Research on Anonymous credential based medical blockchain

Pan Xu, Shaofei Wu*
School of Wuhan Institute of Technology University, Wuhan, China
*Corresponding author: 04005047@wit.edu.cn

Abstract. For the current medical block chain, the access rights of electronic medical records and agent encryption are mainly controlled by permission contract. The way of protecting electronic medical records realizes the sharing of medical data, but neglects the management and protection of user identity. In this paper, this article studies the privacy protection technology of the blockchain, combined with the stealth address and zero-knowledge proof technology, and proposes a plan to weigh anonymity and security to protect the privacy of users in the medical blockchain.

1. Introduction
With the development of digital currency represented by Bitcoin, the block chain technology has attracted more and more attention. The immutability and traceability of the blockchain provide a new 005047on medical block chain [1, 2, 3]. However, there is no suitable identity authentication system in these schemes, and the management and protection of user identities are ignored, so that the identity of patients cannot be effectively protected. Traditional identity authentication is based on PKI and uses public key cryptography to exchange secure information in the network. Some studies use blockchain technology to build a decentralized certificate issuance and verification center to realize identity management and authentication [4, 5]. However, the certificate-based authentication method cannot meet the authentication requirements in various scenarios in real life. Therefore, some studies [6, 7] have abandoned the concept of certificates, and realized the block by binding the offline identity and the digital identity on the blockchain. The identity management on the chain meets the authentication requirements in different situations. However, these schemes use identity identifiers to represent a user, just like traditional blockchain systems use public keys to represent users, there is still a risk of user privacy leakage. Therefore, the anonymity of blockchain has also become a research hotspot [8, 9, 10]. Using cryptography to hide the user’s identity can achieve perfect privacy, and at the same time, it also brings the danger of unsupervised identity abuse. Combined with the actual situation of the current social medical level, the abuse of anonymous identities will inevitably lead to the waste of medical resources and fraud. At present, some supervised anonymous mechanisms have been proposed. Regulators can track anonymous identities through ring signature technology, thereby preventing the abuse of anonymous identities [11]. However, in these schemes, the supervisory authority can only review after the transaction, and cannot avoid problems in time. Therefore, this article combines the stealth address [12] and zero-knowledge proof technology to study the privacy protection technology of the blockchain, weighs anonymity and security, and proposes a plan to protect the privacy of patients in the medical blockchain.
2. Blockchain based anonymous credential scheme

This scheme relies on the W3C standards providing DID [13] and Verifiable Claim [14]. Decentralized Identifiers (DIDs) are a new type of identifier for verifiable decentralized digital identities. These new identifiers are designed to enable the controller of a DID to prove control over it and to be implemented independently of any centralized registry, identity provider, or certificate authority. A verifiable credential can represent all of the same information that a physical credential represents. The addition of technologies, such as digital signatures, makes verifiable credentials more tamper-evident and more trustworthy than their physical counterparts. The verifiable credential contains the user's DID, so the verifiable credential can be linked by the DID. Therefore, if these credentials are collected, some characteristics of the user can be analyzed, thereby threatening the user's privacy. This paper proposes an anonymous credential, so that the verifiable credential does not contain the user's DID, and the user can prove the association between the two credentials through zero knowledge.

The system includes four kinds of roles: user, KYC, credential issuer and verifier. The user first registers an identity ID, trusted authentication organization in the block chain to generate an ID certificate for the user. The user can request the business certificate from other voucher publishing institutions through the ID certificate. The user shows the ID certificate and business certificate and verification parameter to the validator, who verifies the integrity and authentication of the certificate. The scheme structure is shown in Fig. 1.

![Figure 1. Anonymous Credential Scheme.](image)

2.1. Initialization

Initialize the common parameter \( \{\mathbb{G}, q, g, H(\cdot), h\} \), where \( \mathbb{G} \) the is the elliptic curve group, and the \( q \) is the prime number representing the elliptic curve. The \( g \) is the basepoint of group \( \mathbb{G} \). \( H(\cdot) \) is an anti-collision hash function, \( h \) is a random point of group \( \mathbb{G} \).

