Editable Blockchain Scheme Based on threshold changeable secret sharing

Ou Ruan*[1], Xin Jiang[2]

Hubei University of Technology,China,

[1]ruanou@hbut.edu.cn;[2]102101041@hbut.edu.cn

Abstract. The blockchain has the characteristics of decentralization and tamper-proof and has received extensive attention. As the application of blockchain becomes more and more extensive, problems also follow. Once the data is stored in the blockchain, it cannot be tampered with. Therefore, the editable blockchain scheme under certain conditions has broad application prospects. This paper proposes a threshold changeable editable blockchain scheme based on bivariate polynomial chameleon hash secret sharing. The variable threshold makes the scheme adapt to various dynamic environments.

Keywords: Editable Blockchain; threshold changeable secret sharing; chameleon hash.

1 Introduction

Blockchain technology has practical editing requirements in data updates [1]. Blockchain cannot tamper with this the characteristics limit the wide application of blockchain in many fields, there may be such a scenario of the blockchain-based electronic case system: after the medical institution successfully chained the patient's data. When patients need to update the stored electronic medical record files, it is difficult to solve because of the immutability of the blockchain. On the one hand, some illegal data and redundant data are stored on the blockchain [2]; on the other hand, malicious or erroneous data stored in the chain can cause resource consumption [3] and increase the management burden.

Ateniese et al. [4] proposed the concept of modifiable blockchain, which is a licensed blockchain scheme based on chameleon hashing [5]. When editing blocks, the chameleon hash function with trapdoor is used, so the collision of the chameleon hash function can be effectively calculated and the hash consistency can be maintained after any modification. However, the scheme modifies the data structure of the block head and has poor applicability. Derler et al. [6] proposed an editable blockchain scheme based on ciphertext-policy attribute-based encryption (CP-ABE) algorithm. The scheme is mainly used in the field of virtual currency, and the threshold key cannot resist the collusion attack of users with different attributes.

K. Subramaniyam et al. (eds.), *Proceedings of the 3rd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2024)*, Atlantis Highlights in Computer Sciences 21,
https://doi.org/10.2991/978-94-6463-419-8_27

The use of chameleon hashing technology to implement editable blockchains is an important innovation. If the trapdoor of chameleon hash is controlled by a user, there may be a problem of centralization. Fan et al.[7] proposes an editable blockchain scheme EB-SC based on (t,n) Shamir secret sharing. The trapdoor of chameleon hash is shared with multiple users, and multiple users work together to complete the editing of block history. However, the threshold value in the scheme is unchangeable, which can not be well applied in the application of a dynamic environment. Harn et al.[8] propose a novel idea to construct a dealer-free and non-interactive TCSS based on pairwise keys between users to achieve the variability of the threshold.

2 PROPOSED SCHEMES

2.1 Detailed scheme

- **Private key sharing**

1. $(ppara, spara) \leftarrow Setup(1^{\lambda})$: The Leader node runs the system parameter initialization function. Enter the safety parameter λ, randomly generate a large prime p. G is a multiplicative group of order p, g is a generator of the multiplicative group G, and GF is a finite field of order p. There exists a chameleon hash function H. The leader selects a random number d as the private key of the chameleon hash function, which is denoted as sk_l, and calculates the public key $pk_l = g^d$. The function generates $ppara = (Z_p, G, GF, p, g, pk_l, H)$ and $spara = (sk_l)$. Set the unique ID_i for the entire network for each user U_i of the system.
2. $(s_1, ..., s_i, ..., s_n) \leftarrow Share(d)$: The private key is shared in groups through the symmetric bivariate polynomial to achieve threshold changeable chameleon hash key management.

   Step 1: The Leader gets a symmetric bivariate polynomial $f(x,y) = d + a_{1,0}x + a_{0,1}y + a_{2,0}x^2 + a_{1,1}xy + a_{0,2}y^2 + ... + a_{t-1,0}x^{t-1} + a_{t-2,1}x^{t-2}y + ... + a_{t-1,t-1}x^{t-1}y^{t-1} \bmod p$, where the private key, d, is $d = f(0,0)$. $a_{i,j} \in GF(p)$, and $a_{i,j} = a_{j,i}$, $s_i = f(ID_i, y)$.

   Step 2: Leader distributes shars s_i to the user U_i, $i = 1,2,3,...,n$.

- **Reconstruct the private key**

3. $(req) \leftarrow ReqEdit(m, m')$: The user node in the system submits an edit request, takes the edited original content m and the edited content m' as input, and the function generates an edit request req, which is broadcast to the whole network.

