Indonesian Journal of Electrical Engineering and Computer Science
Vol. 13, No. 3, March 2019, pp. 910~918
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v13.i3.pp910-918

A comparison between the secp256r1 and the koblitz secp256k1 bitcoin curves

Azine Houria[1], Bencherif Mohamed Abdelkader[2], Guessoum Abderezzak[3]
[1]Institute of Aeronautics and Space Studies, Laboratory of Aeronautical Sciences, Blida1 University Algeria, Algeria
[2]College of Computer and Information Sciences, Center of Smart Robotics ResearchKing, Saud University, Arab Saudi
[3]Department of Electronics, LATSI Signal Processing and Imaging Laboratory, Blida1 University Algeria, Algeria

Article Info

Article history:

Received Sep 18, 2018
Revised Nov 23, 2018
Accepted Dec 11, 2018

Keywords:

Bitcoin
ECC
Mining
Secp256k1
Secp256r1

ABSTRACT

Bitcoin uses elliptic curve cryptography for its keys and signatures, but the specific secp256k1 curve used is rather unusual. The ECDSA keys used to generate Bitcoin addresses and sign transactions are derived from some specific parameters. Due to this characteristic, several questions come up concerning Satoshi's choice of this curve rather than that of the NIST standard secp256r1 curve. Former President Dan Brown's address to Bitcoin users on the Bitcoin talk.org online forum concerning the use of secp256k1 in Bitcoin of SECG showed his surprise to see someone uses SECG secp256k1 instead of secp256r1 of NIST. In this article, we will analyze the random secp256r1 curve and the Koblitz Secp256k1 curve (parameters, equation, automorphism…), by giving the strengths and weaknesses of each one of them, in order to justify the choice of Bitcoin's creator, and then we will tackle the mining using the new graphic cards.

Corresponding Author:

Azine Houria,
Institute of Aeronautics and Space Studies.
Laboratory of Aeronautical Sciences.
Blida1 University Algeria , Algeria
Email: azinehou@yahoo.fr

1. INTRODUCTIO

Elliptical Curve Cryptography (ECC) introduced by Neal Koblitz and Victor Miller allows the achievement of asymmetric cryptography and faster signature than in RSA for a similar level of security [1], [2]. In addition, compared to RSA, ECC allows the computation of pairings that currently allows building of new cryptographic protocols, which can be an advantage for some applications.

Two organisms are known to patent most of the elliptic curve algorithmic properties, namely NIST [3] and Certicom [4].They both propose the use of Weierstrass-based curves that utilize a, b and p parameters.The tuning choices of these parameters remain in many studies a complete secret.

namely the NIST [3] and Certicom [4]. They both propose to use Weierstrass-based curves that use the Secp256r1 and secp256k1 Curves are two examples of two elliptic curves used in various cryptographic protocols such as TLS, SSH, ECDSA, ECDHE, ECDH and ECDLP.

In fact the calculations on the elliptic curves, are governed by some special mathematical group law operations (addition of points in a Finite field) particularly greedy in terms of modular operations of addition, multiplication and inversion. The cost of the operations depends on the elliptic scalar multiplication operation. The implementation of elliptic curve ciphers requires a fine architectural study and design, in order to find the best compromise between complexity and speed computation.

The two major properties for the data communication are Confidentiality and Secrecy. Therefore, the security of the curves relies on several mathematical criteria, which are currently mainly shared by the cryptography community. The main tension, around the selection of the curves to be normalized, is running

on the evaluation of the advantages and disadvantages of each curve (the equation, choice of curve parameters, performance and resistance to attacks by auxiliary channels, simplicity of implementation, efficiency, rigidity, back doors and safety).

2. CURVES SECP256R1/NIST P-256 OVER THE FINITE FIELDS

The most used elliptic curves are those proposed by the NIST on (p) introduced in FIPS [5]. They use special numbers. The curve parameters must be carefully chosen to avoid using a weak curve, and that can withstand all known attacks. There may also be other constraints for security or implementation reasons. Following SEC 2 [6], the domain parameters of the elliptic on Fp are a six-fold T = (p, a, b, G, n, h). Domain parameters as shown in Table 1.

