Exploration and Discussion for Bitcoin Encryption Algorithms

Keyu Yan
Software Engineering, North University of China, Shanglan street, Taiyuan, China

Keywords: Bitcoin, Hash Functions, ECDSA, Asymmetric Cryptography, Digital Signatures.

Abstract: Bitcoin has played a more and more crucial role today with the popularity of the Internet and it has unparalleled advantages compared with physical currency since its quality of high safety and privacy-protection. The development of bitcoin encryption algorithms is engrossing, so this article reviews some main encryption algorithms. One of the most important functions used in bitcoin encryption is hash function, the Secure Hash Algorithm-256 (SHA-256) and RACE Integrity Primitives Evaluation Message Digest (RIPEMD-160) are based on it and the complex process makes sure reliability. Another cryptographic algorithm the paper will illustrate is Elliptic Curve Digital Signature Algorithm (ECDSA) which is a typical asymmetric cryptography. The process primarily comprises three sections, which will be elaborated upon in this paper. These algorithms are broadly implemented in numerous areas of daily life, yet they are not without their drawbacks. To address these issues, scientists have undertaken extensive optimization efforts, resulting in the creation of safer and more advanced technologies to satisfy a greater range of needs. Thus, this article provides a detailed review of Bitcoin encryption algorithms.

1 INTRODUCTION

Before Bitcoin was made available as open-source software in 2009, Satoshi Nakamoto originally presented the idea behind the cryptocurrency (Rahouti et al. 2018). A virtual coin with peer-to-peer encryption is called Bitcoin. It is projected to reach its issuance completion by the year 2140, with a total cap of 21 million units, underscoring its rarity. As a consensus-driven network, Bitcoin introduces an entirely digital payment system (Vyas & Lunagaria 2014). To some extent, bitcoin is an excellent reflection of Internet currency, because of its characteristics of celerity, security and no national boundary. In contrast to official cash, bitcoin is created by network nodes and is distributed globally instead of being issued by the third-party institution, so everyone is able to produce it. People just need a computer which access internet can buy and sell bitcoin, and outsiders cannot obtain user's message during the process of transaction.

Traditionally, physical currency was the norm, but it posed issues of convenience and was susceptible to loss or theft. In contrast, bitcoin has several advantages. Firstly, it enhances personal data protection. Users of Bitcoin are not at great danger in the event that a merchant or transaction partner experiences a cyber-attack and loses traditional financial or personal data belonging to its clients or itself (Dumitrescu 2017). Secondly, it has lower transaction fees. The transaction fees of bitcoin are almost five times lower than that of credit cards for the same transaction values (Dumitrescu 2017). Thirdly, refund fraud is prevented for the merchants by the rapidity of the transfer. The speed of bitcoin is quicker than banks (approximately 10 to 30 minutes) and before they deliver the goods or services, the sellers have enough time to review the transaction. Fourthly, it is not affected by inflation (Dumitrescu 2017). Bitcoin also has the advantages of decentralization, immutability, transparency, borderless circulation and exclusive ownership, all of which are crucial features.

The genesis and evolution of Bitcoin present a fascinating narrative. Wei Dai first proposed a term called "crypto-currency" on the cypherpunks mailing list in 1998 to explain the concept of a virtual currency that is created and transacted using encryption rather than a central authority (Vyas & Lunagaria 2014). Bitcoin was regarded as the first successful implementation of this concept (Vyas & Lunagaria 2014). Then Satoshi Nakamoto appeared

Yan, K.
Exploration and Discussion for Bitcoin Encryption Algorithms.
DOI: 10.5220/0012819100004547

In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 52-57
ISBN: 978-989-758-690-3

on the cryptography scene in 2008 and published his famous whitepaper (Ducrée 2022). Then in 2009, the first bitcoin with serial number 0 genesis and blockchain was connected to form a chain, marking the birth of bitcoin.