2.2. Registration

Refer to DKSAP Privacy Address Protocol [12], two secret keys are generated for the user, which are used as payment key and scanning secret key respectively, so that the user can use his own stealth address in the credential, and can calculate the secret key of the stealth address through the open blind factor, reducing the burden of storing key. User registration is to create an account and store it on block chain. The fields in the account are shown in the Table 1.
Table 1. Three Scheme comparing.

| User ID | Payment public key | Scan public key |
|--------|--------------------|-----------------|
| \(d\)  | \(pk_p\)          | \(pk_s\)       |

2.3. Generating Credential

In this paper, the content of credential is divided into two parts, one is used to realize anonymous authentication called anonymous parameters, the other is used for business proof called business parameters, the fields in the credential are shown in Table 2. Anonymous parameters are generated by the user and verified by issuer. If the verification passes, the issuer generates credential for user. KYC knows the identity of the user, but the semi-trusted issuer does not, so the verification methods of the them are slightly different.

Table 2. Three Scheme comparing.

| Blind factor | Anonymous parameters | Business parameters |
|--------------|----------------------|---------------------|
| \(R\)        | \(pk_{st}\)          | \(C\) \(s_{user}\) |
| \(C\)        |                       | \(m\) \(d_{issuer}\)\(s_{issuer}\) |

2.3.1. Generating anonymous parameters. User calls the algorithm \(params(d, pk_p, sk_p, pk_s, sk_s, m)\) to calculate the user’s anonymous parameters and sends them to the issuer. The algorithm is as follows: User selects blind factor \(R = g^r\) in accordance with DKSAP calculates user’s stealth address by formula 1.

\[
pk_{st} = g^{H(pk_p)}pk_s
\]  

(1)

And, user calculates the private key of the stealth address according to the formula 2, and then calculates the ID’s Pedersen Commitment according to the formula 3.

\[
sk_{st} = H(R^{sk_p}) + sk_s
\]  

(2)

\[
C = h^{sk_{st}}g^d
\]  

(3)

Finally, user selects random number \(K = g^k\), then computes a signature of the blind factor, content, stealth address, and identity commitment by formula 4.

\[
s = k + H(R \parallel m \parallel pk_{st} \parallel C)sk_{st}
\]  

(4)

2.3.2. Generating ID credentials. KYC call algorithm \(generate1(d, pk_{st}, h^{sk_{st}}, C, s, m)\) to verify the anonymous parameters and generate ID credential. KYC is mainly to verify the user’s ID, and whether the commitment is computed by ID. The algorithm is as follows: KYC query user’s payment public key \(pk_p\) and scanning public keys \(pk_s\) on block chain, then use Zero-knowledge proof to prove that the user’s ID, interaction process as shown in Fig. 2.
Figure 2. Interaction of checking user’s identity.

and, KYC calculates user ID commitment by according to formula 3. If the result is equals to the commitment, KYC continues to verify the signature of anonymous parameters. KYC calculate the signature according to the formula 5 and compare it with the signature provided by the user.

\[ g^s = pk_{st} H(R || m || pk_{st} || C || s_{user} || b || d_{issuer}) K \]  

Finally, KYC selects random number \( K = g^k \), then compute a signature of the anonymous parameters, business content and KYC’s ID are signed by formula 6.

\[ s = k + H(R || m || pk_{st} || C || s_{user} || b || d_{issuer}) sk_{issuer} \]  

2.3.3. Generating business credentials. semi-trustable issuer call \( generate2(pk_{st}, pk_{st}', s, C, C', \Delta x) \) to verify the anonymous parameters and generate ID credential. The semi-trusted organization does not know the identity of the user, so it needs to use the identity certificate to verify the parameters provided by the user. Let \( C \) is the new commitment, \( C' \) is the commitment in ID credential, and \( \Delta x = sk_{st}' - sk_{st} \) is secretly calculated by user for verification the anonymous parameters. The algorithm is as follows: issuer verifies commitment and stealth address by formula 7 and formula 8 respectively.

\[ C + h^{\Delta x} = C' \]  
\[ pk_{st} + g^{\Delta x} = pk_{st}' \]  

Then, issuer verify the signature of anonymous parameters by formula 5, if anonymous parameters are correctly, issuer continues to generate credential the same as KYC.

