Application and Implementation of Multivariate Public Key Cryptosystem in Blockchain (Short Paper)

Ruping Shen[1(✉)], Hong Xiang[2], Xin Zhang[1], Bin Cai[1], and Tao Xiang[3]

[1] School of Big Data and Software Engineering, Chongqing University, Chongqing, China
{shenruping,zhang.x,caibin}@cqu.edu.cn
[2] Key Laboratory of Dependable Service Computing in Cyber Physical Society Chongqing University, Ministry of Education, Chongqing, China
xianghong@cqu.edu.cn
[3] School of Computer Science, Chongqing University, Chongqing, China
txiang@cqu.edu.cn

Abstract. Blockchain is one of the most revolutionary and innovative technologies in recent years. The traditional asymmetric encryption algorithms guarantee the security of data on blockchain. However, with the rapid development of quantum computing technologies, as long as large-scale quantum computers appear, these kind of encryption systems can be deciphered by shor algorithm in polynomial time. Therefore, blockchain technologies are going to face potential security threats. To solve this problem, the best solution at present is to replace the asymmetric encryption algorithms in the blockchain with post-quantum cryptosystems. In this paper, we apply the Rainbow algorithm with high signature efficiency to the existing Ethereum platform, and test the feasibility of the scheme by building a private chain. In addition, we compare the signature efficiency of Rainbow algorithm with ECDSA, which is expected to provide direction and inspiration for future research on blockchain resistance to quantum computing.

Keywords: Blockchain · Quantum computers · Post-quantum cryptosystems

1 Introduction

Blockchain [1,2] has attracted increasing attention because of its decentralization in recent years. Blockchain emerges originally as the core technology of Bitcoin [3]. Subsequently, in 2015, the emergence of blockchain platforms represented by Ethereum [4] and Hyperledger [5] has once again pushed blockchain technology to a climax of research. However, as a new technology, blockchain will inevitably face various problems and challenges [6]. The security of data on the

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 419–428, 2019.
https://doi.org/10.1007/978-3-030-30146-0_29

blockchain mainly depends on the traditional asymmetric cryptosystems. For example, the most popular Bitcoin and Ethereum use the Elliptic Curve Digital Signature Algorithm (ECDSA) [7]. However, with the rapid development of quantum computing technologies, as long as large-scale quantum computers appear, the Shor algorithm [8] can decipher the ECDSA in a short time, thus the blockchain technologies face a huge security threat [9].

Although the current blockchain security issue is particularly important, everyone is still on the stage of theoretical analysis stage. Until 2017, the UK released Quantum-Resistant Ledger [10], which uses an encryption algorithm that can resist quantum attacks, and successfully combines blockchain technologies with post-quantum cryptosystems. However, if post-quantum cryptosystems want to be widely used, which requires the formulation of relevant international standards.

In 2017, NIST published the results of the first batch of post-quantum cryptosystems [11,12]. It can be seen that post-quantum cryptographic design schemes have been officially put on the agenda, which means that blockchain technologies relying on the traditional cryptosystems must make an alternative plan to the advent of quantum computers. There are mainly five categories of post-quantum cryptosystems [13]: Hash-based, Code-based, Lattice-based, Isogeny-based and Multivariate Public Key cryptosystems. Compared with other post-quantum cryptosystems, the research on multivariate public key cryptosystems started relatively early. Thus, there are many mature multivariate public key cryptosystems. Apart from resisting the attack of quantum computers, it also has the advantages of fast computing speed and less computing resource, which is consistent with the real-time needs of blockchain. Therefore, combining multivariate public key cryptosystems with blockchain technologies is of great significance.

Based on the above backgrounds, we apply the most popular Rainbow signature scheme to Ethereum. The rest of this paper is organized as follows: Sect. 2 introduces the knowledge of Blockchain, Multivariate Public Key Cryptosystem and Rainbow signature scheme; In Sect. 3, we introduce the details of experiments; Sect. 4 analyzes the experimental results and Sect. 5 summarizes the full text.

2 Preliminaries

2.1 Blockchain

This section takes Ethereum as an example to introduce related technologies of blockchain. The total architecture of Ethereum is shown in Fig. 1.

Smart contract [14] is the main innovation of Ethereum. It is a collection of code and data (state), and can also be understood as a contract written in code that can be executed automatically on blockchain.