Table 1. Domain Parameters

P	The order of the prime field Fp
Seed	The seed selected to randomly generate, the coefficients of the elliptic curve. The 160-bit SEED input seed to the SHA-1 based on algorithm (the seed parameter domain)
r	The output of SHA-1
a,b	The coefficients of the elliptic curve y2 = x3+ax +b satisfying r b2 ≡ a3 (mod p).
n	the (prime) order of the base point P.
h	The cofacteur
x,y	The x and y coordinates of P.

2.1. Mathematical approach

2.1.1 The prime number p

The p of the P-256 curve is a prime number of generalized Mersien. It is recommended to work on a field whose size is 256 bits. This prime number has the property that it can be written as the sum or difference of asmall number of powers of 2:

The powers appearing in this expression are all multiples of 32. These properties give reduction algorithms that are particularly rapid on machines with wordize of 32 [7]. This optimization is particularly efficient on CPU.Let t = 232 then (1) becomes:

$$p256 = 2^{256}-2^{224} + 2^{192} + 2^{96}-1 \quad (1)$$

We can then reduce the powers higher than 2 by using the congruence for (2) so the congruence relation is:

$$P = t^8-t^7 + t^6 + t^3-1 \quad (2)$$

$$t^4 \equiv t^2 + t \pmod p,\ 2^{256} \equiv 2^{128} +2^{64} \pmod p \quad (3)$$

This P-256 prime number is chosen for efficiency (modular multiplication can be performed more efficiently than in general).Algorithm 2.1 shows the fast reduction by p256. Rapid reduction modulo p256 as shown in Figure 1.

Algorithm [7] :Rapid reduction modulo $p_{256} = 2^{256} -2^{224} +2^{192} +2^{96} -1$

INPUT: An integer $c = (c_{15},..., c_2, c_1, c_0)$ in base 2^{32} with $0 \le c < p^2{}_{256}$.
OUTPUT: c mod p_{256}.

1. Define 256-bit integers:
   $s_1 = (c_7, c_6, c_5, c_4, c_3, c_2, c_1, c_0)$,
   $s_2 = (c_{15}, c_{14}, c_{13}, c_{12}, c_{11},0,0,0)$,
   $s_3 = (0, c_{15}, c_{14}, c_{13}, c_{12},0,0,0)$,
   $s_4 = (c_{15}, c_{14},0,0,0, c_{10}, c_9, c_8)$,
   $s_5 = (c_8, c_{13}, c_{15}, c_{14}, c_{13}, c_{11}, c_{10}, c_9)$,
   $s_6 = (c_{10}, c_8,0,0,0, c_{13}, c_{12}, c_{11})$,
   $s_7 = (c_{11}, c_9,0,0, c_{15}, c_{14}, c_{13}, c_{12})$,
   $s_8 = (c_{12},0, c_{10}, c_9, c_8, c_{15}, c_{14}, c_{13})$,
   $s_9 = (c_{13},0, c_{11}, c_{10}, c_9,0, c_{15}, c_{14})$.
2. 2. Return (s1 +2s2 +2s3 +s4 +s5 −s6 −s7 −s8 −s9 mod p256)

Figure 1. Rapid reduction modulo p256

2.1.2 Elliptic curve Equation

The elliptic curve is isomorphic to a curve with a reduced Weierstrass equation of the form ((p)):

$$y^2 = x^3 + a \cdot x + \text{b mod } p \text{ if } p \neq 2. \quad (4)$$

a) The Discriminant and J-invariant

$$\Delta = 4a^3 + 27b \text{ and } j(E) = (-48a)^3/\Delta \text{ [8]} \quad (5)$$

