Proceedings of the 1st Faculty of Industrial Technology International Congress
International Conference
Bandung, Indonesia, October 9-11, 2017
ISBN 978-602-53531-8-5

Realization and Key Analysis on Blockchain Bitcoin

Muhammad Lazuardi Wirananda*, Surya Michrandi Nasution*, Marisa W. Paryasto*
Department of Computer System, Telkom University, Bandung - INDONESIA
* Corresponding author e-mail: lazuardiwp24@gmail.com, michrandy@telkomuniversity.ac.id, marisa.paryasto@telkomuniversity.ac.id

Abstract

Bitcoin is the digital currency which implements the Blockchain system an open financial accounting, The security bitcoin used key, where this key is a security and identity of Bitcoin owner. The key is use implements asymmetric cryptography digital signature scheme ECDSA (Elliptic Curve Digital Signature Algorithm) with elliptic curve point multiplication, where for the curve uses Secp256K1. Asymetric cryptogrphy generate two key ie private key and public key. Private key is a point in the curve and the public key is a coordinate (x, y) that represents the private key within the curve. This research implements key on Bitcoin system where by going through all step, it will generate private key and public key corresponding to key in Bitcoin system.

Keywords: Private Key, Public Key, Elliptic Curve Digital Signature Algorithm, Secp256K1,-

1. Introduction

In today's technological era, many sectors are changing because it is influenced by computer technology. Even the financial sector began to change because of the impact of technological developments, where banking activities have been using computerization, today the use of paper money has begun to be shifted by the use of digital money. For example, Bitcoin is currently the most widely used digital currency, in which Bitcoin is designed with the Blockchain system ie recording all transaction activities that exist on Bitcoin open to the public so it can be seen by anyone.

How the working Bitcoin system and how is security from Bitcoin, because all record transaction in spread to public. In the Bitcoin system is divided into three parts: key, address, and transaction. Key is use for security Bitcoin uses an asymetric key system, this key is also the identity of the owner, because it consists of private key and public key where each key has a different function and important in the system Bitcoin.

2. Methodology

2.1. Bitcoin

Bitcoin is an electronic money made in 2009 by Satoshi Nakamoto. Bitcoin uses a database that is distributed and spreads to the nodes of a P2P network using transaction journals, and uses cryptography to provide basic security functions, such as the bitcoin requirement can only be spent by its owner, and can never be done more than once. The design of Bitcoin allows for unauthorized (anonymous) ownership and transfer of wealth. Bitcoin can be sent via the Internet to people who have a Bitcoin address.

- Coin Bitcoin
  Bitcoin coins are defined as an electronic coin with a digital signature chain. Each owner can transfer the coins to their next owner by way of digital signature on the hash of the previous transaction and the public key of the next owner and add this to the end of the coin. The receiver verifies the signature to verify the ownership chain.
- Key Bitcoin
  The key on Blockchain Bitcoin is divided into two parts, namely the public key and private key, where the two keys are generated from the same process but different designations. Where public key will be used for the process of creating blockchain address to receive and send Bitcoin, and private key is used as digital signature in transactions.
  The use of different addresses has a positive effect, which only processes with a new public key and private key, at no cost, where the use of the same address will increase the attack

2.2 Cryptography

Cryptography (cryptography) comes from the Greek cryptós (secret) and gráphein (Writing). So, cryptography means secret writing. Cryptography is the science and art to maintain the confidentiality of the message by encoding it into a form that can no longer be understood meaning. There are four purposes of this cryptography which is also an aspect of information security that is confidentiality (keeps the content of information from anyone except who has authority), data integrity (custody of unauthorized data changes), authentication (identification and introduction of sender and recipient), and non -repudiation (prevents denial of sending any information by the sender).