As a transaction currency, bitcoin mainly uses three functions of cryptography: hash functions, asymmetric cryptography and digital signatures. The development of bitcoin encryption algorithm has experienced long periods. MD5 was designed in 1992 as an improvement of MD4, but its collisions have been attacked so the function is not safe today (Wang & Yu 2005). Then The National Institute of Standards and Technology (NIST) adopted a function called Secure Hash Algorithm-1(SHA-1) in 1995, which produces 160-bit hashes (Gayoso Martinez et al. 2020). However, many international organizations and institutions abandoned it after its collisions were found (Gayoso Martinez et al. 2020).Another function that also offers 160-bit hashes is the RACE Integrity Primitives Evaluation Message Digest(RIPEMD-160) function which was designed in 1996 and its collision resistance has not been compromised, so the function is still utilized in certain situations today. Then the SHA-2 Hash Family Standard, which includes SHA-256, SHA-384, and SHA-512, was introduced by the NIST in 2002 (Sun et al. 2007). Its implementation has the ability to replace the MD5 and SHA-1 hash functions in an efficient manner, resulting in enhanced protection strength (Sun et al. 2007). In addition to hash algorithm, there is an asymmetric encryption algorithm in bitcoin called Elliptic Curve Digital Signature Algorithm (ECDSA), which has gained popularity in this day and age. Meanwhile, some new research has been done to optimize the existing algorithms and develop a higher security based on bitcoin today.

The rise of cryptocurrency is unquestionably poised to have a substantial impact on the global economic landscape (Fauzi et al. 2020). Bitcoin encryption ensures the security and stability of the Internet. It not only is a digital currency, but also provides a creative innovation in the future. Therefore, it is necessary to summarize the encryption algorithm of bitcoin and its development, which is the aim of this paper. The rest of this review paper is divided into three sections. Section 2 investigates some main bitcoin encryption algorithms. Section 3 discusses some defects, current development and future prospects of bitcoin encryption. Finally, conclusions are presented in Section 4.

2 MAIN BITCOIN ENCRYPTION ALGORITHMS

2.1 Framework of Encryption Algorithm Related to the Bitcoin

The hash function that is mainly used technology for encryption algorithm in the bitcoin has been applied in widespread fields such as communication, finance, and confidential computation as a fundamental cryptographic algorithm (Yang et al. 2017). There are some obvious features of hash functions: the information being provided can be any length, while the outcome has a set length; fast calculating speed; hiding and one-way. Furthermore, hash functions have the quality of collision resistance which refers that finding any two diverse inputs x, y, then making H(x)=H(y) is computationally infeasible. The initial hash functions were suggested for application in digital signature protocols with the aim of enhancing their efficiency. This was achieved by constructing signatures using the digest of data elements rather than the entire elements (Gayoso Martinez et al. 2020). Digital signature has played a more and more important role nowadays due to its feature of unforgeability and non-repudiation.

Another encryption algorithm used in bitcoin is ECDSA which is an asymmetric cryptography system. The ECDSA is a simulation of digital signature algorithm (DSA) using Elliptic Curve Cryptography (ECC) (Abidi et al. 2014). The process of ECDSA mainly involves three distinct phases and ECDSA provides integrity, authentication, and non-repudiation. The safety of ECDSA algorithms is the primary factor determining the cybersecurity of Bitcoin (Wang 2014).

2.2 Hash Functions

2.2.1 SHA-256

For any length of message, a 256-bit long hash value will be generated, known as the message digest. The process of the algorithm includes two sections: preprocessing and digest computing (Algredo-Badillo et al. 2013).

The preprocessing is required to add additional bits to the message until it reaches a size that is a multiple of 512 bits. Subsequently, the padded message is divided into N message blocks, denoted as M_1,M_2,...,M_n, with each block having a size of 512 bits. The specific implementation method is to append a bit 1 and k bits 0 at the end of the message

to satisfy $L + 1 + k = 448 \bmod 512$ (The "L" stands for the original message's length, expressed in bits.) (Kammoun et al. 2020). It is vital to emphasize that padding is still required, regardless of whether the message's initial length has become a multiple of 512 (Kammoun et al. 2020). So, Padding is to add at least one character, up to a maximum of 512 characters. Then, to reflect the length of the initial text, the end of the message need insert 64 more bits. Finally, the padded message is broken into n blocks of 512-bit. Fig. 1 depicts the particular procedure.

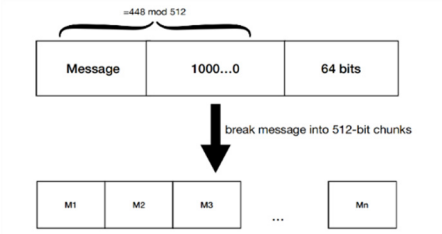


Figure 1: Preprocessing in SHA-256 (Original).