1) if $\Delta = 0$ then equation (4) is not an elliptic curve, it is a singular cubic.
2) If $\Delta < 0$ then the graph of the elliptic curve has only one component. The cubic polynomial $x^3 + ax + b$ has a single root that corresponds to the abscissa of the intersection point of the curve with the abscissa axis.
3) If $\Delta > 0$ then the graph of the elliptic curve has two components. The cubic polynomial x^3+ax+b has 3 roots, which correspond to the abscissa of the three points of intersection of the curve with the abscissa axis. J-invariant $\neq 0$ and K is a field of characteristic $\neq 2, 3$ then the order of the automorphism is equal to 2.

b) Complexity

In general, the group of points of an elliptical curve behaves like a "Generic group", the discrete logarithm has an exponential complexity [9]. The group of regular points is then isomorphic to an additive or multiplicative group, and the discrete logarithm is sub-exponential, even polynomial. It is imperative that $\Delta \neq 0$ (what happens with $P \approx 1$). More precisely, the complexity of a discrete logarithm is dominated by $\sqrt{q}$, where q is the utmost prime divisor of the number of points of the curve so to increase the complexity it is necessary to have a number of points (almost) first. There are generic attacks of complexity $O(\sqrt{q})$, where q is the utmost prime divisor of N. A safe curve must therefore have $q \approx N$; ideally, $q = N$.

The probability that a random curve has a primary order is approximately the same as a random number of the size of p is prime, $P \approx 1/\log p$ [9]. Complexity of generic attacks as shown in Table 2 and Figure 2.

Table 2. Complexity of Generic Attacks

Method	Fastest known attack the fastest known attack
RSA	Number Field Sieve exp(1/2(logN)1/3(loglogN)2/3)
ECC	Pollard-rho $\sqrt{r} = \exp(1/2 \log r)$

Algorithm 2 Point Doubling ($y^2 = x^3 - 3x + b$, Jacobian coordinates)
INPUT: P = (X1 : Y1 : Z1) in Jacobian coordinates on E/K : $y^2 = x^3 - 3x + b$.
OUTPUT: 2P = (X3 : Y3 : Z3) in Jacobian coordinates.
1. If P = ∞ then return (∞).
2. $T_1 \leftarrow Z^2_1$.{$T_1 \leftarrow Z^2_1$ }
3. $T_2 \leftarrow X_1 - T_1$. {$T_2 \leftarrow X_1 - Z^2_1$ }
4. $T_1 \leftarrow X_1 + T_1$. {$T_1 \leftarrow X_1 + Z^2_1$ }
5. $T_2 \leftarrow T_2 \cdot T_1$. {$T_2 \leftarrow X^2_1 - Z^4_1$}
6. $T_2 \leftarrow 3T_2$. {$T_2 \leftarrow A = 3(X1 - Z^2_1)(X_1 + Z^2_1)$}
7. $Y_3 \leftarrow 2Y_1$. {$Y_3 \leftarrow B = 2Y1$}
8. $Z_3 \leftarrow Y_3 \cdot Z_1$. {$Z_3 \leftarrow B\ Z1$}
9. $Y_3 \leftarrow Y^2_3$.{$Y_3 \leftarrow C = B^2$}
10. $T_3 \leftarrow Y_3 \cdot X_1$.{$T_3 \leftarrow D = C\ X_1$}
11. $Y_3 \leftarrow Y^2_3$. {$Y_3 \leftarrow C^2$}
12. $Y_3 \leftarrow Y_3/2$. {$Y_3 \leftarrow C^2/2$}
13. $X_3 \leftarrow T^2_2$. {$X_3 \leftarrow A^2$}
14. $T_1 \leftarrow 2T_3$. {$T_1 \leftarrow 2D$}
15. $X_3 \leftarrow X_3 - T_1$. {$X_3 \leftarrow A^2 - 2D$}
16. $T_1 \leftarrow T_3 - X_3$. {$T_1 \leftarrow D - X_3$}
17. $T_1 \leftarrow T_1 \cdot T_2$. {$T_1 \leftarrow (D - X_3)A$}
18. $Y_3 \leftarrow T_1 - Y_3$. {$Y_3 \leftarrow (D - X_3)A - C^2/2$}
19. Return($X_3 : Y_3 : Z_3$).