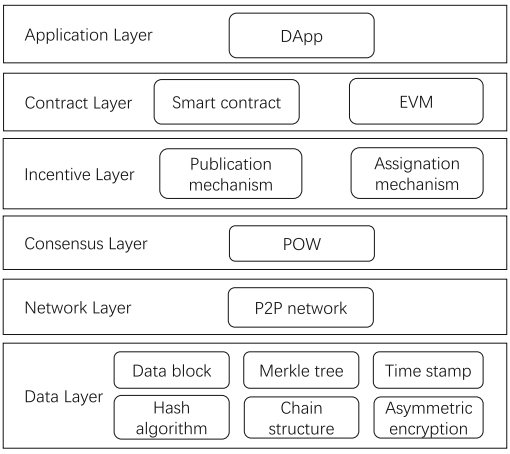

Fig. 1. Ethereum overall architecture.

2.2 Multivariate Public Key Cryptosystem

Multivariate Public Key Cryptosystem [15] aims to design a secure encryption and signature scheme by constructing a multivariate quadratic equation as a public key. Its key compositions are respectively: $pk = P = S \circ F \circ L, sk = \{S, F, L\}$. This paper focuses on the multivariate signature scheme. Its process of signature and verification is shown in Fig. 2.

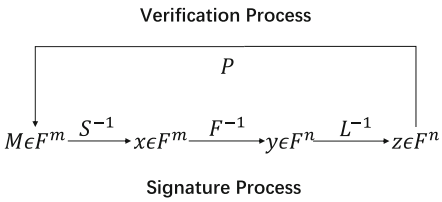

Fig. 2. Process of Multivariate Signature Scheme.

Signature: Suppose $M \in F^m$, one calculates sequentially $x = S^{-1}(M) \in F^m, y = F^{-1}(x) \in F^n$ and $z = L^{-1}(y) \in F^n$. z is the signature of M.

Verification: One calculates $M^{'} = P(z) \in F^m$, if $M^{'} = M$, the signature is accepted; otherwise, reject the signature.

2.3 Rainbow Signature Scheme

In 2005, Ding and Schmidt [16] improved the Unbalanced Oil-Vinegar (UOV) scheme and proposed Rainbow [17], which is a multilayer UOV scheme. Due to its high signature efficiency, Rainbow signature scheme is considered as one of the most promising multivariate signature schemes. The core difference between

different multivariate public key cryptographic algorithms is that the construction of the private key F is different. Therefore, the key generation process of the Rainbow algorithm is described in detail:

Let $F = F_q$ be a finite field with q elements, $n \in N$ and $v_1 < v_2 < \cdots < v_l < v_{l+1} = n$ be a sequence of integers. We set $m = n - v_1, O_i = \{v_i + 1, \ldots, v_{i+1}\}$ and $V_i = \{1, \ldots, v_i\}(1, \ldots, l)$

The private key consists of two invertible affine map $S : F^m \to F^m$, $L : F^n \to F^n$ and a central map $F(f^{(v_1+1)}(x), \ldots, f^{(n)}(x)) : F^n \to F^m$. The expression of the polynomials $f^{(i)}(i = v_1 + 1, \ldots, n)$ is

$$f^{(i)} = \sum_{k,l \in V_j} \alpha_{k,l}^{(i)} \cdot x_k \cdot x_l + \sum_{k \in V_j, l \in O_j} \beta_{k,l}^{(i)} \cdot x_k \cdot x_l + \sum_{k \in V_j \cup O_j} \gamma_k^{(i)} \cdot x_k + \eta^{(i)}$$

Here, the coefficients are randomly selected from F. The public key is composed of the map $P = S \circ F \circ L : F^n \to F^m$. The process of signature and verification is the same as in Sect. 2.2, so there is no longer a description.

3 Experiment

Ethereum offers several open source projects on github, Go-ethereum [18] project based on Go is currently the most widely used Ethereum Geth client. It provides an interactive command console that includes all functional interfaces, such as building a private chain, mining, deploying smart contracts and so on.

This experiment made full use of the convenience brought by open source thinking. We replaced ECDSA in the Go-ethereum project by Rainbow algorithm. Finally, we tested the feasibility by building a private chain.

Each Ethereum's user has a pair of secret keys, one public and one private. By using Rainbow algorithm, users can use the public key hash as address of the account to identify different users. When the transaction is sent, in order to prove that the transaction is actually carried out by sender itself, the sender must sign the transaction content with its own private key, while other recipients can verify the legality of the signature. On the one hand, this can guarantee that the user's account is not impostor. On the other hand, the senders can't deny the transaction they have signed.

