4	1	1067605	4069	3517	357	1067606	430	4040	132
3	1	2135211	1622	679	3966	2135212	637	658	2312
2	0	4270422	1866	759	1953	4270423	2922	3907	2930
1	0	8540844	2755	1766	3711	8540845	942	1354	466
0	0	17081688	1	0	0	17081689	4	2	1

Table 6. Take $p = 4111$ and $n = p^2 + p + 1 = 16904433 = 1000000011111000011110001_2$, then a projection point is on the left $n\otimes(4, 2, 1) = (x_n, y_n, z_n)$ goes back to an identity $(1, 0, 0)$

i	b_i	*L*	x_L	y_L	z_L	*R*	x_R	y_R	z_R
24	1	1	4	2	1	2	44	23	12
23	0	2	44	23	12	3	505	264	138
22	0	4	1689	3032	1585	5	838	1935	1760
21	0	8	3252	312	2436	9	4075	138	1287
20	0	16	3192	3653	2741	17	2715	3184	907
19	0	32	1039	2704	2231	33	875	3845	3038
18	0	64	2791	4088	1068	65	1289	633	2906
17	0	128	764	3972	1155	129	1809	835	995
16	1	257	3554	3514	3806	258	1656	2585	1140
15	1	515	1159	936	1178	516	3014	1975	3632
14	1	1031	3451	993	1173	1032	178	2641	1907
13	1	2063	3158	3674	2285	2064	453	8	3202
12	1	4127	2060	1117	2715	4128	626	2927	2821
11	0	8254	3921	3245	1825	8255	2284	709	1267
10	0	16508	2314	1094	3013	16509	1542	1318	110
9	0	33016	3207	1736	187	33017	2932	2334	3316
8	0	66032	1320	239	3450	66033	1810	3080	3265
7	1	132065	1212	1526	1614	132066	1127	3382	2498
6	1	264131	2409	2110	1312	264132	1664	1887	3655
5	1	528263	1392	468	1482	528264	815	2697	34
4	1	1056527	3554	3259	1422	1056528	3494	1321	3427
3	0	2113054	2294	4013	995	2113055	1865	2939	1967
2	0	4226108	68	1615	3232	4226109	3382	443	3893
1	0	8452216	3008	168	1526	8452217	1684	926	1226
0	1	16904433	1	0	0	16904434	4	2	1

In order to gain a better gain in general ECC, the third case will be chosen. We will take a small instance on each case on parameter $c = 7$. From an identity $e = P_0(x_0, y_0, z_0) = (1, 0, 0)$ and a base point $P_1(x_1, y_1, z_1) = (4, 2, 1)$, we can compute $P_n(x_n, y_n, z_n)$ via a BALANCED point projection algorithm [13].

An instance on each case is given in Tables 4–6 respectively. This cubic Pell cryptosystem has been designed to ride on a larger order at about the same computing requirement in an elliptic curve cryptosystem.

E. Base Point and Parameter Selection

Take a sequence starting from an identity point $P_0(x_0, y_0, z_0) = (1, 0, 0)$ and an initial BASE point $P_1(x_1, y_1, z_1)$ which satisfies a cubic Pell equation. The smallest sample point $(x, y, z) \in G^3$ on small parameters $c = 2, \ldots, 7$ are listed in Table 7.

Table 7. The smallest point is an ideal base point on a cubic Pell equation

c	*x*	*y*	z
2	1	1	1
3	4	3	2
4	5	3	2
5	41	24	14
6	109	60	33
7	4	2	1

Take $P_2(x_2, y_2, z_2) = P_1(x_1, y_1, z_1) \oplus P_1(x_1, y_1, z_1)$ and $P_{n+1}(x_{n+1}, y_{n+1}, z_{n+1}) = P_n(x_n, y_n, z_n) \oplus P_1(x_1, y_1, z_1)$ for $n = 2, 3, \ldots$ and so on. Since there is an issue on a random base point of SECP256R1, a basepoint shall be prescribed from a fundamental solution of a cubic Pell equation and project it to a $(p+1)$ point. For instance, let $c = 7$ then a fundamental solution is (4, 2, 1). Take a base point as

$(p + 1) \otimes (4, 2, 1) =$
(2,10157275726438095253509741252252377269949
10428035570447108865955669809341414 02,0).