2.4. Certificate validation

In the verification phase, the verifier verifies the integrity of the credentials, and then verify the relationship between the two credentials, that is, the certificates belong to the same identity. The verification process is as follows: According to the issuer ID on the certificate, query the issuer's public key on the blockchain, and then use the formula 9 to verify the signature.

\[ g^s = pk_{st} H(R || m || pk_{st} || C || s_{user} || b || d_{issuer}) K \]  

Then, verifier needs to verify the commitment, the stealth address, and the signature of anonymous parameter. The verification method is the same as that of a semi-trusted institution verifying user parameters.
2.5. Security analysis

2.5.1. Security of key. Firstly, user passes the zero-knowledge proof and the trusted organization for identity authentication, without revealing the user's secret key. When generating credentials, only the stealth address is required, and the user's key is not required. Secondly, the security of the stealth address is based on the discrete logarithm problem on the elliptic curve. The security of the stealth address ensures that the payment key $sk_{st}$ corresponding to the stealth address $pk_{st} = g^{sk_{st}}$ will not be cracked, that is, the attacker cannot derive the secret key $sk_{st}$ from the stealth address.

2.5.2. Unforgeability. If the verifier checks the parameters according to the protocol, the user's parameters cannot be forged by anyone. Let $\Delta x$ random oracles, and $C', pk_{st}$ are calculated according to formula 7 and 8 respectively. However, it is also necessary to sign anonymous parameters with a private address. The security of the stealth address was analyzed in the previous section, so the corresponding private key cannot be calculated here, so the formula 4 cannot be used to sign the anonymous parameters. At the same time, the signature cannot be calculated according to the formula 5.

2.5.3. Resist collusion. Assuming that the identities of two users are d and e respectively, in the stage of generating a credential by a semi-trusted institution, the user may generate a credential that uses the identity of another user, so that others can use the credential. The following equation holds if the user generates a commitment to another identity $d'$ by $\Delta x$.

$C(x, d)h^{\Delta x} = C(x', d') \Rightarrow h^{x+\Delta x}g^d = h^{x'}g^{d'} \Rightarrow h^{\Delta x} = h^{x'-x}g^{d'-d}$  \hspace{1cm} (10)

2.6. Anonymous analysis

Anonymity of this scheme is based on the security of stealth address and Pedersen commitment. User’s ID would not be known by semi-trustable issuer nor certificate.

Firstly, when a trusted organization generates credentials, it does not analyze its anonymous letter because it does not consider the problem of identity disclosure by a trusted organization. For the scheme of generating credentials by semi-trusted institutions, because the anonymous parameters are executed by the users themselves, the semi-trusted institutions cannot obtain the private key of the invisible address of the users, so the identity of the users cannot be deduced by enumerating all the identity trial formula 2. In addition, user provides verification $\Delta x$ to semi-trust organizations, which can not get the invisible address of users through the $\Delta x$, so they can not launch the identity of user.

Second, in verification phase, the user can verify that the user is the holder of the ID certificate through the authentication information on the ID certificate, and then submitting $\Delta x$ to verify that both the certificate and the business certificate are commitments to the same ID. Therefore, you can determine the user himself to use the certificate. The validator can not get any information related to the user's stealth address and user identity through the $\Delta x$, so it can ensure that the user's identity $d$ will not be leaked.