Criticism is divided into two parts: Symmetric key cryptography where to perform encryption and decry using one key the same example of a caesar chiper. And another is Key-Asymmetric cryptography where to do the encryption and decryption done by different key for example ECDSA

- ECDSA
  Security key is very important because with this key someone can control, use, transfer our data. where to secure it is used Elliptic Curve Digital Signature Algorithms (ECDSA) is an asymmetric algorithm which will generate private key and public key, which algorithm is good for making random keys because using mathematical curve.
  Asymmetric Encryption is used to create keys, in order to get private key and public key. Key has a function as a digital signature. Because ECDSA is an amalgamation of Elliptic Curve Cryptography algorithm with digital signature algorithm. Where it makes this algorithm more secure than DSA algorithm.

In making the public key using Elliptic curve point multiplication process, where it exists

1. Point Addition

$$xr = \lambda^2 - xp\ xq$$

$$yr = \lambda\ (xp - xr) - yp$$

$$\lambda\ = \frac{Yq - Yp}{Xq - Xp}$$

2. Point doubling

$$x = \lambda^2 - 2xp$$

$$y = \lambda\ (\mathrm{xp} - \mathrm{xr}) - \mathrm{Yp}$$

$$\lambda\ = \frac{3Xp^2 + a}{2Yp} \qquad (1)$$

2.3. Secp256K1

Secp256K1 is an elliptic curve domain parameter used as one of the standard curves, that can be used in the ECDSA algorithm scheme where this curve has the information T = (p, a, b, G, n, h) where for the curve shape is defined By equation $y^2 = x^2 + ax + b$.And the attributes of this kurba are defined by:p = Private keya & b = Values a and b on the curveG = Generation Pointn n = Range in curve

- p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F= 2256 -232 -29 -28 -27 -26 -24 –1
- a = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
- b = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007
  Base point for G That have been compressed
- G = 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798Base point for G That have been before compressed
- G = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8
- n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141
- h = 01

3. System Design

Bitcoin hashavestandard to facilitate the user formaking key, address and transasksi, formaking key bitcoin use asymetric algorithm,so that will generate two key is private key and public key.

To generate a public key corresponding to the standard bitcoin is used:

- Algorithm used is ECDSA where the algorithm is asymetric algorithm so it will generate private key and public key, to generate public key useelliptic curve point multiplication for private key representation in curve.
- The curve used is Secp256K1 where this curve becomes a standard bitcoin and can not be replaced by another curve to produce a public key corresponding to the standard bitcoin.

In this section is design flowchart used specialized on key topics. This study begins by applying the creation of a key, starting with the creation of a private key and then entering into the creation of a public key in which both are designed with standard issued bitcoin.

