ISBN 978-981-11-7861-0
Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering (WCSE 2018)
Bangkok, 28-30 June, 2018, pp. 561-568
doi: 10.18178/wcse.2018.06.095

A Novel Algorithm for Bitcoin Address Generation using Elliptical Curve Cryptography

Pritam Gajkumar Shah[1+] , Namita Pritam Shah[2], Ramesh K B[3]

[1]Jain University Bangalore, India

[2] R V College of Engineering, Bangalore, India

[3] R V College of Engineering, Bangalore, India

Abstract. Bitcoin is a digital currency having no central bank and operates on peer to peer network. Bitcoin helps account users for anonymous transactions with very less processing fee. To trade Bitcoins user must have account called as Bitcoin address. This paper illustrate in depth process of generating Bitcoin address with novel Elliptical Curve Digital Signature algorithm. As Kolbitz curves are losing their confidence in user due to advancement of quantum computer, this paper advocate use of NIST recommended P192 elliptical curve over prime field. In depth experimentation has been done on MIRACL crypto library to demonstrate applicability of NIST p192 elliptical curves to generate bitcoin addresses.

Keywords: Bitcoin, crypto currency, peer to peer network, public key, private key, hash function, elliptical curve digital signature algorithm etc.

1. Introduction

Bitcoin is a decentralized electronic payment system based on peer-to-peer (P2P) network and a probabilistic distributed consensus protocol [1]. The Bitcoin is based on public key cryptography such as RSA [2] and ECC [3] having two keys namely 'public' key and 'private' key. Bitcoin account is always recognized by user's public key rather than BSB/IFSC code, account number and is referred as 'Bitcoin address'. The account number in Bitcoin is generated by performing Elliptical Curve Digital Signature Algorithm followed by hash functions with fixed output as shown in Fig.1. Bitcoin user uses this output in the form of QR code as receiving address for bitcoin. In Bitcoin, a user can have multiple addresses by generating multiple public keys. The Bitcoin and other crypto currencies are stored in 'wallet' which is software for recording of transactions of Bitcoins. The private key of the user is used to spend Bitcoins.

1.1. Bitcoin transaction procedure

In crypto currency Bitcoin transactions are administered by the nodes called as 'miners' for verifying its authenticity. Miners group together multiple transactions in to one package called as a 'block'. After verification of block, miners put blocks on public ledger called as 'block chain'. Miner charge small amount of money for validation work.

1.2. Trading of Bitcoin

Before trading of Bitcoin user must have digital wallet on his personal computer or smart phone. Wallet is the software which keeps the track of your Bitcoin transactions. In this wallet, you can transfer money by doing

+ Corresponding author. Tel.: +919964485911.
E-mail address: wsnpgs@gmail.com.

online banking in the local currency. Once the funds are available in your wallet, you can place order for Bitcoin through online exchanges like Unocoin, Bitbns as shown in Fig 1.

1.3.The Bitcoin Wallets

There are three different applications that user can use for Bitcoin Wallet creation.

Full client– This is like a standalone email server that handles all transactions without third-party server. User can control whole transaction on their own.

Lightweight client – This is a standalone email client that connects to a mail server for access to a mailbox. It stores user's Bitcoin but need server to access the network and make the transaction.

Web client – This is third party web server as shown which completes all transactions on behalf of you and easy to use. The Fig. 2 shows dashboard of commercial web client.

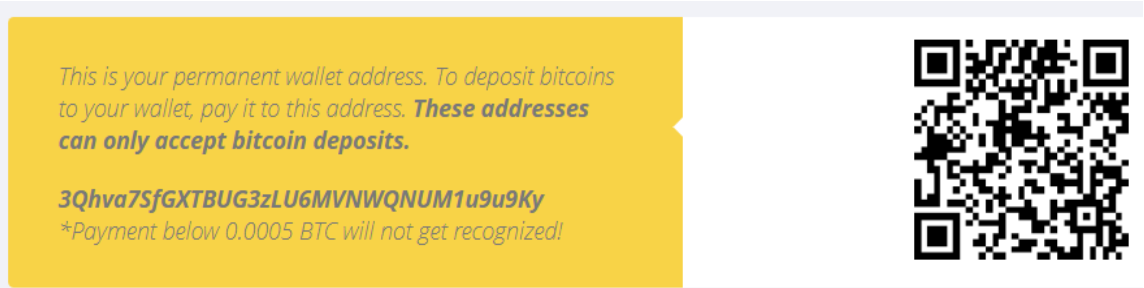

Fig. 1: Bitcoin Address with QR code [4].