Following the completion of the padding phase, the 8 state registers A to H are set to predetermined 32-bit constants h0 to h7 for the initial message block (Gilbert & Handschuh 2003). The eight hash original values are shown in Fig. 2. The second step is to break each block into sixteen 32-bit big-endian words: W_0,…,W_{15}, and expand them into 64 words W_0 to W_{63}, One for each iteration of the compression function (Gilbert & Handschuh 2003). The message's corresponding block breaks down the first 16 words immediately, then the remaining words can be computed by the formula (Gilbert & Handschuh 2003): $W_t = \sigma_1(W_t - 2) + W_t - 7 + \sigma_0(W_t - 15) + W_t - 16$ (where $\sigma_0(x) = R^7(x) \oplus R^{18}(x) \oplus S^3(x)$ and $\sigma_1(x) = R^{17}(x) \oplus R^{19}(x) \oplus S^{10}(x)$). Finally, 64 iterations for the compression function which operates on a 512-bit block generated by the padding process need to be executed using some intricate formulas and 64 fixed constants K_t. Fig. 3 illustrates the process.

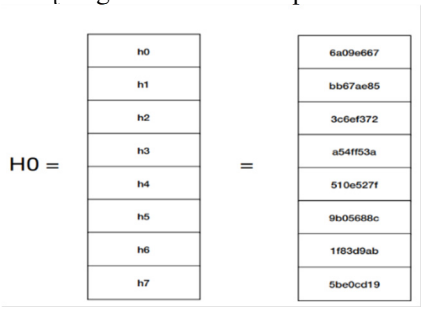
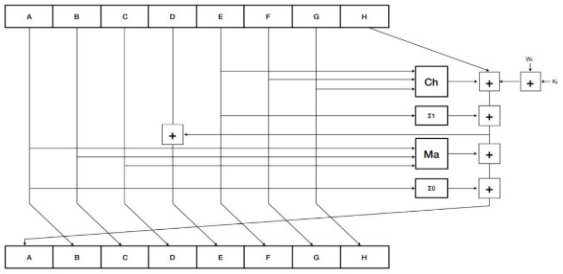


Figure 2: Eight original hash values which have been defined in advance so they can be used directly (Original).



$$Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z)$$

$$Ma(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$\Sigma_0(x) = R^2(x) \oplus R^{13}(x) \oplus R^{22}(x)$$

$$\Sigma_1(x) = R^6(x) \oplus R^{11}(x) \oplus R^{25}(x)$$

Figure 3: The process of 64 iterations in SHA-256.K_t is the t th key, corresponding to the 64 fixed constants. Four formulas have been shown and $\wedge, \oplus, \neg$ respectively represent the bitwise AND, bitwise XOR and bitwise NOT.R^n,$S^n$are respectively right rotation and right shift n bits (Picture credit: Original).

2.2.2 RIPEMD-160

The input format of RIPEMD-160 is similar to that of SHA-256 which based on MD5, and generates a 160-bit output (Ng et al. 2004). Firstly, the message that was entered needs to be padded to make sure the overall input size is a multiple of 512 bits and the method is identical with that of SHA-256(appending a single 1 followed by a sequence of 0s, with the number of 0s ranging from 0 to 511) and each block is divided into 16 32-bit words. Each word is used as a block, and 5 words are input each time to enter a hash operation, then a hash value with 32 bits is output.5 32-bit hash values will be gained after all the blocks are computed and a 160-bit hash value can be obtained when connecting them together (Preneel et al. 1997).

Fig. 4 reveals the operation of RIPEMD-160.Each 5 32-bit words does the operation 16 times. In each iteration, there are two concurrent lines, and five distinct non-linear functions that correspond to the five rounds. When the left and right wheel each calculate 80 rounds of the operation, the value of the chaining variable is modified by incorporating the five preceding variables values, five succeeding variables values and the present value of the chaining variable (Ng et al. 2004).

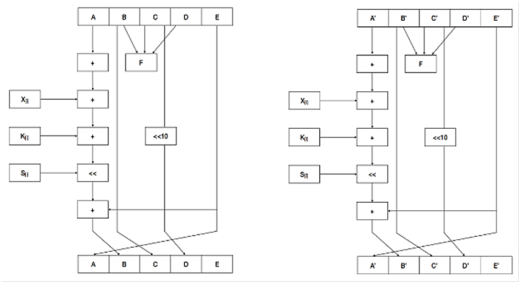


Figure 4: Operation of RIPEMD-160 (Picture credit: Original).

The basic operation is the same in each round, the difference is only in the F function which is called compression function. The input of F is $32 \times 3 = 96$bits,but the output is only 32bits.The output after each computation is computed as the input for the next round,transforming the F function every 16 turns until the end of the fifth round of 16.The input word is denoted as $X_{[i]}$, and $K_{[i]}$ represents one of the ten 32-bit constants utilized in the algorithm which have been defined in advance. $S_{[i]}$ is indicative of the 4-bit control for the left rotation operation, and "$\ll 10$" signifies a 10-bit left cyclic shift operation.