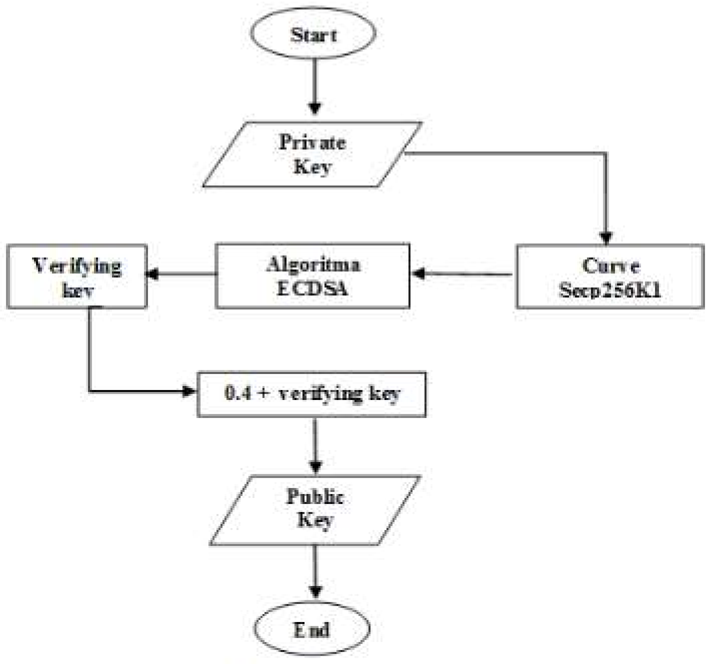


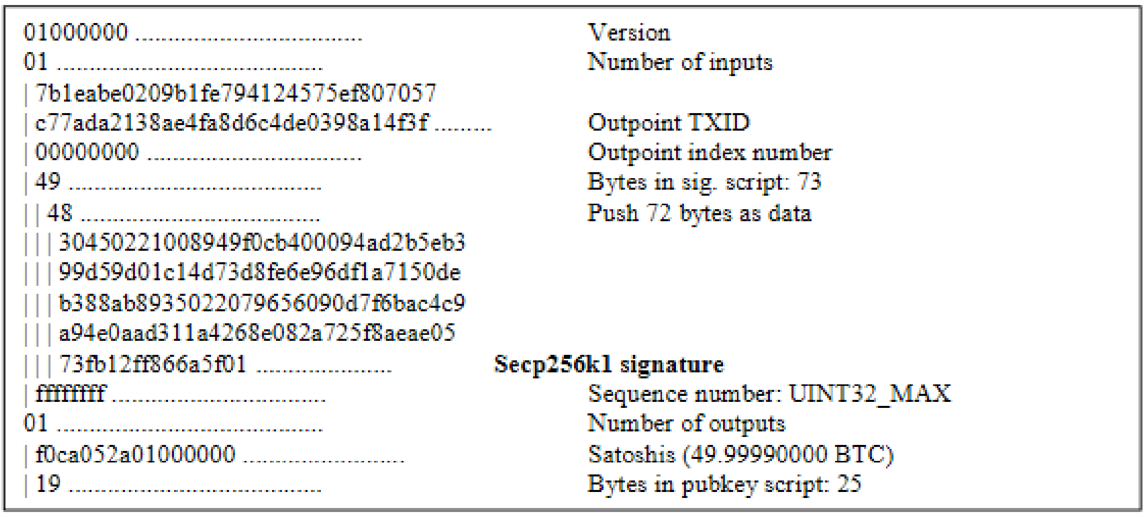
Fig. 1: Figure flowchart used specialized on key topics

Explanation:

- Private key is user input in the form of hex data.
- Private key is plotted in the Secp256k1 curve.
- Private key will be processed in ecdsa algorithm.
- From the plot process on the curve, will be generated in the form of veifying key (private key * Generation key).
- Veifying key will be added 04 at the beginning of the data, 04 itself is Bitcoin provision for public key initialization.
- The above process results into a public key.

4. Implementation and Analysis

The key on bitcoin is as a safeguard as well as the identity of the secretive bitcoin owner, in bitcoin transactions providing the raw for transfer of bitcoin.



```
01000000 ....................................           Version
01 ..........................................           Number of inputs
| 7b1eabe0209b1fe794124575ef807057
| c77ada2138ae4fa8d6c4de0398a14f3f .........            Outpoint TXID
| 00000000 ..................................           Outpoint index number
| 49 ........................................           Bytes in sig. script: 73
| | 48 ......................................           Push 72 bytes as data
| | | 30450221008949f0cb400094ad2b5eb3
| | | 99d59d01c14d73d8fe6e96df1a7150de
| | | b388ab8935022079656090d7f6bac4c9
| | | a94e0aad311a4268e082a725f8aeae05
| | | 73fb12ff866a5f01 .....................     Secp256k1 signature
| ffffffff ..................................           Sequence number: UINT32_MAX
01 ..........................................           Number of outputs
| f0ca052a01000000 ..........................           Satoshis (49.99990000 BTC)
| 19 ........................................           Bytes in pubkey script: 25
```

Fig. 2: Figure raw of transaction

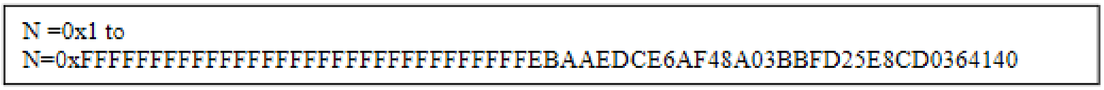
Where the data is sent to the blockchain network to spread widely, it is seen that in the transaction process there is the use of key, both private key when making transaction where the private key will be converted into public key that will be sent entered into the transaction raw and disseminated. From this it appears that the importance of knowing the key used in bitcoin, to know the security of this key