Figure 2. Point doubling

c) Selection of the parameter a = -3

Most standards see the IEEE 1363-2000 standard [10], choose a = -3 because practically all curves have low order isogenies and this for reasons of efficiency so this choice does not affect safety. Choosing small values for a and b parameters makes it possible to accelerate the arithmetic of the curve. Similarly, Brainpool [11] uses this equation for its advantages. This choice saves 2 of the 10 multiplications required for adding points. A random curve on Fp is isomorphic with a curve a = -3 with probability: P = 1/4 if $p \equiv +1 \pmod 4$ and P = 1/2 if $p \equiv -1 \pmod 4$. And finally "a" the selection = -3 for the coefficient in the elliptic curve equation has been made so that the points of the elliptic curve represented in the jacobian projective coordinates could be added using a field multiplication of less. The Figure 2 describes the Point Doubling .

The order of the elliptic curves used in cryptography must respect some constraints in order to avoid known attacks. For example, this order must be a prime number of large size or the Product of a prime number and a small integer or cofactor, which is 1 in the case of a prime order curve.

d) Cofactor

NIST takes the cofactor as small as possible for efficiency reasons:

$$h = \frac{\text{card}\,(E(F))}{n} \tag{6}$$

With h the cofactor = the order of the elliptic curve/n; with n order of the point which is the smallest integer such that (n.G) = 0 (0: element identity of the finite group) and G must be chosen so that n is a large integer.

So some standards cryptographic, such as FIPS-186-4 [5], advocate the use of curves with a "small" cofactor h. In practice, the constraints may differ from one standard to another.

For example, the first version of SEC1 (2000) imposed a cofactor $h \leq 4$ whereas the first version of 2009 recommends rather $h \leq 2\alpha$ and α for a higher level of security.

The choice of the cofactor value depends therefore on its value because:

$$\begin{cases} \text{si } h \leq 1 \text{ For effciency reasons} \\ \text{si } h > \text{Improve performances} \end{cases}$$

Citing as examples the Montgomery curves used by Apple which have a cofactor h > 4 and that to improve the performance of the curve. The Table 3 summarizes the forms of elliptic curves on Fp usable according to the cofactor.

Table 3. Forms of Elliptic Curves on Fp Usable According to the Cofactor [11]

Cofactor h	Form
1	Weierstrass
2	Extended Jacobi Quartic form
3	Generalized Hessien
4	Jacobi Quartic form or Edwards form

e) Parameter b

For the parameter b of the P-256 curve, the following formula is used to generate it:

$$b = \sqrt{\left(-\frac{27}{\text{SHA1}(s)}\right)} \tag{7}$$

With:s=c49d360886e704936a6678e1139d26b7819f7e90 [12].

This procedure generates random data by feeding the seed into SH1 [13]. Verifiable random parameters offer additional conservative characteristics [1]. These parameters are selected from a seed using SHA-1 as specified in ANSI X9.62 [14]. This process ensures that the parameters cannot be redetermined. It is so extremely improbablethat the parameters will be susceptible to future special-purpose attacks and no traps could be placed in the parameters during their generation.

2.2. Algebric approach

a) Group law: for E/K: $y^2 = x^3 + ax + b$, char (K) ≠ 2,3
b) Identity: $P + \infty = \infty + P = P$ for all $P \in E$ (K).
c) Negative: If P = (x, y) ∈ E (K), then (x, y) + (x, -y) = ∞. The point (x, -y) is denoted by-P and is called the negative of P; note that -P is indeed a point in E (K). Also,$-\infty = +\infty$
d) Addition: Let P = (x1, y1) ∈ E (K) and Q = (x2, y2) ∈ E (K), where P = ± Q.
Then P + Q = (x3, y3), where:

$$x3 = \left(\frac{y2-y1}{x2-x1}\right)^2 - 2x1 - x2 \text{ and } y3 = \left(\frac{y2-y1}{x2-x1}\right)^2 - (x1 - x3) - y1 \quad (8)$$

e) Point Doubling Let P = (x1, y1) ∈ E (K), where P =-P.
Then 2P = (x3, y3), where:

$$x3 = \left(\frac{3x^2+a}{2y1}\right)^2 - 2x1 \quad \text{and} \quad y3 = \left(\frac{3x^2+a}{2y1}\right)^2 - (x1 - x2) - y1 \quad (9)$$

The cofactor is always h = 1.