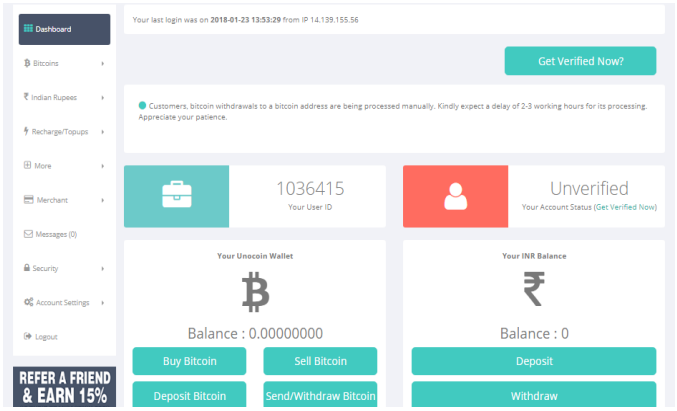

Fig. 2: Digital Wallet for trading Bitcoin [4].

2. Bitcoin Transactional Properties

The following properties are implied by [1],

1. Irreversible: After transaction is over you cannot reverse the transaction as no central authority involved.

2. Nameless users: Bitcoin account holders identity cannot be established with real world individuals. They are represented by their public keys as shown in Fig 1 and Fig 2 as 32 ASCII character whose identity cannot be established. It is usually possible to analyze the transaction flow but is very difficult to connect the real world identity of users with those addresses.

3. Physical location masking: Transaction is propagated instantly like email and are approved in a couple of minutes. Since they happen in a global network of computers which leads to hiding physical locations of the user.

4. Secure: Bitcoin transactions are based on public key cryptography. Only the owner of the private key can send crypto currency.

5. Permission less: Governments and central banks do not have any control over crypto currency at the moment.

2.1. Problems Associated with Bitcoin Addresses

A bitcoin address is 32 ‘alpha numeric character’ string which are printable and is very difficult to remember. Typing wrong Bitcoin address may result in wrong transactions and bitcoin transactions are always irreversible. This problem may be solved by use of ‘Quick Response’ code or by using Netki’s wallet name service. Netki services helps user to convert 32 characters string to easy wallet name provided that receiver has Netki software installed on their computers.

2.2. What is paper wallet in Bitcoin?

In case of digital wallets the java script generate the pair of private and public keys by using ECDSA as said earlier. The public and private keys are mathematically related which are discussed in in details in Section V. It is always advisable to take the print out of public and private key pair as the chances of misplacement are there due to hard drive issues and other reasons. In that case user may lose all the bitcoins. Fig 3 shows ‘paper wallet’. On the left hand side ‘hashed public key’ is shown which is your bitcoin address and has to be given to sender to transfer bitcoin in to your account. On the right hand side user private key is shown which is used to approve the transactions.

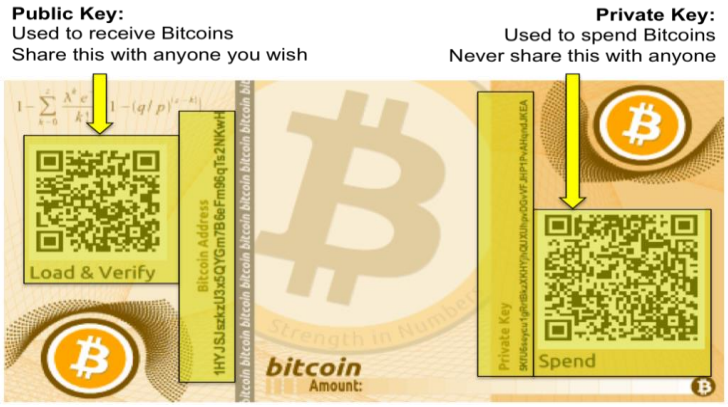

Fig. 3: Paper wallet indicating private and public key [5].