Next, this paper will introduce the experimental environment, the implementation of Rainbow algorithm API and the specific application of it in Ethereum's account generation and transaction transmission and the experimental results.

3.1 Environmental Environment

All the experiments in this paper are executed on a PC with an Intel Core i5 processor and 8GB of RAM. The Operating System is Windows 10 Professional, 64-bit. As for the software, we take Eclipse 4.8.0 as IDE and *go* as the development language.

3.2 API of Rainbow Signature Algorithm

Since Rainbow signature algorithm essentially performs matrix operations on a finite field, we need to complete the implementation of the library that the Rainbow algorithm relies on:

- Element operations on the specific finite field $GF(256)$, such as addition and multiplication.
- Matrix operations on the specific finite field $GF(256)$, such as transposition, inversion and so on.

On this basis, the main functions of Rainbow signature algorithm are implemented: key generation, signature, signature verification, address generation and recover public key. The external interfaces are shown in Table 1.

Table 1. Rainbow Algorithm External Interfaces and its Introduction

Function	Introduction
*func GenerateKey() (*PrivateKey)*	Generate a public-private key pair
*func SignMPKC(hash []byte, prv *mpkc.PrivateKey) ([]byte, error)*	Calculate signature by private key and hash of message
func VerifySignatureMPKC(pubkey, mpkc.PublicKey,hash []byte,, signature []byte) bool	Verify signature according to public key, hash of message and signature
func EcrecoverMPKC(hash,sig []byte), ([]byte, error)	Recover public key according to hash of message and signature
func PubkeyToAddressMPKC(pub, mpkc.PublicKey) common.Address	Generate an address based on the public key

3.3 Account Generation Process

When creating a new account, users first need to enter a passphrase. Then the program internally generates a public-private key pair by calling the *GenerateKey()* function of Rainbow algorithm. After the public key is hashed, it is used as an account address, and the public key-address pair is stored in public key storage server, so that the public key is queried according to address information. The private key is encrypted by using passphrase as a password of AES-CTR algorithm. Finally, the account address and randomly generated parameters in the encryption process are written to the wallet file. Among them, Mac values are used to verify the legitimacy of passphrase for preventing others tampering when decrypting. It actually has an effect of signature. This detailed procedure is shown in Fig. 3.

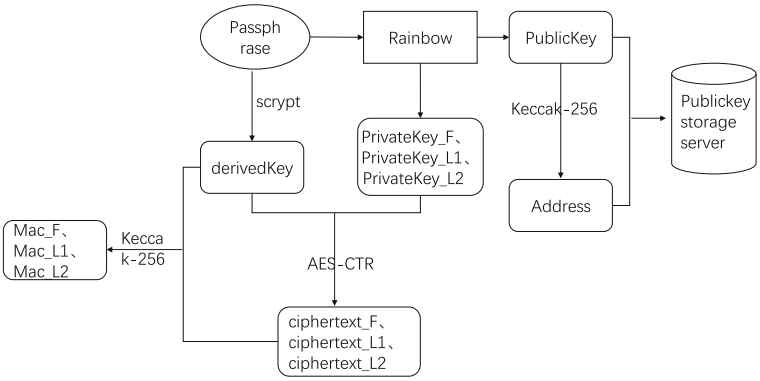

Fig. 3. Account Generation Process.

3.4 Transaction Transmission Process

When a transaction is sent, in order to prove that the transaction was actually carried out by sender itself, the sender must sign this transaction with its own private key. When signing a transaction, first call RLP encoding this transaction, then perform a Keccak-256 hash; next, call *SignMPKC()* function of Rainbow algorithm to sign the hash value of transaction; after that, the signature and sender address are respectively encapsulated into this transaction. This procedure is explained by Fig. 4. Hereafter, the sender broadcasts this signed transaction to each node in the network. When receiving a transaction, node can index the corresponding public key according to address information contained in this transaction, then call *VerifySignatureMPKC()* function of Rainbow algorithm to verify the correctness of signature according to public key. Finally, this transaction is recorded into the blockchain through Proof Of Work (POW) consensus mechanism.