In this Cubic Pell Cryptosystem, such a base point can be generated since a cubic Pell equation gives a luxury of riding

on a larger periodic order of p^2+p+1. A random secret key in this ECC is within $(2, p-2)$. In a DIGITAL signature scheme, a projection is done twice. First, it is done by a private key. Second, it is done by a secret session key. A double projection in this scheme will not go over bound beyond the periodic order p^2+p+1.

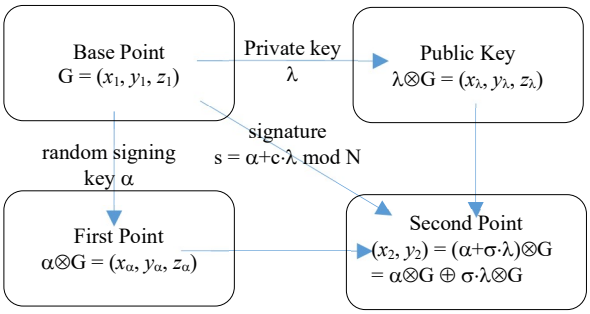

Fig. 6. Point projection in a basic digital signing and verification.

F. Selection on Digital Signature and Verification Scheme

An Elliptic Curve Digital Signature Algorithm (ECDSA) is based on the Digital Signature Algorithm (DSA). SECP256R1 curve is the most popular elliptic curve as part of NIST standards (FIPS 186-4) [24]. In an open public ledger, for example, Bitcoin CAN process 7,000 transactions per second. A modern payment needs to process about 100,000 transactions per second using SECP256R1. However, digital signing and verification using EdDSA on ED25519 is faster and more secure than ECDSA on SECP256R1.

A digital signature scheme with anonymity and spontaneity are typically referred to as a ring of signatures. In the context of digital ringgit, they will ultimately allow for unforgeable, signer-ambiguous transactions that leave currency flows largely untraceable.

System parameters in this cryptosystem initial proposal are as follows,

i. a prime modulo $p = 2^{256} - 1299$,
ii. an identity point $P_0(x_0, y_0, z_0) = (1, 0, 0)$,
iii. a parameter $c = 7$,
iv. a periodic order $^{\#}E(\mathbb{F}_p) = \phi = p^2 + p + 1$,
v. a base point $G = P_1(x_1, y_1, z_1) = (p + 1)\otimes(4, 2, 1)$,
vi. a public key $\lambda G = P_\lambda(x_\lambda, y_\lambda, z_\lambda) = \lambda\otimes(x_1, y_1, z_1)$.

An output pair (α⊗G, s) is expected to be a digital signature on a message *m* from an owner of public key λ⊗G. It should be noted that α is a random 512-bit session number in a traditional digital signature algorithm. A digital signature here consists of an EC point α⊗G, a signature scalar *s* and a public key λ⊗G [25]. They are compactly represented in 32+32+32 bytes. They will be visualized as 3 emblems in the near future.

Digital Signature: Let $G = P_1(x_1, y_1, z_1)$ be a base point generator and λ be a private key. Then take precomputed $\lambda\otimes G = P_\lambda(x_\lambda, y_\lambda, z_\lambda)$ as a public key. Computing a multiple λ of point G is considered as a one-way function. Given both base point G and λ⊗G, it is intractable to extract λ from them.

i. Generate random scalar 512-bit α and compute $\alpha\otimes G = P_\alpha(x_\alpha, y_\alpha, z_\alpha)$
ii. Compute $\sigma = \text{SHA2}(m)$.
iii. Calculate a signature scalar $s \equiv \alpha + \sigma \cdot \lambda \pmod{\phi}$.
iv. Output a signature pair (α⊗G, *s*) of message *m*.