2.3. Selection of the parameters of the curve

The selection of the curve is conditioned by the following parameters. The p and n integersare given in decimal form; bit strings and field elements are given in hexadecimal.

y2=x3-3x + 41058363725152142129326129780047268409114441015993725554835256314039467401291

Table 4. NIST-Recommended Random Elliptic Curves Over Prime Fields [6]

Parameters	Value
p	2^ 256-2 ^ 224 + 2 ^ 192 + 2 ^ 96-1 ffffffff00000001000000000000000000000000ffffffffffffffffffffffff
b	5ac635d8 aa3a93e7 b3ebbd55 769886bc 651d06b0 cc53b0f6 3bce3c3e 27d2604b
n	ffffffff 00000000 ffffffffffffffff bce6faad a7179e84 f3b9cac2fc632551
Seed	c49d3608 86e70493 6a6678e1 139d26b7 819f7e90
c	7efba166 2985be94 03cb055c 75d4f7e0 ce8d84a9 c5114abc af317768 0104fa0d
Gx	6b17d1f2 e12c4247 f8bce6e5 63a440f2 77037d81 2deb33a0 f4a13945 d898c296
Gy	4fe342e2 fe1a7f9b 8ee7eb4a 7c0f9e16 2bce3357 6b315ece cbb64068 37bf51f5

3. CURVES OF KOBLITZ SECP256K1

Secp256k1 refers to ECDSA parameters of the curve used in Bitcoin and is defined in Standards for Efficiency Cryptography (SEC) [6]. Secp256k1 has almost never been used before Bitcoin became popular, but it is gaining popularity due to its many properties. This has been generated by Certicom (a Canadian company) and not by the NIST like the Secp256r1 curve.

3.1. Mathematical Approach

3.1.1 The prime number p

The Weierstrass coefficients defining (a, b) of the curve are (0, 7). SEC2 [6] states in Section 2.1 that the recommended parameters associated with a Koblitz curve have been selected by repeatedly selecting parameters that admit an effectively calculable endomorphism until a first order curve has been found.

The size of the field defining p seems to be a 256-bit boot of the special form :

p = 2 ^ 256-s where s is small with the form s = 2 ^ 32 + t, where t <210 , and t = 29+ 28+ 27+ 26+ 24+ 1.

So P is the seed number used in secp256k1, Bitcoin uses it as the high limit for valid private keys. If a private key is randomly generated larger than n, it is rejected and a new key is regenerated. The probability of such occurrence is low because P is "almost" as large as 2256-1 (256 bits all set to 1). Algorithm of the random generator of private key as shown in Figure 3.

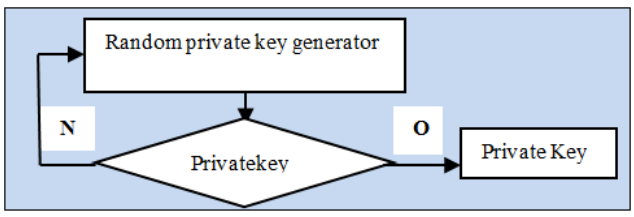

Figure 3. Algorithm of the random generator of private key

3.1.2 Elliptic curve equation

It has a first order of 256 bits. Interestingly, this choice deviates from those made in FIPS 186-4 in that the coefficients of the curve are a = 0 and b = 7.