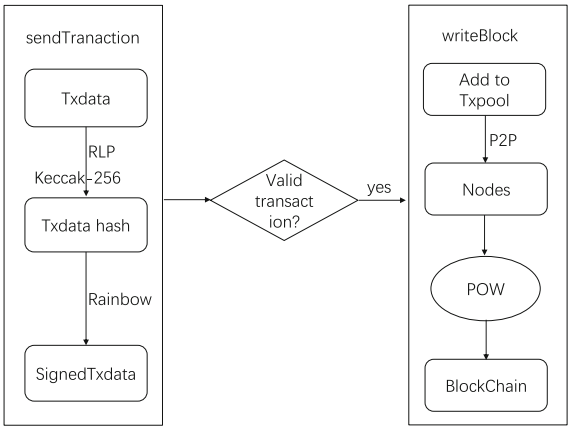

Fig. 4. Transaction Transmission Process.

Here, it should be noted that blockchain nodes need sender's public key when verifying the transaction's signature. So far, there is no standard method of publishing public key. In general, public key is placed in transaction data and sent to the network along with the transaction. In Bitcoin, the signature and public key are combined as a signature script that is a part of the transaction. However, ECDSA used by Bitcoin and Ethereum has a very peculiar nature: public key can be derived from the hash value and signature of transaction. Therefore, Ethereum transactions only contain signature part, and then algorithm is used to derive public key before verifying correctness of the signature.

Since Rainbow algorithm doesn't have the nature as ECDSA and its public key is relatively large, we have added a public key storage server to manage the public key-address pairs information of all users. In this way, when a transaction is sent, the public key is not directly sent out as the part of transaction, but the address generated by public key is encapsulated in the transaction. After that, when verifying the validation of signature, they only need to request the public key storage server for obtaining public key according to address information provided by transaction, then verifying signature.

4 Results

This experiment verifies the correctness of result by building a private chain using the runned geth client. The experimental result shows that after replacing the ECDSA with Rainbow signature algorithm, it does not affect the normal use of its original functions, such as creating an new account, sending a transaction and so on. Moreover, since the Rainbow algorithm is a post-quantum cryptographic algorithm, the Rainbow-based blockchain in the future will be able to resist the attack of quantum computers.

The different choices of parameters o_1, v_1, o_2 in Rainbow algorithm will result in different lengths of public key, private key and signature. The security level

Table 2. The Key and Signature Size of ECDSA and Rainbow Algorithm

Security level (bit)	Algorithm	Private key size (kB)	Public key size (kB)	Signature size (byte)
80	ECDSA	0.010	0.020	40
	Rainbow(13, 17, 13)	19.1	25.1	43
100	ECDSA	0.013	0.026	52
	Rainbow(16, 26, 17)	45.0	59.0	59
128	ECDSA	0.016	0.032	64
	Rainbow(21, 36, 22)	101.5	136.1	79
192	ECDSA	0.024	0.048	96
	Rainbow(34, 63, 34)	434.5	582.9	131
256	ECDSA	0.032	0.064	128
	Rainbow(46, 85, 47)	1073.1	1463.1	178

will vary greatly. In the experiment, this paper refers to several parameter recommendation schemes given by Professor Petzoldt [19] on the finite field $GF(256)$, and compares the key and signature sizes of ECDSA and Rainbow algorithm under different security levels. The specific data is shown in Table 2.

It can be seen from the table that private key and public key of Rainbow signature algorithm are very large. In order to reduce the size of public key, this experiment drew on the special public key construction method in the cyclicRainbow signature scheme [20], and compressed the original Rainbow's public key. The results of public key compression are shown in Table 3.

Table 3. The Key and Signature Sizes of Three Algorithms

Security level (bit)	Algorithm	Private key size (kB)	Public key size (kB)	Signature size (byte)
80	ECDSA	0.010	0.020	40
	Rainbow(13, 17, 13)	19.1	25.1	43
	cyclicRainbow(13, 17, 13)	19.1	10.4	43
100	ECDSA	0.013	0.026	52
	Rainbow(16, 26, 17)	45.0	59.0	59
	cyclicRainbow(16, 26, 17)	45.0	21.7	59
128	ECDSA	0.016	0.032	64
	Rainbow(21, 36, 22)	101.5	136.1	79
	cyclicRainbow(21, 36, 22)	101.5	47.3	79
192	ECDSA	0.024	0.048	96
	Rainbow(34, 63, 34)	434.5	582.9	131
	cyclicRainbow(34, 63, 34)	434.5	185.4	131
256	ECDSA	0.032	0.064	128
	Rainbow(46, 85, 47)	1073.1	1463.1	178
	cyclicRainbow(46, 85, 47)	1073.1	458.3	178

Apart from the size of key and signature, the efficiency of signature algorithm plays an essential role in the blockchain technologies. The ECDSA used in Ethereum has reached a security level of 256 bits. Therefore, this paper compared the signature time and verification time of ECDSA and Rainbow algorithms under the same level. The results are shown in Table 4.

