This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI

10.1109/ACCESS.2020.3016986, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.DOI

**An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks**

**FEIHONG YANG1,2, YING WANG1,2, CHUNLEI FU1,2, CHUNQIANG HU1,2 (MEMBER, IEEE), ARWA ALRAWAIS3, (Member, IEEE)**

* School of Big Data & Software Engineering, Chongqing University, Chongqing 400044, China (e-mail: {yfh, wang, clfu, chu}@cqu.edu.cn)

2 Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education(Chongqing University)

3 College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia. Email:a.alrawais@psau.edu.sa

Corresponding authors: Chunlei Fu and Chunqiang Hu.

This work was supported in part by National Key Research and Development Project(2018YFB2101202), in part by the Fundamental Research Funds for the Central Universities (2019CDQYRJ006), in part by National Natural Science Foundation of China (61702062), in part by Chongqing Research Program of Basic Research and Frontier Technology (cstc2018jcyjAX0334), in part by Key Project of Technology Innovation and Application Development of Chongqing (cstc2019jscx-mbdxX0044, cstc2019jscx-mbdxX0020), and in part by Overseas Returnees Innovation and Entrepreneurship Support Program of Chongqing (cx2018015), the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University under the research project (2019/01/1041).

 **ABSTRACT** In social networks, the personal attributes or hobbies of the users are exposed to the serverto establish the relationships. Service providers may store these information for commercial purpose or statistical analysis. Furthermore, the server may expose to external attacks, which may disclose users’ privacy information. In this paper, we present a hierarchical blockchain-based attribute matching scheme, which realizes privacy-preserving attribute matching under multiple semi-trusted servers. The scheme employs CP-ABE and bloom filter to satisfy the requirements of the users to make friend discovery, and reduces the computation cost of users by outsourcing decryption of CP-ABE. Besides, the hierarchical blockchain only implements the consensus and storage of matching results on the blockchain, while the complex calculations and a large amount of data storage are off-chain, which reduces the consumption of the blockchain and improves the operation efficiency. Finally, we prove the scheme can resist single point failure, collusion attack, internal attack and external attack, the experimental results demonstrate the proposed scheme is feasibility and efficiency.

 **INDEX TERMS** Attribute Matching, Friend Discovery, CP-ABE, Outsourcing Decryption, HierarchicalBlockchain, Social Network.

**I. INTRODUCTION**

OCIAL social networking provides an online platform Sto people to build social relationships with others, who have similar personal attributes such as age, home address, education background, etc. However, there is a risk of privacy leakage. Social network platforms may use user attributes for statistical, advertising or profit-making purposes [1], [2]. Such behavior will compromise users’ privacy, which affect users’ real life [3], [4]. For example, more than 25 gigabytes user information of an extramarital affair platform named Ashley Madison was leaked, including real name, home address and other information, and many users feared being publicly shamed [5], [6].

The basic idea of friend discovery in social networks is to

compare the similarity of attributes between two users with-out leaking users’ information. There are three categories to address the problem. The first category uses a set of attributes to summarize user’s information, and employs Private Set Intersection(PSI) or Private Set Intersection Cardinality(PSI-CA) to execute attribute matching [5], [7]–[9]. The second category employs vectors to represent user’s information, and the vector distance is calculated by dot product calcu-lation to represent social distance [10]–[12]. The third cat-egory takes advantage of Ciphertext Policy Attribute-Based Encryption(CP-ABE) and access control to achieve friend discovery [13]–[18]. Nevertheless, there are some issues in existing schemes:

Secure Multiparty Computing(SMC), homomorphic en-

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 1 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

cryption or vector-based dot product calculation are employed to execute the friend discovery. But these methods either consume massive computing resources or are vulnerable to statistical analysis attacks.

The schemes just considers the requirements of the initiator rather than the needs of both, which is not in line with the actual situation of making friends.

The existing schemes do not consider that social net-work platforms collude with some users to steal the privacy of others.

Additionally, most of the above solutions are based on point-to-point implementation. The server can reduce the computation cost of users during friend discovery, but the server is semi-trusted [19]. Blockchain has attracted much attention from scholars because of its characteristics of de-centralized, immutable and traceable. In recent years, it has been widely used in Internet of Thing, social network, crowd-sourcing, vehicle network and other fields [20]–[24].

To solve the problems of friend matching in social net-work, we propose an efficient and privacy-preserving friend matching based on blockchain in social networks. The con-tributions of the paper are summarized as follows:

We proposed an attribute matching mechanism based on the hierarchical blockchain and outsourcing CP-ABE for friend discovery in social networks, which can achieve the attribute matching in semi-honest social network platforms and reduce computing consumption of users.

The scheme satisfies the needs of the users by using CP-ABE and bloom filter, which is more appropriate to the actual situation of making friends.

A novel blockchain architecture is proposed to achieve the decentralization and auditability of friend matching in social network, which reduce storage consumption of the blockchain.

The rest of this paper is organized as follows. Section II discusses related work. Section III provides detailed system model, new blockchain architecture and threat model. The preliminaries are introduced in Section IV. And the proposed scheme is elaborated in Section V. Section VI and Section VII provides security analysis and performance evaluation for our scheme, respectively. In Section VIII, we summarize our research and future work.

**II. RELATED WORKS**

The most existing schemes consider users’ profiles as sets of attribute and measure similarity by calculating the intersec-tion of attributes. The Private Set Intersection(PSI) technique for achieving attribute matching was first proposed by Li et al. [7], which is based on Secure Multi-party Comput-ing(SMC). Yi et al. [5] proposed a profiles matching scheme based on homomorphic encryption in multiple social net-works, which provides profile privacy-preserving. The basic idea is to judge whether the dissimilarity of two users is less than the threshold given by the user.