The elliptic curve is isomorphic to a curve with a reduced Weierstrass equation of the form ((p)):

$$y^2 = x^3 + \text{b mod } p \text{ if } \neq 2.3 \quad (10)$$

As a constant is zero, the term ax of the equation of the curve is always zero, hence the equation of the curve becomes $y^2 = x^3 + 7$.

a) The Discriminant and J-invariant

$\Delta = 4a^3 + 27b^2 \neq 0$ and $j(E) = (-48a)^3/\ \Delta = 0$ because a = 0

This means that secp256k1 has j-invariant 0 so this curve is said to be super-singular and therefore has a very special structure and calculable endomorphism that can be used to accelerate implementations, for example by using the GLV decomposition for scalar multiplication [15]. This idea was introduced by Gallant, Lambert and Vanstone (GLV).Elliptic curves having efficiently-computableendomorphisms should be regarded as "special" elliptic curves. Using "special" instances of cryptographic schemes is sometimes done for efficiency reasons [15].

b) Complexity

This could lead to a more serious attack on secp256k1 because an attacker could get scalar multiples with one-point scalars on any curve on Fp with coefficient a = 0, that is, on the one of the twists of secp256k1.

3.2. Algebraic Approach

3.2.1 Automorphism

Elliptic curves with effectively calculable endomorphismsare considered as "special" elliptic curves, with a small coefficient. But efficient endomorphisms accelerate scalar multiplication, but also Pollard's rho algorithm] for calculating logarithms discreet. For this special class of curves, the acceleration can reach up to 50% compared to the best general methods of point multiplication [16]. If J-invariant = 0 and K is a field of characteristic ≠ 2, 3 then the order of the automorphism is equal to 6.

3.2.2 Fast Scalar Multiplication "GLV decomposition"

There are two methods for accelerating the computation of the scalar multiplication Q = kP on elliptic curves having a non-trivial character effectively calculable endomorphism that are:

a) The Solinas method [15]: This method could only be applied for an elliptic curve defined on binary fields, the endomorphism considered to be theFrobenius.
b) The Gallant-Lambert-Vanstone (GLV) method [15]: its method of elliptic curves defined is applied on primary fields Fp, the decomposition is the basis of the computation acceleration.

Another consequence of the larger automorphism group is the existence of six twists (including the curve itself and the standard quadratic twist). The automorphism group of E has the order 6 and is generated by the map ψ. The curve secp256k1: ≡1 (mod6), there exists a 6th primitive root of the unit ∈Fp, ζ∈$\mathbf{F}p$, and a corresponding automorphism of curve such that ζ6 = 1 [15].

$$\Psi: \rightarrow E, (x, y) \rightarrow (\zeta x, -y) \quad (11)$$

Fast scalar multiplication $\psi P = \lambda P$ for an integer λ6≡1 (mod n). The main advantage of these curves is that dot multiplication algorithms can be designed that does not use dot dubbing.

3.2.3 Selection of parameters of the special Koblitz curve

The elliptic curve parameters of the domain on Fp associated with a Koblitz curve Secp256k1 are defined by the sixfold T = (p, a, b, G, n, h) where the finite field Fp. Parameter of Secp256k1as shown in Table 5 [6].

The curve of Secp256k1 is in the form: E: y2 = x3 + 7 mod p on Fp

Table 5. Parameter of Secp256k1[6]

Parameters	Value
p	2^ 256-2 ^ 32 -2 ^ 9 -2 ^ 8-2 ^ 7 -2 ^ 6 -2 ^ 4-1 fffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f
a	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
b	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007
G	04 79be667e f9dcbbac 55a06295 ce870b07 029bfcdb 2dce28d9 59f2815b 16f81798 483ada77 26a3c465 5da4fbfc 0e1108a8 fd17b448 a6855419 9c47d08f fb10d4b8
n	fffffffffffffffffffffffffffffffe baaedce6 af48a03b bfd25e8c d0364141
h	1

The parameters a, b and p must correctlybe chosen in order to resist the mathematical attacks.

4. COMPARISON OF SECP256R1 AND SECP256K1 CURVES

After studying the two curves, we mention the main differences in Table 5. The SafeCurves website [17] presents security assessments of various. The comparison between secp256r1 and secp256k1 as shown in Table 6.