Table 4. The Signature and Verification Time of ECDSA and Rainbow Algorithm

Security Level (bit)	Algorithm	Sign time (ms)	Verify time (ms)	Post-quantum?
256	ECDSA	0.25	0.35	no
	Rainbow(46, 85, 47)	204.78	28.66	yes

As can be seen from the table, the signature verification time of the Rainbow algorithm is obviously much higher than the ECDSA algorithm. At this point, an apples-to-apples comparison of operational speed should't be. The operational speed of two algorithms is shown here for reference only. Nevertheless, regardless of speed, the main selling point of our scheme is its reliance on different computational problems from those used in other branches of cryptography. Considering future attacks of quantum computers, we can sacrifice some time and space in exchange for the security of blockchain.

5 Conclusion

In this paper, we explored the application mode and method of multivariate public key cryptosystem in Ethereum platform. We realized the replacement of the original signature algorithm with *multivariate public key cryptosystem-rainbow scheme* in Ethereum, and verified the feasibility of this scheme by building a private chain. It shows that after replacing the Ethereum's ECDSA with Rainbow signature algorithm, it does not affect the normal use of its functions.

Our design demonstrates that the combination of dedicated multivariate signature scheme and blockchain technologies. Our scheme solves the security problem that blockchain cannot resist quantum computer attacks, and provides more secure and efficient underlying support for the application developed with the blockchain technologies in the future.

Funding Information. This work was supported by National Key R&D Program of China No. 2017YFB0802000.

References

1. Jun, Z., Zhang H.-N., Tang, Y., Li, L.: Blockchain Technical Guide. China Machine Press (2016)
2. Swan, M.: Blockchain: Blueprint for a New Economy. OReilly Media Inc., Sebastopol (2015)
3. Satoshi, N.: Bitcoin: a peer-to-peer electronic cash system (2009). http://www.bitcoin.org/pdf
4. Ethereum White Paper. A next-generation smart contract and decentr-alized application platform (2015). http://github.com/ethereum/wiki/wiki/WhitePaper
5. HYPERLEDGER (2016). http://www.hyperledger.org/BLOCKSTREAM
6. Yong, Y., Fei-Yue, W.: Blockchain: the state of the art and future trends. Acta Automatica Sin. **42**(4), 481–494 (2016)
7. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Secur. **1**(1), 36–63 (2001)
8. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J.Comput. **26**(5), 1484–1509 (1997)
9. Divesh, A., Troy, L.: Quantum attacks on Bitcoin, and how to protect against them (2017). arXiv:1710.10377v1quant-ph
10. Quantum-Resistant Ledger (2017). https://github.com/theQRL/QRL

11. Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R.: NISTIR 8105 Report on Post-Quantum Cryptography, NIST, 10.6028/NIST.IR.8105 **5** (2016)
12. Post-Quantum Cryptography Round-1-Submissions, NIST (2017). https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions
13. Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.): Post Quantum Cryptography. Springer, Heideberg (2009). https://doi.org/10.1007/978-3-540-88702-7
14. Stark, J.: Making sense of blockchain smart contracts (2018). https://www.coindesk.com/making-sense-smart-contracts/
15. Ding, J., Gower, J.E., Schnidt, D.S.: Multivariate Public key Cryptosystems. Advances in Information Security. Springer, Boston (2006)
16. Patarin, J.: The oil and vinegar signature scheme. In: Dagstuhl Workshop on Cryptography (1997)
17. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (2005). https://doi.org/10.1007/11496137_12
18. Buterin, V.: Ethereum go-ethereum source code [EB/OL] (2018). https://github.com/ethereum/go-ethereum
19. Petzoldt, A.: Selecting and Reducing Key Sizes for Multivariate Cryptography (2013)
20. Petzoldt, A., Bulygin, S., Buchmann, J.: CyclicRainbow – a multivariate signature scheme with a partially cyclic public key. In: Gong, G., Gupta, K.C. (eds.) INDOCRYPT 2010. LNCS, vol. 6498, pp. 33–48. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17401-8_4