4.1 Implementation Key

Key in bitcoin is divided into two parts: private key and public key this is because the bitcoin uses the concept of asymmetric cryptography algorithm, where to get the key through step:

4.1.1 Private Key

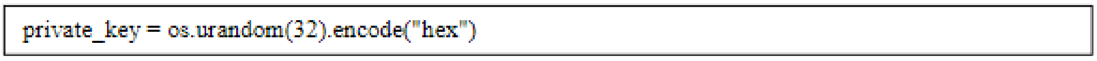
The private key consists of 256-bit hexadecimal characters and must meet the criteria of the Secpt256K1 curve, this is done because the private key is a point within the curve, so that the point does not cross the range within the curve and to get the private key corresponding to the standard bitcoin. A range of usable ranges that match the range of the Secpt256K1 curve:



```
N =0x1 to
N=0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364140
```

Fig. 3: Figure range private key

In this research used 2 ways to generate private key that is by way of user input and generate with software, to use generated with software can use command:



```
private_key = os.urandom(32).encode("hex")
```

Fig. 4: Figure python command line for make random hex character

Implementation private key results

Table 1: Table use of manual input and software

Use	Private Key
By User	11abab5610ae3cc6b820a2c4e6f830aabbccddee40fed0198cba50221b3d5fac
Software	77acefbc10aadef45ad20abcdef1230abe34dc400febd65ae5022bb44dd60ab3

4.1.2 Public Key

The public key is the coordinate (x, y) of the private key into the Secp256K1 curve, where this curve has the equation: $y^2 = x^3 + ax + b$and the parameters T = (p , a, b, G, n):

Information :

p = Private key

a = Curve

b = Curve

G = Generation Point

n = Range curve

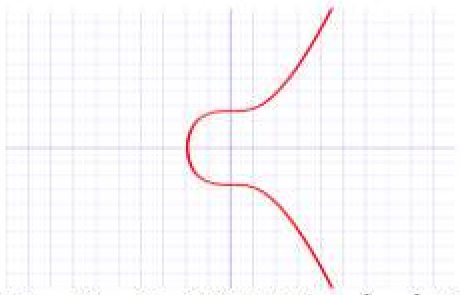
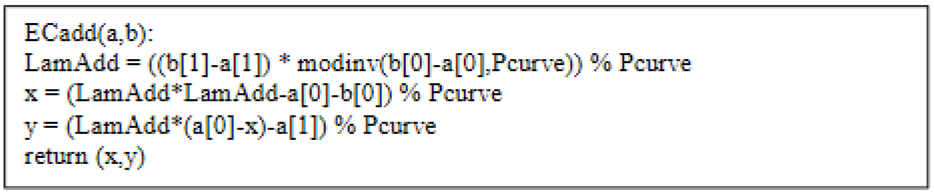


Fig. 5: Figure Curve Secp256K1 in the form$y^2 = x^3 + 0x + 7$

Public key in ECDSA is the result of multiplying private key with generation point: Public key = p * G (verifyng key)To get public key,private key we have to plot into curve by way of Elliptic curvepoint multiplication.

1. Point Addition

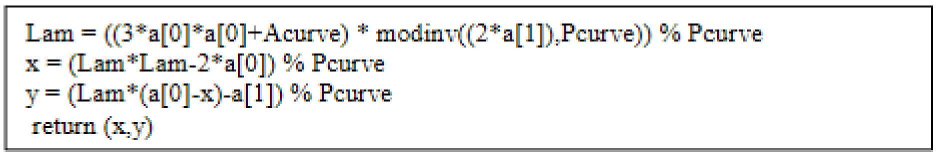


```
ECadd(a,b):
LamAdd = ((b[1]-a[1]) * modinv(b[0]-a[0],Pcurve)) % Pcurve
x = (LamAdd*LamAdd-a[0]-b[0]) % Pcurve
y = (LamAdd*(a[0]-x)-a[1]) % Pcurve
return (x,y)
```