Table 6. The comparison between secp256r1 and secp256k1

curve	Secp256r1	Secp256k1
security	$2\sqrt{\frac{\pi nsec256r1}{4}} = 127.83$	$2\sqrt{\frac{\pi nsec256k1}{2}} = 127.03$
Automorphism Order	2	6
Parameters “a”	3 are claims of effectiveness, not safety claims	a = 0 the term ax of the equation of the curve is always zero
Cost for a combine attak[17]	2 ^ 120,3	2 ^ 109,5

Koblitz curves are generally known to be a few bits weaker than first-order field curves, but when it comes to 256-bit curves, it has little impact. Bitcoin works with a fixed curve and generates only private and public keys, according to Safecurves [17] the elliptic curve secp256k1 can be considered somewhat "rigid" which means that almost all parameters are transparent to the public and can therefore be supposed not generated to be weak. The rho method breaks the ECDLP using on average additions of about 0.886 √l so the safety is comparable for both curves.

The cost for a combined attack is almost the same for both curves. Certainly the Secp256k1 curve has comparable security as the curve but it has additional twists [16], which lead to more possibilities for an attack. On the other hand, an elliptic curve with j-invariant different from 0 and 1728 as the case of the curve secp256r1 only has a group automorphism of order 2, so that the acceleration of the Pollard rho algorithm [16] is a constant factor up to √3 on such a curve.

Secp256k1 is often more than 30% faster than the other curves if the implementation is sufficiently optimized and the criterion of speed is a very important criterion for the Bitcoins because a payment with Bitcoin is almost instantaneous. However, secp256r1 uses the very suspicious seed "c49d360886e704936a6 678e1139d26b7819f7e90" which is strangely similar to the backdoor in Dual_EC_DRBG [18].

The elliptic curve Bitcoin has the lowest |D| of all known standardized elliptic curves, and therefore is potentially less secure.

5. THE MINING OF BITCOIN

The minors of the Bitcoin protocol use special software and hardware to solve the problem of discrete logarithm or hash functions (Hash256). Hash rates are an important factor that miners must use to determine profits. Several parameters are taken into consideration during the mining, such as the difficulty, the rate of hashing, the cost of electricity and of course, without forgetting the complexity, the slowness and the cost of the equipment, the renewal of the equipment which quickly becomes obsolete and the heat released by the Bitcoins mining equipment tends to easily overheat, which can interrupt its operation.

To overcome all these parameters, miners work in pools to reduce the cost of mining by pooling the computing power of their computers and increase their block resolution capacity.

Mining with a processor (CPU) was the only way to mine bitcoins. Graphics cards (GPUs) eventually replaced CPUs because of their nature, which allowed an increase between 50x to 100x [18] in the computing power in using less electricity per megahash compared to a CPU. The mining world has evolved

into the use of Field Programmable Gate Arrays (FPGAs) as a mining platform. Although FPGAs did not offer a 50x to 100x increase in computing speed as the transition from CPU to GPU [19], they offered better energy efficiency. The world of bitcoin mining is now migrating to the Application Specific Integrated Circuit (ASIC). The rigidity of an ASIC allows it to offer an increase in computing power of 100x [19] while reducing power consumption compared to all other technologies.

We find that the mining power is high and becomes higher, thanks to the development of new mining equipment. The required number of zeros at the beginning of a hash is changed twice a week to adjust the difficulty of creating a block and more zeros means more difficulty. The Bitcoin protocol adds these zeros to maintain the speed at which blocks are added to a new block every 10 minutes. The idea is to compensate for the mining equipment becoming more and more powerful. When the hash is harder, more calculations are needed to create a block and thus more effort to gain new bitcoins, which are then added to the traffic. Transition of Mining Technology as shown in Figure 4 and Comparison of computing power as shown in Figure 5.