3. Anonymous credential based medical blockchain

In order to solve the lack of a suitable identity management solution for the current medical blockchain, this article introduces anonymous identity authentication to the medical block chain. In the supervised anonymous authentication scheme [11], user can selectively expose the attributes on credential. In the scheme, credential is mainly used for identity authentication, but in the medical blockchain, the certificate is also used in data exchange and other aspects. For example, when user use prescriptions to buy medicines in pharmacies, the staff of the pharmacy needs to confirm the identity of the user and the authenticity of the prescription. In order to verify, pharmacies and hospitals need to use the same attributes for identity authentication and generate prescription credential. In addition, in order to avoid
waste of resources, patients in hospitals need more authentication methods, while in pharmacies, only one contact information is needed. Therefore, in order to avoid exposing more identity information to the less secure entities such as pharmacies, this paper uses anonymous credentials for authentication. Next, the above scheme is implemented based on the WeIdentity [15].

3.1. User DID generation
WeIdentity creates an identity identifier for the user, only a pair of ECDSA secret keys can be generated by default, which are signed by the user to the certificate or transaction. Therefore, when creating identity identifiers for users, two pairs of secret keys are generated and stored in the DID Document public key list in order. The first public key is the payment public key and the second public key is the scanning public key. Generating user DID pseudo code is as follows.

1. did = createWeId() #create account (id, pk_payment)
2. (pk,sk) = createEcKeyPair() #create a key pair base secp256k1 curve
3. Did = addPublicKey(did,pk) #add scan public key to account
4. return did

3.2. The anonymous credentials CPT
CPT is a JSON-SCHAME to define the format of the credentials, the structure of the credential is shown in the Table 2. The anonymous parameter is used to prove that two credentials belong to the same identity, while the business field can set different fields according to the actual situation. For example, the identity credential contains the user's photo, contact information and other information, while the prescription credential contains information such as diagnosis results. The credential supports selective disclosure, that is, only the fields to be verified are displayed, and other fields are represented by hash.

3.3. Generate anonymous credentials
the user generates anonymous parameters locally, and then sends the anonymous parameters to the credential publisher, the authentication organization and the general certificate issuing organization generate the certificate according to the verification method in the above scheme. According to the user's DID documents, the mechanism extracts the user's payment public key and scanning public key in the background system, and randomly extracts the user's payment public key and scanning public key A blind factor call is selected to calculate the stealth address algorithm to obtain the public key of the user's stealth address, and then the proxy commitment algorithm is called to generate the commitment value of the user's identity ID. Finally, the organization uses the open value of the blind factor, the commitment value ID the user identity and the business value as the parameter to call the certificate contract to generate the user credentials.

3.3.1. anonymous parameters. The user generate anonymous parameters as follows:
1. input (d,sk_pay,sk_scan,m)
2. (r,R) = createEcKeyPair()
3. c = hash256(ECAlgorithms.referenceMultiply(sk_pay,m)) #compute DKSAP secret value
4. sk = c.add(sk_scan) # stealth address secret key
5. pk_g = referenceMultiply (g, sk) # stealth address public key
6. pk_h = referenceMultiply(h, sk) # blind factor of Pedersen Commitment
7. commit = pk_h.add(referenceMultiply(g, d)) # Pedersen commitment of d
8. (K, s) = Schnoor.sign(m,pk,commit,sk)
9. return (R,pk_g,commit,K,s,pk_h)

3.3.2. ID credential. KYC generates identity credentials for users according to the identity information submitted by user. The pseudo code is as follows:
1. input (d,R,pk_g,pk_h,commit,K,s,m)
2. params = (R,pk_g,commit,K,s) # anonymous params
3. did = getWeIdDocument(d)
4. commit2 = referenceMultiply(g, d).add(pk_h)
5. assert commit = commit2  # verify commitment of d
6. assert true = Schnoor.verifySign(R, pk_g, commit, K, s)  # verify signature of params
7. (K2, s2) = sign(params, m, dIssuer)  # signature of credential
8. credential = createSelectiveCredential(params, m, dIssuer, K2, s2)
9. return credential