2.3. How to Transfer Bitcoin from Paper Wallet to Digital Web Client?

You can transfer Bitcoin from your paper wallet to digital wallet by using your private key. Most of the digital wallet providers’ web site are having this facility. The connection between your web browser and web server is always secured by use of SSL. But it is advisable not to use same Bitcoin address in future as its private key has touched internet. The private and public are mathematically related.

3. How the Bitcoin Addresses Generated?

This section elaborates the traditional process of bitcoin address generation in detail. As shown in Fig. 4, the bitcoin addresses are generated by using elliptical curve digital signature algorithm ECDSA and hash functions.

The algorithm ECDSA is having three major functions namely ‘key pair generation’, ‘signature generation’ and ‘signature verification’. Bitcoin addresses generation is related to ‘key pair generation of ECDSA’ and it works as follows:

• Choose a random number k from 1 to n-1 where n is the order of elliptical curve.

• Point Q is computed by doing scalar multiplication operation by multiplying k to base point G, Q=kG where k is the ‘private key’ and point Q is the ‘public key’ which is nothing but Bitcoin address without hash. In Bitcoin public key will go through hash operations before disclosing to outside world unlike normal public key cryptography in which public key is open. Due to discrete logarithmic problem involved, it is impossible to get private key from public key.

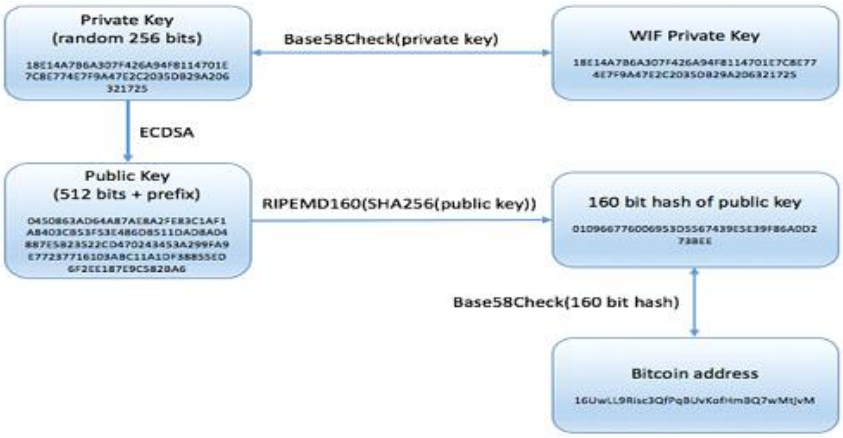

Fig. 4: Bitcoin address generation using ECDSA [6].

The step by step explanation of generating bitcoin address is given below and is based on example provided in Wikipedia.

Step 1- In this step a random number is generated which is needed to sign a Bitcoin transaction and for transfer and spending and is called 'private key'. The private key must be kept secret or else your bitcoins can be stolen.

18E14A7B6A307F426A94F8114701E7C8E774E7F9A47E2C2035DB29/A206321725 ← Private key

Step 2 - The corresponding public key is generated by using ECDSA algorithm. The public key is of 65 bytes in which 1 byte 0x04 prefixed and 32 bytes of X co-ordinate and 32 bytes of Y coordinate. The 0x04 indicates that the public key is compressed. Uncompressed key will be prefixed by 02 or 03 depending on whether y coordinate is even or odd number. Compressed key consists of only x coordinate.

0450863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B23522CD470243
453A299FA9E77237716103ABC11A1DF38855ED6F2EE187E9C582BA6 ← Public key

Step 3- On the above public key SHA 256 is performed.

600FFE422B4E00731A59557A5CCA46CC183944191006324A447BDB2D98D4B408

Step 4- On the result of Step 3, RIPEMD 160 performed.

010966776006953D5567439E5E39F86A0D273BEE

Step 5- Version byte is prefixed with result of step 4. It is 0x00 for main network.
00010966776006953D5567439E5E39F86A0D273BEE
Step 6 - SHA-256 performed on the result of step 6.
445C7A8007A93D8733188288BB320A8FE2DEBD2AE1B47F0F50BC10BAE845C094

Step 7 – Once again SHA-256 performed on the result of the previous SHA-256 hash in step 6.
D61967F63C7DD183914A4AE452C9F6AD5D462CE3D277798075B107615C1A8A30

Step 8 – The first 4 bytes of result in step 7 are called 'address checksum'.