Fig. 6: Figure Command point addition in python

Information :
Lam Add = λ
b [0] = Yq or Gy
a [0] = Xq or Gx
b [1] = b
a [1] = a
Pcurve = Private key
The initial step is to determine the value of LamAdd, after obtaining the value from LamAdd input the value to formula x and y to produce value (x,y)

2. Point Doubling



```
Lam = ((3*a[0]*a[0]+Acurve) * modinv((2*a[1]),Pcurve)) % Pcurve
x = (Lam*Lam-2*a[0]) % Pcurve
y = (Lam*(a[0]-x)-a[1]) % Pcurve
 return (x,y)
```

Fig. 7: Figure Command point doubling in python

Information :
a [0] = Gx
a [1] = Gy
Acurve = a
Pcurve = private key
lam = λ
The initial step is to determine the value of lam (λ), after obtaining th evalue from lam input the value to formula x and y to produce value (x,y).

to display results

```
EccMultiply(GenPoint,ScalarHex): #Double & add
   if ScalarHex == 0 or ScalarHex >= N: raise Exception("Invalid Scalar/Private Key")
   ScalarBin = str(bin(ScalarHex))[2:]
   Q=GenPoint
   for i in range (1, len(ScalarBin)):
      Q=ECdouble(Q); print "DUB", Q[0]; print
      if ScalarBin[i] == "1":
         Q=ECadd(Q,GenPoint); print "ADD", Q[0]; print
   return (Q)
```

Fig. 8: Figure Command show the result Elliptic curve point multiplication in python

Information:
ScalarHex = private key
Gen point = Generation Point

1. The first step is to ensure the private key is in accordance with the provisions of the curve where it should not be 0 and should not exceed the value of N
2. Change ScalarHex = private key to binary

3.If dcalarbin is in range 1 then do ecdouble, then print "Dub", the value of ecdouble
4.If scalarBin == 1 then do ecadd then print "ADD", the result ec add.

```
the private key:
174E1E968188807E3A2E01F6F949641776E6E2FE50BBD59FC61B1E200FC4CF80
the private key (binary) :
10541220698491683960407389552802654973678021542018299799729945808138678882176
the verifyng key :
(36452638215417836447039143289354016844852272259204932072974517859215438473420L,
20712139261975913708573257814401233016007852862210270405001214969154851181194L)
```

Fig. 9: Figure result Elliptic curve point multiplication

From the process of Elliptic curve point multiplication obtained verifyng key where is the coordinates (x, y) representing the private key. Bitcoin has rules in the form of a public key, where the rule is:
Public Key = Xinteger , Yinteger
Public Key Bitcoin = 04 | Xinteger | Yinteger (10)
Xinteger = 32 bytes Y integer = 32 bytes
The next step is to create a corresponding public key, is x and y must be converted with 32 bytes hexadecimal.

```
print "X convert : ";
print "%032x" % PublicKey[0] ; print
print "Y convert : ";
print "%032x" % PublicKey[1] ; print
```

Fig. 10: Figure Command for converted x,y for adjust to the bitcoin rules

Publickey [0] is X and publickey [1] represents Y where% 032x to generate 32 hexadecimal string.

```
X : 36452638215417836447039143289354016844852272259204932072974517859215438473420
X converted : 5097764ac62c32066c9291c1b884ad20df171e974d9df98921a48ce9b35768cc
Y : 20712139261975913708573257814401233016007852862210270405001214969154851181194
Y converted : 2dcaa7c771b4225155c2e76a77f884b612dbca6e52f38d245717ac1d2916128a
```

Fig. 11: Figure Result converted x,y for adjust to the bitcoin rules

After having the X and Y plots converted, the next step is to create a Public key Bitcoin where 04 (public key initialization) is added with the conversion result X and Y

```
print "04" + "%032x" % PublicKey[0] + "%032x" % PublicKey[1];
```

Fig. 12: Figure Command for make Bitcoin public key

```
Public Key :
045097764ac62c32066c9291c1b884ad20df171e974d9df98921a48ce9b35768cc2dcaa7c771b4225155c2
e76a77f884b612dbca6e52f38d245717ac1d2916128a
```