3.3.3. Prescription credential. The user selectively disclosed the photo information in the ID credential, and the hospital verified the user's identity according to the photo, and then generated the prescription credential for the user.
1. input (credential, delta, R, pk_g, commit, K, s, m)
2. params = (R, pk_g, commit, K, s)
3. assert true = verify(credential)  # query issuer's public key and verify signature of credential
4. commit2 = credential.commit
5. assert commit2 = referenceMultiply(h, delta).add(commit)  # verify commitment
6. pk2 = credential.pk_g
7. assert pk2 = referenceMultiply(g, delta).add(pk_g)  # verify stealth address
8. assert true = verifySign(R, pk_g, commit, K, s)
9. (K2, s2) = sign(params, m, dIssuer)  # signature of credential
10. return (params, m, dIssuer, K2, s2)

3.4. Certificate verification
The user selectively disclosed the contact information in the ID credential, and the pharmacy clerk records the user's contact information. Then clerk verify that the ID credential and prescription credential belong to the same identity. The code for certificate verification is as follows.
1. input (credentials, delta)
2. for c in credentials
3. do
4. assert true = verify(c)
5. assert true = verifySign(c.params)  # verify signature of anonymous parameters
6. end
7. deltaCommit = credentials[0].commit - credentials[1].commit
8. assert deltaCommit = referenceMultiply(h, delta).add(commit)
9. deltaPk = credentials[0].pk_g - credentials[1].pk_g
10. assert deltaPk = referenceMultiply(g, delta).add(pk_g)

4. Conclusions
At present, the existing medical block chain research mainly realizes the sharing of medical data by controlling the access rights of electronic medical records and the way of proxy encryption to protect electronic medical records, but neglects the management and protection of user identity. Therefore, this paper introduces the distributed identity system into the medical block chain and propose the anonymous credential scheme is used to solve the problem of identity disclosure in certificate verification. Then analyze the security of the scheme and verify the feasibility of the scheme through the example of buying medicant by prescription.

References
[1] Liu Z, Weng J, Li J, Yang J, Fu C and Jia C 2016 Cloud-based electronic health record system supporting fuzzy keyword search Soft Computing 20 3243–55
[2] Zhang C, Li Q, Chen Z, Li Z and Zhang Z 2019 Medical chain: Alliance Medical Blockchain System ACTA AUTOMATICA SINICA 045 1495–510
[3] Sun J, Yao X, Wang S and Wu Y 2020 Blockchain-based secure storage and access scheme for electronic medical records in ipfs IEEE Access 8 59389–401
[4] Wang R, He J, Liu C, Li Q and Deng E 2019 A privacy-aware pki system based on permissioned blockchains 2018 ieee 9th international conference on software engineering and service science (icsess)
[5] Wang Z, Lin J, Cai Q, Wang Q, Zha D and Jing J Blockchain-based certificate transparency and revocation transparency IEEE Transactions on Dependable and Secure Computing PP 1–1
[6] Lee J H 2018 BIDaaS: Blockchain based id as a service IEEE Access PP 1–1
[7] Xiaoyang, Zhu, Youakim and Badr 2018 Identity management systems for the internet of things: A survey towards blockchain solutions. Sensors (Basel, Switzerland)
[8] Bonneau J, Narayanan A, Miller A, Clark J, Kroll J A and Felten E W 2014 Mixcoin: Anonymity for bitcoin with accountable mixes
[9] Ruffing T, Moreno-Sanchez P and Kate A 2014 CoinShuffle: Practical decentralized coin mixing for bitcoin European symposium on research in computer security (esorics)
[10] Lin L I and Jian-Hua Y 2013 Anonymous authentication mechanisms based on zero-knowledge proof Computer ence
[11] Wang Z, Fan G, Cheng L, An H, Zheng H and Niu J 2019 Supervised anonymous authentication scheme Journal of Software 030 1705–20
[12] Fan X 2018 Faster dual-key stealth address for blockchain-based internet of things systems
[13] Decentralized Identifiers on https://www.w3.org/TR/2020/WD-did-core-20201108/
[14] Verifiable Credentials Data Model on https://www.w3.org/TR/vc-data-model
[15] WeIdentity on https://fintech.webank.com/weidentity