4. $(d) \leftarrow PrivateKeyReconstruction(U_1, U_2, U_3, ..., U_{t'})$: If there are t' users, U_l, $l = 1, 2, ..., t'$, agree to the edit request *req* and participate in private key reconstruction. The original threshold t becomes a new threshold t'.

Step 1: Shared keys are generated between users by using shares s_i. For example, shareholder, U_i, substitudes ID_j into the share s_i, $s_i = f(ID_i, y)$, to get $k_{i,j} = f(ID_i, ID_j)$, and shareholdser, U_j can get $k_{j,i} = f(ID_j, ID_i)$, shared key $k_{i,j} = k_{j,i}$.

Step 2: Every user participated in secret reconstruction uses share s_i to compute Largrange Component(LC) c_i, $c_i = f(ID_i, 0) \prod_{l=1, l \neq i}^{t'} \frac{-ID_l}{ID_i - ID_l} \bmod p$. Each user U_i generate and public v_i, $v_i = c_i + \sum_{l=1, l \neq i, ID_l < ID_i}^{t'} k_{i,l} - \sum_{l=1, l \neq i, ID_l > ID_i}^{t'} k_{l,i}$, based on the shared key $k_{i,j}$.

Step 3: The private key d can be reconstructed by computing $d = \sum_{i=1}^{t'} v_i$.

- **Edite**

5. $(r') \leftarrow Edit(m, m', d, r)$: The input of the function is composed of the edited original content m and the edited content m', the private key d and the original random number r of the chameleon hash. The output is the new random number r' generated by the chameleon hash, where $r' = (m - m' + dr) / d$, $H(m, r) = H(m', r')$.

- **Example**

— Private key sharing stage: The Leader choose a symmetric bivariate polynomial $f(x, y)$ over $GF(19)$ as $f(x, y) = 8 + 2x + 2y + 3xy \bmod 19$,where the private key of chameleon hash key $d = 8$. Leader share the private key d to three users, U_i, $i = 1, 2, 3$. $f(x_i, y)$ is a polynomial in y with degree 1 and in x with degree 1. Therefore, the original threshold t is 2.

$$s_1 = f(1, y) = 8 + 2 + 2y + 3y \bmod 19,$$
$$s_2 = f(2, y) = 8 + 4 + 2y + 6y \bmod 19,$$
$$s_3 = f(3, y) = 8 + 6 + 2y + 9y \bmod 19.$$

Send s_i to the user node.

— Reconstruct Private key stage: It is assumed that all three users participate in the secret reconstruct phase, the threshold is changed from the original threshold $t = 2$ to the new threshold $t' = 3$.

$$c_1 = f(1,0)\prod_{l=1,l\neq i}^{t'}\frac{-ID_l}{ID_i - ID_l} = 10\frac{(-2)(-3)}{(1-2)(1-3)} = 10\cdot 6\cdot(2^{-1} \bmod 19) = 10\cdot 6\cdot 10 \bmod 19 = 11$$

$$c_2 = f(2,0)\prod_{l=1,l\neq i}^{t'}\frac{-ID_l}{ID_i - ID_l} = 12\frac{(-1)(-3)}{(2-1)(2-3)} = 12\cdot 3\cdot((-1)^{-1} \bmod 19) = 12\cdot 3\cdot 18 \bmod 19 = 2$$

$$c_3 = f(3,0)\prod_{l=1,l\neq i}^{t'}\frac{-ID_l}{ID_i - ID_l} = 14\frac{(-1)(-2)}{(3-1)(3-2)} = 14\cdot 2\cdot(2^{-1} \bmod 19) = 14\cdot 2\cdot 10 \bmod 19 = 14$$

User node generates shared key by s_i :

$$k_{1,2} = k_{2,1} = f(1,2) = f(2,1) = 20;$$
$$k_{1,3} = k_{3,1} = f(1,3) = f(3,1) = 25;$$
$$k_{2,3} = k_{3,2} = f(2,3) = f(3,2) = 36;$$

User node computes v_i :

$$v_1 = c_1 + k_{1,2} + k_{1,3};$$
$$v_2 = c_2 - k_{1,2} + k_{2,3};$$
$$v_3 = c_3 - k_{1,3} - k_{2,3};$$

Recover private key d :

$$d = \sum_{i=1}^{t'} v_i = v_1 + v_2 + v_3 = 8;$$

The threshold is changed from the original threshold $t = 2$ to the new threshold $t' = 3$.