D61967F6

Step 9- Add the 4 checksum bytes from stage 8 at the end of extended RIPEMD-160 hash from stage 5. This is the 25-byte Bitcoin Address which is used for user account identification.

00010966776006953D5567439E5E39F86A0D273BEED61967F6

Step 10 - Convert the result of step 9 in to a 'base58 string using base58 encoding'.

Your Bitcoin address is given below which is string of printable characters.

16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM

4. Proposed Method for Bitcoin Address Generation

In the above section IV, we have seen the detailed procedure for generation of Bitcoin by using ECDSA and hash functions. This section throws light on a novel method of Bitcoin generation on MIRACL platform.

MIRACL is ‘Multi precision Integer Rational Arithmetic Crypto Library’ MIRACL in C/C++ language and can be installed on Microsoft Visual Studio 2010 as well as smaller devices. There are two data types in this crypto library namely ‘Big’ and ‘Flash’. Big data type is for large integers and Flash is for large rational numbers. The large integer protocols are based on Knuth’s algorithm. We used the operating system Windows 7 professional with AMD processor of 3.00 GHz and RAM of 2 GB with 32-bit operating system.

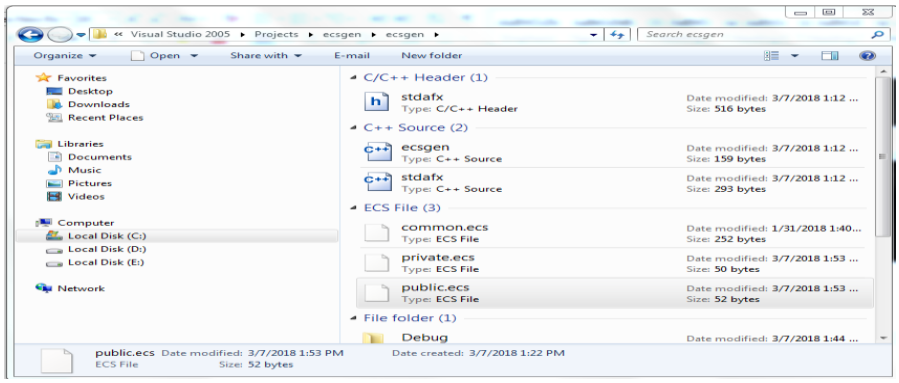

Fig. 5: Creation of public and private key pair by using ECDSA.

Also, the traditional bitcoin address generation uses the elliptic curve over F_p associated with a Koblitz curve secp256k1 which is specified by tuple $T=\{p,a,b,n,x,y\}$ where p is prime, a,b are constants, n is the order of point x,y are the coordinates of base point. The elliptical curve equation is $y^2=x^3+ax+b \bmod p$. The tuple of Kolbitz curve used for bitcoin generations in traditional method are as below-

secp256k1

p=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F

a= 0

b=0000000000000000000000000000000000000000000000000000000000000007

n=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141

Gx=79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798

Gy=483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8

As Kolbitz curves offer speedy scalar multiplication due to use of only point addition operation, they are popular for Bitcoin address generation. On the other hand, due to advancement of quantum computer confidence in Kolbitz curve is reducing day by day. So we proposed novel ECDSA algorithm based on NIST p192 elliptical curves with prime field. The tuples of NIST p192 are given below-

NIST-192

p=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFF

A= -3

B= 64210519E59C80E70FA7E9AB72243049FEB8DEECC146B9B1

n=FFFFFFFFFFFFFFFFFFFFFFFF99DEF836146BC9B1B4D22831

Gx=188DA80EB03090F67CBF20EB43A18800F4FF0AFD82FF1012

Gy=07192B95FFC8DA78631011ED6B24CDD573F977A11E794811

The 32-bit version of crypto library is used with Little Endian architecture. The above curve parameters called tuples are stored in file namely ‘common.ecs’. The running of algorithm which results in two files namely public.ecs and private.ecs as shown in Fig.5.