An output pair (α⊗G, s) is expected to be a digital signature on a message *m* from AN owner of public key λ⊗G. It should be noted that α is a random session number in a traditional digital signature algorithm. A digital signature here consists of an EC point α⊗G, a signature scalar *s* from an owner of a public key λ⊗G. They are represented in six 32-bytes. They will be visualized as 6 emblems in future proposal.

Signature Verification: From a signature pair (α⊗G, *s*), public key λ⊗G and a message *m*.

i. Compute $\sigma' = \text{SHA2}(m)$.
ii. Compute $Q = sG = s\otimes(x_1, y_1, z_1)$
iii. $\sigma'\cdot\lambda G = \sigma'\otimes(x_\lambda, y_\lambda, z_\lambda) = (x_{\sigma'\lambda}, y_{\sigma'\lambda}, z_{\sigma'\lambda})$
iv $Q' = \alpha\otimes G \oplus \sigma'\cdot\lambda\otimes G = (x_\alpha, y_\alpha, z_\alpha) \oplus (x_{\sigma'\lambda}, y_{\sigma'\lambda}, z_{\sigma'\lambda})$
v. Check on validation whether $Q = Q'$.

Referring to Fig. 6, there are 2 paths to compute and project from a base point G to a second point $(\alpha + \sigma \cdot \lambda)\otimes G$. First, given a signature scalar $s = \alpha + \sigma \cdot \lambda$ and system parameter base point G, the second point can be computed directly via a point multiplication $s\otimes G$.

Second, given a first point α⊗G as part of a signature, take a public key λ⊗G and message m, then a scalar c can be independently computed as c′ = SHA2(m). Next, $c'\cdot\lambda\otimes G$ will be projected from a public key λ⊗G via a point multiplication. Thus, α⊗G and $c'\cdot\lambda\otimes G$ will be added together to form $\alpha\otimes G \oplus \sigma'\cdot\lambda\otimes G = (\alpha + \sigma'\cdot\lambda)\otimes G$.

In a case of both first and second paths will give the same answer, then the pair (α⊗G, s) is considered a valid signature on a message m from an owner of public key λ⊗G who must have used a private key λ in computing $s = s = \alpha + \sigma\cdot\lambda$ to digitally sign it.

An Example on Digital Signature: Let us take an example. Let us take a sample 256-bit private KEY from the next prime of a 256-bit fraction of an exponential number *e*,

λ=
831713535784724095196510241312745119742990801481105920105552158153065082921 89.

Then take precomputed public key,

λG = $P_\lambda(x_\lambda, y_\lambda, z_\lambda)$ =
(80652997912631420190387978371245749184911663089942002269536582833308066000638,
14781155146764325842105941936507012120419574794976430637883499412459121819899,
10331671645360432302242651223816484639870195054949637384751936796284443968853 1).

Take a 512-bit random session from a 512-bit fraction of a popular number π,

α=
189844710362284492072474648994184972281789985171207447242400075693851369205545798938826247774701606337367572235687533276603126818975945170305282718558031 1.

Compute the first projection point $\alpha G = P_\alpha(x_\alpha, y_\alpha, z_\alpha)$ =
(78193603571961798132195496358874452069777436011327297247055985179143770400755,
58933705053663476286578595285430900140613

95773834186918523937917344743981929 7,
10423227864793148517672585908193216725183
04517206193095559900908664482431929 02).

Take a simple message m= "abc". Compute

σ = SHA2(m) =
84342368487090800366523834928142263660104
88369501651437746298582971681708996 5.

Calculate a signature scalar

$s = \alpha + \sigma \cdot \lambda$ =
89133160547084829443831285993992937680553
85110723954324713910792815497747633992632
61454688442346277736922519859717120787540
02664864475062712658881053636 96.

An Example on Signature Verification: From a signature pair ($\alpha\otimes$G, s), PUBLIC key $\lambda\otimes$G and a message m.

Compute σ' = SHA2(m) =
84342368487090800366523834928142263660104
88369501651437746298582971681708996 5.