2.3 SCDSA

In the ECDSA, the process of Generating and verifying signatures closely resembles that of the DSA algorithm, and creating key is derived from the ECC algorithm (Wang 2014).

2.3.1 Key Creation

The key couple of an entity A is linked to a specific set of elliptic curve domain parameters D= (q, FR, a, b, G, n, h). The method to generate cryptographically secure domain parameters has been thrown light on (Johnson et al. 2001). Every entity A carries out the subsequent actions:

1) Choose a random integer d that satisfy $1 \leqq d \leqq n - 1$.

2) Compute $Q = dG$.

3)The public key of A is Q, and its private key is d.

2.3.2 Signature Generation

The step is to exploit the domain parameters of A and private key d to sign a given message M. The process is shown below (Wang 2014; Algredo-Badillo et al. 2013):

1) Select an arbitrary integer k ($1 \leqq k \leqq n - 1$).

2) Compute $kG = (x_1, y_1)$ and $r = x_1 \bmod n$(Return to step 1 if $r = 0$).

3)Compute $s = k^{-1}(SHA - 1(M) + dr) \bmod n$ (Return to step 1 if $s = 0$).

4) The signature of A for the message M is(r,s).

2.3.3 Signature Verification

B has the public global domain parameters D=(q, FR, a, b, G, n, h) and the public key Q of A. After receiving the message which sent by A and the digital signature, the verification is followed:

1) Check whether r and s are integers from 1 to n-1.
2) Calculate $w = s^{-1} \bmod n$.
3) Calculate $u_1 = SHA - 1(M) w \bmod n$ and $u_2 = rw \bmod n$.
4) Calculate $X = u_1 G + u_2 Q$.If $X = 0$,then reject the signature. Otherwise, transform x_1 into an integer when $X = (x_1, y_1)$ and calculate $v = x_1 \bmod n$.
5) When $v = r$,accept A's signature.

3 DISCUSSION

SHA-256 provides higher security than MD5 and SHA-1 thanks to its more intricate process that provides resistance to various attacks. Moreover, it is not viable for some attack techniques to assault SHA-256 such as Chabaud and Joux's attack, Dobbertin's Attack and Differential Attacks (Gilbert & Handschuh 2003). However, its slower operational speed is a notable drawback, making the process quite time-intensive. Many researches have challenged its collision resistance. Besides, making minor adjustments to the SHA-2 criteria leads to a hash function that lacks collision resistance, such as Symmetric Constants and Exor (Gilbert & Handschuh 2003). The RIPEMD-160 algorithm is utilized for the creation of the bitcoin account in conjunction with SHA-256 today. Although numerous hash algorithms within the MD-SHA hash family have been compromised, RIPEMD-160 continues to be resilient (Liu et al. 2023). There are a wide range of applications in bitcoin encryption algorithms based on hash functions because of their evident advantages such as the blockchain and cyber security domain. The RIPEMD-160 has high computing speed performance and can complete the hash calculation of a large amount of data in a short time, while there are still certain limitations. For instance, with the development of quantum computing technology, the algorithm may face the risk of quantum attack. The most advanced collision attack was able to penetrate only 34 out of 80 rounds,

as reported in the CRYPTO 2019 conference (Liu et al. 2023). Meanwhile, a new collision attack on RIPEMD-160 using a novel approach for selecting variations in messaging and implementing new methods to effectively manage differing conditions in parallel across both branches has been reported in the paper. As a result, persistent evaluation and improvement should be valued.

As for ECDSA algorithm, it has garnered significant interest from researchers since its incomparable advantages over other public key cryptographic algorithms, so various institutions due to their ability to offer security and high efficiency in applications e.g. electronic finance, online transportation, education and so on. Algorithms utilizing elliptic curves offer significantly higher strength-per-key-bit and the unit bit power of the cryptography exceeds compared to that of other public key schemes. The algorithms boost security measures against a range of attacks while also enhancing system efficiency in speed, recall, and decreasing computing complexity and energy consumption within resource-constrained and large-scale systems (Al-Zubaidie et al. 2019).