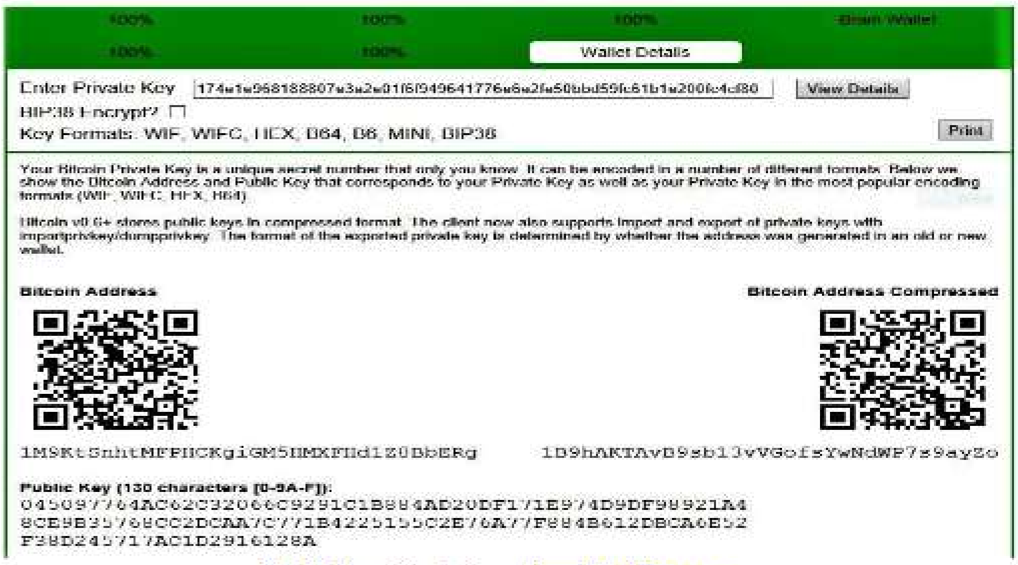
Fig. 13: Figure Result of Bitcoin public key

After going through the above process, it will generate a private key and public key in accordance with the rules imposed bitcoin.

```
Private key  :
174E1E968188807E3A2E01F6F949641776E6E2FE50BBD59FC61B1E200FC4CF80
Public key   :
045097764ac62c32066c9291c1b884ad20df171e974d9df98921a48ce9b35768cc2dcaa7c771b4225155c2
e76a77f884b612dbca6e52f38d245717ac1d2916128a
```

Fig. 14: Figure Result of Bitcoin private key and public key

The above figure is the result of a run program which displays private key and public key, where to determine whether the result of the program is in accordance with the prevailing standard, we use www.bitaddress.org as the comparison data.



Enter Private Key 174e1e968188807e3a2e01f6f949641776e6e2fe50bbd59fc61b1e200fc4cf80
BIP38 Encrypt? ☐
Key Formats: WIF, WIFC, HEX, B64, B6, MINI, BIP38

Your Bitcoin Private Key is a unique secret number that only you know. It can be encoded in a number of different formats. Below we show the Bitcoin Address and Public Key that corresponds to your Private Key as well as your Private Key in the most popular encoding formats (WIF, WIFC, HEX, B64).

Bitcoin v0.6+ stores public keys in compressed format. The client now also supports import and export of private keys with importprivkey/dumpprivkey. The format of the exported private key is determined by whether the address was generated in an old or new wallet.