Compute point Q = $s\otimes$G = $s\otimes$ (x_1, y_1, z_1) =
(67000248439631917674742898485920369160 22
34446484482686828857947832916285879 62,
84027019278154678998474883614044676081906
68037576348717824558406315685936921 5,
42636317058827573429703393632571482481359
47403961954794084418389595220314442 7).

Compute $\sigma'\otimes(\lambda\otimes$G) = $\sigma'\otimes(x_\lambda, y_\lambda, z_\lambda)$ = ($x_{\sigma'\lambda}$, $y_{\sigma'\lambda}$, $z_{\sigma'\lambda}$) =
(10718871988627107781032047484950894973 09
92340009161311181513724026191950117752,
78189457200157595846947313250786321533443
76999208912517714139612602468831101 8,
72444716359751134594684675486446796463466
87821844277356126872019093300970587 9).

Compute an addition point Q′ = $\alpha\otimes$G $\oplus$ $\sigma'\cdot\lambda\otimes$G
= (x_α, y_α, z_α) $\oplus$ ($x_{\sigma'\lambda}$, $y_{\sigma'\lambda}$, $z_{\sigma'\lambda}$) =
(67000248439631917674742898485920369160 22
34446484482686828857947832916285879 62,
84027019278154678998474883614044676081906
68037576348717824558406315685936921 5,
42636317058827573429703393632571482481359
47403961954794084418389595220314442 7).

Both point Q' and point Q are indeed equal. They are moving towards the SAME second point. Thus, this signature has been verified.

VI. Discussion

A cubic Pell curve has been chosen to generate a new 256-bit ECC. A strong prime has been chosen to the modulo of this cryptosystem Cp256-1299 guarded by two large prime factors on both side of p – 1 and p + 1 respectively. A prime modulo $p = 2^{256} - 1299$ is chosen for its efficiency on modular reduction as proposed in Algorithm 1. An addition of two point on a cubic Pell curve requires 9M, 2m and 6A which brings good reduction and efficient arithmetic. A base point is explicitly prescribed while a cubic Pell equation gives a luxury of riding on a larger periodic order of $p^2 + p + 1$.

VII. Conclusion

Elliptic curve cryptosystems are widely recognized for providing compact support with smaller key sizes compared to other traditional public-key cryptosystems. In this paper, we proposed a new digital signature scheme, called CP256-1299. The new scheme is based on cubic Pell curves, a family of curves that are related to elliptic curves. The new scheme is designed to serve as a technical support for a Central Bank Digital Currency (CBDC) implementation. The new scheme is slightly more efficient. CP256-1299 has been securely designed and its security is comparable to traditional schemes based on elliptic curve cryptography such as ED25519, SECP256K1, and SECP256R1.

Conflict of Interest

The authors declare there is no conflict of interest.

Author Contributions

Nur Azman Abu has contributed to a major apart of this new design on Cubic Pell Cryptosystem CP256-1299. Abderrahmane Nitaj has done an overview and back ground check on a cubic Pell equation especially on the proof of Lemma 1. Muhammad R. Kamel Ariffin has been working on the general comparison works on Cubic Pell Cryptosystem against 3 major elliptic curves cryptosystem. All authors have approved the final version.

References

[1] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[2] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203–209, 1987.

[3] V. S. Miller, "Use of elliptic curves in cryptography," *Lecture Notes in Computer Sciences*, vol. 218, pp. 417–426, 1986.

[4] National Institute of Standards and Technology (NIST), (2023) "Digital Signature Standard (DSS)," *Federal Information Processing Standards Publication (FIPS)*, NIST FIPS 186-5, Department of Commerce, Washington, D.C., 2023.

[5] P. K. Varshney, A. Kukreja, and S. Dewan, "Digital signatures," *IITM Journal of Management and IT*, vol. 11, pp. 86–90, 2020.

[6] M. Pooja and M. Yadav, "Digital signature," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, vol. 3, no. 6, pp. 71–75, 2018.