The second innovation authors have suggested in compression technique of public key. The compression of public key saves memory requirement in block chain. In traditional method 04 is prefixed to indicate the uncompressed key. In Compressed key x-coordinate is prefixed with 02 or 03 depending on whether y-coordinate is even or odd respectively.

In our implementation of 'compressed public key' we use only least significant bit of y- coordinate (either 1 or 0) to distinguish between two solutions of quadratic equation of elliptical curve as shown in Fig 6. Compressed public key of Fig 9 shows prefixing of 1 in fornt of x coordinate as y coordinate last hex character is 'F'. In binary format it is '1111'. The LSB is shown in red ink. Also compressed key of Fig.10 shows 0 prefixed with x co-ordinate as LSB of y coordinate is '4' means '0100'. It is possible to solve quadratic equation by using x-coordinate in elliptical curve to get y- coordinate. But due to symmetry of curve along y axis there exists two solutions and LSB of y coordinate helps to choose proper one. Our proposed technique can save 50 % of storage in block chain due to single coordinate system in compressed format.

In traditional Bitcoin address, public key is represented by using 64 bytes (32 bytes each for x- and y-coordinates), on the other hand we have proposed shorter public key of 48 bytes (24 bytes each for x- and y-coordinates) for Bitcoin address as shown in Fig 8. In our implementation the seed of length 9 is used for generation of private key k of 24 bytes as shown in Fig. 7.The Private Key k which we obtained is,

98D9666F14EDDFE134675484F8EAA0C126DE7F1E515C9594

The uncompressed public key with x and y coordinate is as per below,

public key X coordinate =

F72BAD322D6C20CEBE39E86FB3498543A4ECBB35EC172B5C

public key Y coordinate =

2FB431106965311607A629E6B8E8951432B16EE463E597F

Compressed public key = 1

F72BAD322D6C20CEBE39E86FB3498543A4ECBB35EC172B5C

Fig. 6: Public key created in compressed format.

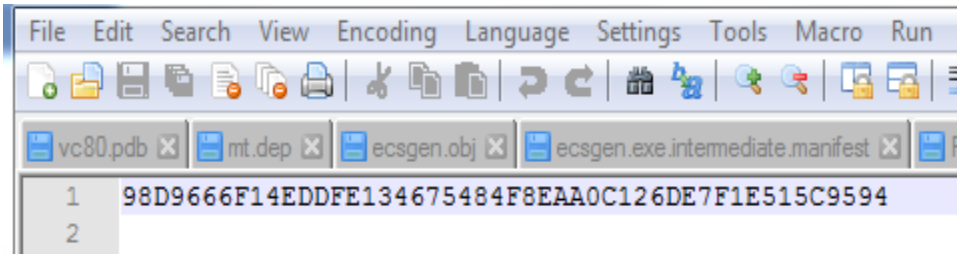

Fig. 7: Private key stored in private.ecs file.

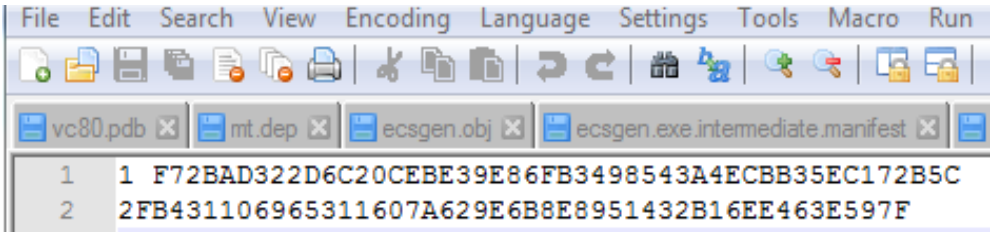

Fig. 8: Public key showing x and y coordinates.