Not only researchers have implemented many optimizations on existing encryption algorithms, but also some new technologies have appeared to increase protections. For instance, ciphertext-policy hierarchical attribute-based encryption (CP-HABE) has been developed to keep users' privacy from illegal users when an illegal transaction happens. In this scheme, a novel signature algorithm is employed to create wallet key pairs, replacing the elliptic curve signature, thereby linking wallet addresses with encrypted identities. The implement process has been demonstrated clearly and it is resistant for chosen-plaintext attacks (CPA) in the standard model, adapting the Bilinear Diffie-Hellman Exponent (BDHE) presumption (Wang & Gao 2018). Moving forward, the researchers will more concentrate on the domain and enhance the quality of the algorithms. Meanwhile, the majorization of the existing bitcoin encryption algorithms should also be a main research topic especially for the collision resistance and calculation speed.

4 CONCLUSION

This paper described bitcoin and its importance as well as development. Besides, the detailed implement process of some main bitcoin encryption algorithms was provided, including SHA-256, RIPEMD-260 and ECDSA. Then, the current advantages and limitations of these algorithms have been shown and this paper discussed some new cryptography based on bitcoin. However, this article did not contain some new domains which combining bitcoin with other emerging fields like AI models and machine learning. A more thorough investigation and more specialized article will be planned to completed in the future.

REFERENCES

M. Rahouti, K. Xiong, and N. Ghani. Bitcoin concepts, threats, and machine-learning security solutions.Ieee Access 6: 67189-67205(2018).

C. Vyas and M. Lunagaria. Security concerns and issues for bitcoin. Int. J. Comput. Appl, 10-12(2014).

G. Dumitrescu. Bitcoin–a brief analysis of the advantages and disadvantages. Global Economic Observer, 5(2): 63-71(2017).

J. Ducrée. Satoshi Nakamoto and the Origins of Bitcoin--Narratio in Nomine, Datis et Numeris. arXiv preprint arXiv:2206.10257 (2022).

X. Wang, H. Yu. How to break MD5 and other hash functions. Annual international conference on the theory and applications of cryptographic techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, 19-35(2005).

V. Gayoso Martinez, L. Hernández-Álvarez,L. Hernandez Encinas. Analysis of the cryptographic tools for blockchain and bitcoin. Mathematics, 8(1): 131(2020).

W. Sun, H. Guo, H. He, et al. Design and optimized implementation of the SHA-2 (256, 384, 512) hash algorithms. 2007 7th International Conference on ASIC. IEEE, 858-861(2007).

M. Fauzi, N. Paiman, Z. Othman. Bitcoin and cryptocurrency: Challenges, opportunities and future works. The Journal of Asian Finance, Economics and Business (JAFEB), 7(8): 695-704(2020).

Y. Yang, F. Chen, X. Zhang, et al. Research on the hash function structures and its application. Wireless Personal Communications, 94: 2969-2985(2017).

A. Abidi, B. Bouallegue, F. Kahri. Implementation of elliptic curve digital signature algorithm (ECDSA). 2014 Global Summit on Computer & Information Technology (GSCIT). IEEE, 1-6(2014).

D. Wang. Secure implementation of ECDSA signatures in bitcoin. MSc in Information Security (2014): 1-78.

I. Algredo-Badillo, C. Feregrino-Uribe, R. Cumplido, et al. FPGA-based implementation alternatives for the inner loop of the Secure Hash Algorithm SHA-256. Microprocessors and Microsystems, 37(6-7): 750-757(2013).

M. Kammoun,M. Elleuchi,M. Abid, et al. FPGA-based implementation of the SHA-256 hash algorithm. 2020 IEEE international conference on design & test of integrated micro & nano-systems (DTS). IEEE, 1-6 (2020).

H. Gilbert, H. Handschuh. Security analysis of SHA-256 and sisters. International workshop on selected areas in

cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg, 175-193 (2003).
C. Ng, T. Ng, K. Yip. A unified architecture of MD5 and RIPEMD-160 hash algorithms. 2004 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2: II-889(2004).
B. Preneel, A. Bosselaers, and H. Dobbertin. The cryptographic hash function RIPEMD-160. (1997): 9-14.
D. Johnson, A. Menezes, S. Vanstone. The elliptic curve digital signature algorithm (ECDSA). International journal of information security, 1: 36-63(2001).
F. Liu, et al. Analysis of RIPEMD-160: New Collision Attacks and Finding Characteristics with MILP. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cham: Springer Nature Switzerland (2023).
M. Al-Zubaidie,Z. Zhang, and J. Zhang. Efficient and secure ECDSA algorithm and its applications: A survey. arXiv preprint arXiv:1902.10313 (2019).
Y. Wang, and J. Gao. A regulation scheme based on the ciphertext-policy hierarchical attribute-based encryption in bitcoin system. IEEE Access 6 (2018): 16267-16278.