2.2 Security analysis

- **Theorem 1 Private key can be recovered with any t or more than t shares.**
  There are t' ($t' > t$) shareholder participating in private key reconstruction, each shareholder can compute $c_i = f(ID_i,0)\prod_{l=1,l\neq i}^{t'}\frac{x - ID_i}{ID_i - ID_l} \bmod p$ based on Lagrange interpolation theorem, the polynomial of degree $t-1$ with respect to x can be recovered by summing c_i , where $f(x,0) = \sum_{i=1}^{t'} f(ID_i,0)\prod_{l=1,l\neq i}^{t'}\frac{x - ID_i}{ID_i - ID_l} \bmod p$ and private key $d = f(0,0)$. So , the private key d can be recovered when the number of participants in the secret reconstruction t' exceeds the threshold t .

- **Theorem 2 Private key can not be recovered with fewer than t shares.**
  When there are $t-1$ shareholders participating in private key reconstruction, they can generate $t(t-1)$ equations. Note that these equations are not linearly independ-

ent, because $t-1$ shareholders can also establish $C_2^{t-1}=\frac{(t-1)(t-2)}{2}$ pairs of keys . Therefore, we can conclude that for $t-1$ shareholders can generate $t(t-1)-\frac{(t-1)(t-2)}{2}=\frac{(t+2)(t-1)}{2}$ linear independent equations and each equation is denoted by the coefficient $a_{i,j}\in GF(p)$ of the bivariate polynomial $f(x,y)$. In the $t-1$ symmetric bivariate polynomial $f(x,y)$, there are $\frac{t(t+1)}{2}$ different coefficients $a_{i.j}$, and the share s_i held by $t-1$ shareholder produces $\frac{(t+2)(t-1)}{2}$ linear independent equations. Because $\frac{t(t+1)}{2}>\frac{(t+2)(t-1)}{2}$, therefore, the $t-1$ share s_i provided by the $t-1$ sharholder who agrees to modify cannot recover the private key d.

- **Theorem 3 Threshold changeable property, the original threshold t is changed to a new threshold t', where $t\le t'$.**
  When there are t' shareholder participating in private key reconstruction, each shareholder can establish $t'-1$ shared key with other shareholders. Since each release value v_i is the sum of the c_i and +/- values of the $t'-1$ key and the other shareholders, it is not possible to obtain each c_i from each release value v_i. Further, it is necessary to add all the released values, $\sum_{i=1}^{t'}v_i$, and all shared keys can be cancelled correctly. Therefore, if the released value v_i is less than the threshold t', the private key d cannot be recovered.

3 Conclusions

We propose an editable blockchain scheme based on variable threshold of bivariate polynomial. The chameleon hash is used securely through random grouping and key sharing algorithms. At the same time, the variable threshold makes the scheme adapt to various dynamic environments.

References

1. LUO Bin,WEN Jinming,WU Yongdong,CHEN Jie. (2023). State of the Art and Challenges of Redactable Blockchain[J].Journal of Cyber Security, 8(4), 62-84.
2. Matzutt, R. , Hiller, J. , Henze, M. , Ziegeldorf, J. H. , & Wehrle, K. . (2018). A quantitative analysis of the impact of arbitrary blockchain content on bitcoin.

3. B, A. D. A. , A, S. S. K. , & B, R. J. . (2019). Mof-bc: a memory optimized and flexible blockchain for large scale networks. Future Generation Computer Systems, 92, 357-373.
4. Ateniese, G. , Magri, B. , Venturi, D. , & Andrade, E. . (2017). Redactable Blockchain – or – Rewriting History in Bitcoin and Friends. 2017 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE.
5. Ateniese, G. , & Medeiros, B. D. . (2005). On the Key Exposure Problem in Chameleon Hashes. Proceedings of the 4th international conference on Security in Communication Networks. Springer-Verlag.
6. Derler, D. , Samelin, K. , Slamanig, D. , & Striecks, C. . (2019). Fine-Grained and Controlled Rewriting in Blockchains: Chameleon-Hashing Gone Attribute-Based. 26th Annual Network and Distributed System Security Symposium, NDSS 2019.
7. Fan, S., & Chen, Y. (2022). Editable Blockchain Scheme Based on Shamir Chameleon Hash Secret Sharing. IEEE 6th Information Technology and Mechatronics Engineering Conference, ITOEC 2022, 1125–1128. https://doi.org/10.1109/ITOEC53115.2022.9734554.
8. Harn, L., Hsu, C., Xia, Z., Xu, H., Zeng, S., & Pang, F. (2023). Simple and efficient threshold changeable secret sharing. Journal of Information Security and Applications, 77, 103576. https://doi.org/10.1016/j.jisa.2023.103576