Figure 4.Transition of Mining Technology

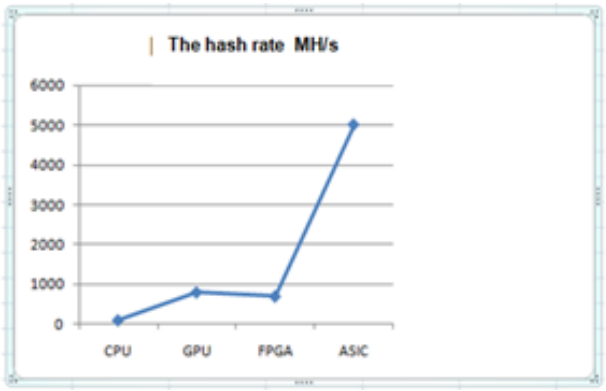

Figure 5.Comparison of computing power

6. CONCLUSION

Constant time calculations help prevent information leaks on the secret key by measuring how long it takes to create the signature. As a result, besides simplicity and efficiency, Secp256k1 could leak information for side channel attacks because the time for some calculations is not constant.

With the new boost that Bitcoin cryptocurrency is having, the research community will turn its attention to two aspects, the cryptography behind the Bicoin and the possible attacks.The major problem is the disambiguity of the possible backdoor, except for mathematical indication, and the various choices in the parameters are not clear or are not completely specified. SafeCurves argues that attackers could have manipulated the choice of standard curves to be vulnerable to a secret attack that applies to a small fraction of curves. The mathematics behind Bitcoin and ECC are based on the solution of very difficult problems of discrete logarithmic problems, that is to say, it is a computationally complex problem. With the introduction and advancement of graphics processing units and cloud computations, NIST standards and other organizations need to be updated.

The new era of computing and the speed of new GPUs that can affect the cryptoanalysis market might be a serious problem for Bitcoin ciphering, especially if it represents the new possiblecurrency.

REFERENCES

[1] https://bitcointalk.org/index.php, 18 septembre 2013
[2] SECG “the Standards for Efficient Cryptography Group”, 1998.
[3] Draft NIST Special Publication 800-57, Recommendations for Key-Management, 2012.

[4] https://www.certicom.com/
[5] NIST, FIPS Publication 186-4, Digital Signature Standard (DSS), 2000 and change notice 1, 2001.
[6] SEC2"Standards for Efficient Cryptography Group": Recommended Elliptic curve Domain Parameters. Version 1.0, 2000.
[7] Marie-Angela CORNELIE « Implantations et protections de mécanismes cryptographiques logiciels et matériels », Doctor Of The Community University Grenoble Alps, 2016.
[8] Younsung Choi "Cryptanalysis on Privacy-aware Two-factor Authentication Protocol for Wireless Sensor Networks "TELKOMNIKA" Vol. 8, No. 1, February 2018, pp. 605-610.
[9] Jean-Pierre Flori, Jérôme Plût, Jean-René Reinhard, Martin Ekerå « Diversité et transparence : choix des courbes elliptiques » NSSI/SDE/ST/LCR, 2015.
[10] IEEE 1363-2000 "IEEE Standard Specifications for Public-Key Cryptography" Sponsor Microprocessor and Microcomputer Standards Committee of the IEEE Computer Society Approved 30 January 2000 IEEE-SA Standards Board.
[11] https://safecurves.cr.yp.to/refs.html #2005/brainpool.
[12] Rogel L.Quilala, Ariel M. Sison, Ruji P. Medina "Modified SHA-1 Algorithm" TELKOMNIKA (Indonesian Journal of Electrical Engineering and Computer Science) Vol. 11, No., pp. 1027-1034, 2018.
[13] ANSI "AMERICAN NATIONAL STANDARD" X9.62-1998.
[14] R. P. Gallant, R. J. Lambert, and S. A. Vanstone." Faster point multiplication on elliptic curves with efficient endomorphisms" In J. Kilian, editor, CRYPTO, volume 2139 of LNCS, pages 190-200. Springer, 2001.
[15] http://safecurves.cr.yp.to/index.html.
[16] https://slashdot.org/story/13/09/11/1224252/are-the-nist-standard-elliptic-curves-backDoored.
[17] https://spectrum.ieee.org/energy/policy/ the-ridiculous-amount-of-energy-it-takes-to-run-bitcoin.