Bitcoin Address
1MSKtSnhtMFFHCKgiGM5HMXFHd1Z0BbERg

Bitcoin Address Compressed
1D9hAKTAvB9sb1JvVGofsYwNdWP7s9ayZo

Public Key (130 characters [0-9A-F]):
045097764AC62C32066C9291C1B884AD20DF171E974D9DF98921A4
8CE9B35768CC2DCAA7C771B4225155C2E76A77F884B612DBCA6E52
F38D245717AC1D2916128A

Fig. 15: Figure Result from of www.bitaddress.org

Private Key Hexadecimal Format (64 characters [0-9A-F]):
174E1E968188807E3A2E01F6F949641776E6E2FE50BBD59FC61B1E200FC4CF80
Public Key (130 characters [0-9A-F]):
045097764AC62C32066C9291C1B884AD20DF171E974D9DF98921A48CE9B35768CC2DCAA7C
771B4225155C2E76A77F884B612DBCA6E52F38D245717AC1D2916128A

Fig. 16: Figure Result from of www.bitaddress.org copy to text box

4.2 Analysis

The key in the bitcoin system is a security factor and also the identity of the bitcoin owner where the private key is the main and secret identity of the bitcoin owner and the public key as the global identity of the owner.

This private key consists of 256 bits or 64 hexadecimal. This allows the private key to be made up of many possibilities, if the private key is changed in decimal form it will generate a quinn decilion number so that there is 10^{48} probability of its inclusion which will make it more difficult to do logging.

Public key in the manufacture using curve secp256k1 where the curve has a large range, in the process of making it used Elliptic Curve Point Multiplication to determine the point of coordinates of the private key in the curve, because the public key is a combination of x and y coordinates which is the represent point of the private key.

Elliptic curve point multiplication consists of point addition and doubling point it also increases the security of this key, where the process is done until found a suitable coordinate for private key, in its application in addition program or doubling usually done more than 300 times for a private key 64 character, so to do brute force from public key to get private key really need big resource.

With a large private key range and ECDSA process with Elliptic curve point multiplication will improve the security quality of the key, if a computer can do brute force with 1 trillion guess / second cap, it will take a lot of time to determine the private key of 10^{48} possibilities.

5. Conclusion

The conclusion of this research is that key is a safety factor in bitcoin system where this key consists of two parts: private key as the main identity of bitcoin and public key where the bitcoin public key system will be used in address creation.

To create a private key enough to generate 256 bits or 64 hexadecimal characters, this is a large number, for generating a public key where using the secpt256k1 curve that has a range for a large private key, since this public key is actually the private key coordinate in the curve, where to determine the coordinates are done Elliptic curve point multiplication process, this process is also one of the key security because in the process there is a complex calculation.

Using a private key consisting of 64 hexadecimal characters and using the secp256k1 curve and the Elliptic curve point multiplication calculation will generate a private key and public key corresponding to the key blockchain bitcoin provision and corresponding to www.bitaddress.org.

6. References

Benny Roy P.N., et al, Elliptic Curve Digital Signature Algorithm (ECDSA). Institut Teknologi Bandung (ITB) at Bandung.

Danielle Drainville, 2012, An Analysis of the Bitcoin Electronic Cash System. University of Waterloo at Canada.
Danny Yuxing Huang, et al, 2014, Botcoin: Monetizing Stolen Cycles. Network and Distributed System Security Symposium

Daniel R. L. Brown 2010 SEC 2: Recommended Elliptic Curve Domain Parameters Certicom Corp.

Nicolas Dorier. 2015. Programming Blockchain. Github

https://github.com/ProgrammingBlockchain/ProgrammingBlockchain

Rama Febriyan. 2015. Perbandingan Digital Signature Algorithm dan Elliptic Curve Digital Signature Algorithm.

Institut Teknologi Bandung (ITB), at Bandung.

Richard Caetano. 2015. Learning Bitcoin Embrace the new world of finance by leveraging the power of crypto-currencies using bitcoin and blockchain. PACK PUBLISHING www.it-ebooks.info

Rizqi Firmansyah ., Wahyu Suadi ., 2011, Implementasi Kriptografi dan Steganografi pada Media Gambar dengan Menggunakan Metode DES dan Region-Embed Data Density, Institut Teknologi Sepuluh Nopember at Surabaya.
Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf
Secp256k1. https://en.bitcoin.it/wiki/Secp256k1. Accessed : 18 February 2017.

Technical background of version 1 Bitcoin addresses , available at : https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses . accessed on 28 April 2017.
Raw Transaction Format, available at : https://bitcoin.org/en/developer-reference#raw-transaction-format , accessed on 04 April 2017.