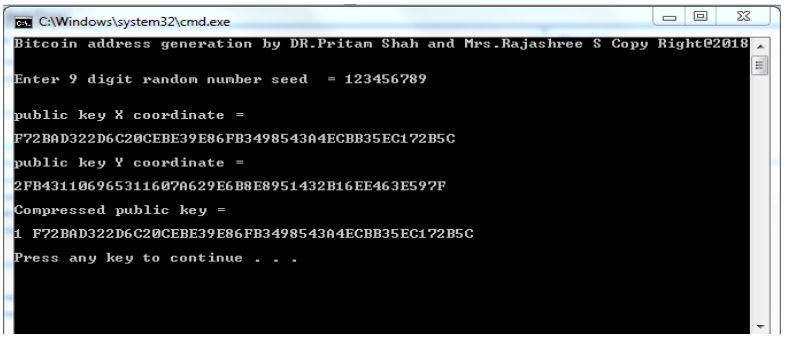

Fig. 9: Compressed Public key with LSB of y- cordinate as 1.

Fig. 10: Compressed Public key with LSB of y-coordinate as 0.

As shown in Fig 11, author implemented ECDSA based on 'projective coordinates' rather than affine coordinate of elliptical curve and with pre-computation technique. Scalar multiplication is based on addition subtraction technique. It is found that calculating bitcoin address with pre-computation and projective coordinates is faster as compared to traditional method of affine coordinates as there is no need to do 'modular inversion' operation in projective coordinates. The signature generation completed in only 0.52 ms as compared to without pre-computation which took 2.92 ms.

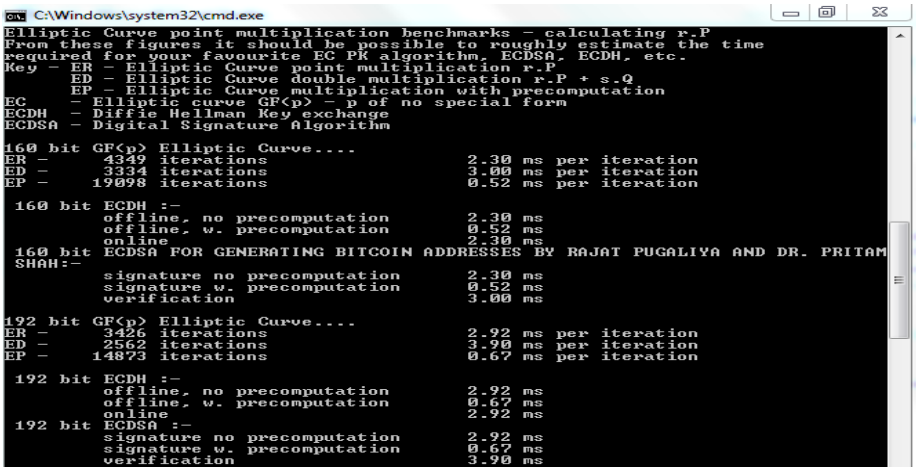

Fig. 11: Benchmarking of ECDSA based on pre-computation.

5. Conclusion

In this paper, we have presented a novel ECDSA algorithm to calculate Bitcoin addresses based on NIST p-192 elliptical curve over prime field as people are losing confidence on Kolbitz curve due to advancement in quantum computers. NIST p-192 curve are safer and also results in 48 bytes of shorter public key which will acquire less storage on block chain. We have also proposed use of 'projective coordinate' to calculate public, private key pair and use of pre-computational approach for faster scalar multiplication process. The new approach will result in stronger Bitcoin addresses with shorter length and fast calculation time.

6. References

[1] Conti, M., et al., A Survey on Security and Privacy Issues of Bitcoin. CoRR, 2017. abs/1706.00916.

[2] Rivest, R.L., A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 1978. 21(2): p. 120-126.

[3] Blake, I., et al., Advances in Elliptic Curve Cryptography (London Mathematical Society Lecture Note Series). 2005: Cambridge University Press.

[4] Unocoin, https://www.unocoin.com/

[5] https://steemit.com/cryptocurrency/

[6] https://asecuritysite.com/encryption/Bitcoin

[7] Hankerson, D, Vanstone, S.;Menezes, A. (2004). Guide to Elliptic Curve Cryptography. Springer Professional Computing. New York

[8] Accredited Standards Committee X9, American National Standard X9.62-2005, Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA), November 16, 2005

[9] Daniel R. L. Brown, Generic Groups, Collision Resistance, and ECDSA, Designs, Codes and Cryptography 119–152, 2005