[7] N. Murru and F. M. Saettone, "A novel RSA-like cryptosystem based on a generalization of the Rédei rational functions," in *Number-Theoretic Methods in Cryptology. NuTMiC 2017. Lecture Notes in Computer Science*, vol. 10737. Springer, Cham, 2018.

[8] A. Nitaj, M. R. Kamel Ariffin, N. N. H. Adenan and N. A. Abu, "Classical attacks on a variant of the RSA cryptosystem," in *Proc. 7th International Conference on Cryptology and Information Security in Latin America (LatinCrypt 2021)*, Progress in Cryptology, Springer, Cham, 6–8 October 2021, pp. 151–167.

[9] N. N. H. Adenan, A. Nitaj, M. R. K. Ariffin, and N. A. Abu, "Cryptanalysis of a cubic Pell variant of RSA with primes sharing least significant bits," *Journal of Information & Optimization Sciences*, vol. 45, no. 5, pp. 1263–1280, 2024.

[10] A. Houria, B. M. Abdelkader, and G. Abderezzak, "A comparison between the SECP256R1 and the koblitz SECP256K1 Bitcoin curves," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 3, pp. 910–918, 2019.

[11] CERTICOM, "SEC 2: Recommended elliptic curve domain parameters, version 2.0," *Certicom Research*, January 27, 2010.

[12] S. Nakamoto. (31 October 2008). Bitcoin: A peer-to-peer electronic cash. [online]. Available: https://bitcoin.org/bitcoin.pdf

[13] M. Wertheimer, "Encryption and the NSA role in international standards," *Notices of the AMS*, vol. 62, no. 2, pp.165–167, February 2015.

[14] K. Okeya, H. Kurumatani, and K. Sakurai, "Elliptic curves with the Montgomery-form and their cryptographic applications," *PKC 2000*, *Lecture Notes in Computer Science (LNCS),* vol. 1751, pp. 238–257, 2000.

[15] D. J. Berstein and T. Lange, "Faster addition and doubling on elliptic curves," in *Proc. Advances in Cryptology—ASIACRYPT 2007: 13th International Conference on the Theory and Application of Cryptology and Information Security*, vol. 4833, 2007, pp. 029–050.

[16] J. A. Cunningham, N. Ho, K. Lostritto, J. A. Middleton, and N. T. Thomas, "On large rational solutions of cubic thue equations: What Thue did to Pell," *Rose-Hulman Undergraduate Mathematics Journal*, vol. 7, no. 2, pp. 1–21, 2006.

[17] N. A. Abu and A. H. Abd Ghafar, "A secure cryptographic algorithm against side channel attacks," *International Journal of Cryptology Research*, vol. 5, no. 2, pp. 045–055, 2015.

[18] K. M. Alonso and S. Noether, *Zero to Monero: A Technical Guide to a Private Digital Currency; for Beginners, Amateurs, and Experts*, 2nd ed. April 4, 2020.

[19] M. Rahmani, A. Nitaj, and M. Ziane, "Partial exposure attacks on a new RSA variant," *Cryptography*, vol. 8, p. 44, 6 October 2024.

[20] J. M. Pollard, "Theorems of factorization and primality testing," *Proceedings of the Cambridge Philosophical Society*, vol. 76, no. 3, pp. 521–528, 1974.

[21] H. C. Williams, "A p+1 method of factoring," *Mathematics of Computation*, vol. 39, no. 159, pp. 225–234, July 1982.

[22] Y. El Housni, "Edwards curves," HAL-01942759, p. 22. 2018.

[23] D. Kohel, "Twisted μ_4-normal form for elliptic curves," in *Proc. EUROCRYPT 2017 Proceedings of 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 10211, 2017, pp. 659–678.

[24] National Institute of Standards and Technology (NIST), "Digital Signature Standard (DSS)," *Federal Information Processing Standards Publication (FIPS)*, FIPS PUB 186-4, p. 91, 2023.

[25] Y. Borissov and M. Markov, "Efficient approach to point-counting on elliptic curves from a prominent family over the prime field $\mathbb{F}_p$," *Mathematics*, vol. 9, no. 12, pp. 1–9, 2021.