The other methods regards user’s attributes as a vector, and the server calculates the dot product of two attribute vectors to obtain their similarity. Gao et al. [11] presented a multiple keys profile-matching protocol based on additive homomor-phism to calculate the dot product of two vectors. Then, some dot product schemes were proposed which abandon homomorphic encryption and have lower computing costs. Luo et al. [10] set weight for each attribute, i.e., the attribute set is represented as a matrix, and then used a lightweight confusion matrix transformation algorithm to protect user information. And Li et al. [25] mixed the each attribute vector with random noise to realize attribute matching.

CP-ABE has been widely used in social networks [26]–

1. Bethencourt et al. [29] first proposed CP-ABE in 2007. Waters [30] proposed a more efficient implementation of CP-ABE. Luo et al. [31] designed a friend discovery architecture based on CP-ABE and multiple attributed authority, which uses Shamir’s scheme to store the master key distributedly. Li et al. [32] proposed a point-to-point pre-matching scheme, using Bloom filter to reduce the computational load of users performing decryption of CP-ABE, and they elaborated how to establish a verifiable secure communication channel be-tween matched users. Based on this, Cui et al. [16] designed a receiver anonymous attribute matching scheme using CP-ABE and bloom filter. Qi et al. [15] combined searchable encryption with CP-ABE and proposed a friend discovery protocol with hidden attributes and fine-grained access con-trol.

Considering the unreliability of the centralized server, the researchers applied the blockchain technology to social networks architecture. Jiang et al. [33] and Gu et al. [34] proposed social network architectures that use the blockchain and smart contracts instead of centralized servers to provide social networking services. They mainly take advantage of the immutability of traditional blockchain and the fairness of smart contracts.

**III. SYSTEM OVERVIEW AND THREAT MODEL**

In this section, we formalize the system model, the hierarchi-cal blockchain architecture and the threat model.

**A. SYSTEM MODEL**

As depicted in Figure 1, the proposed system model consists of five entities, Trusted Authority(TA), Social Networking Platform(SNP), Users, Consortium Blockchain(BC), Proxy Cloud Computing Server(PCCS).

TA: A trusted third party is responsible for key genera-tion.

SNP: It’s a social networking platform with storage and computing capabilities such as Twitter and Facebook. It is the consensus node in the blockchain, which is called the miner in traditional blockchain networks. Under the chain, it provides social networking services and information storage services.

Users: The users would like to find friends in social net-works, which has the similarly attributes. The initiator

2 VOLUME 4, 2016

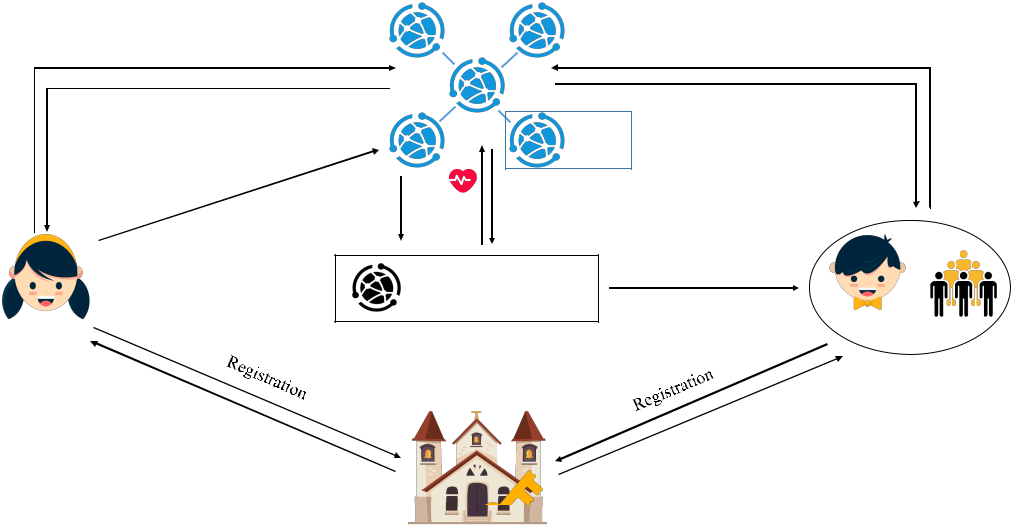
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Blockchain** | | | |  |  |  |
| Matching parameters |  |  |  |  |  | Matching parameters |  |  |
|  |  |  |  |  |  |  |
| Matching results |  |  |  |  |  | Matching results |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  | **SNP** |  |  |  |
| ABE CT |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| ABE CT |  | heart | | Sequential | |  |  |  |
|  | election | |  |  |  |
|  | beat | |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  | Partially |  |  |
|  |  | **Proxy Cloud** | | | | decrypted data | **…** |  |
|  |  |  |  |
|  | **Computing Server** | | | | |  |  |
|  |  |  |  |



**Bob**

**Alice**

**Trusted Authority**

**FIGURE 1.** System Architecture

Alice and the responder Bob are two users, who want to establish the social relationship each other.

BC: SNPs constitute the blockchain consensus nodes. Users have blockchain accounts, which do not partici-pate in the consensus and storage. Smart contract on BC determines whether two users can become friends.

PCCS: PCCS is selected from SNPs periodically via smart contract. It is responsible for partial decryption of CP-ABE to reduce users’ computation consumption.

In the system, each user has two attribute sets S and R, S is representing his/her private attributes, and R is representing his/her requirements for making friends. To solve that people have same attributes which describe them differently, for example, Alice and Bob both like to sing, but Alice inputs "sing" and Bob inputs "I like to sing". Therefore, we provide attribute space A. The elements in S and R are the hash values of the elements selected from A, i.e.,

* = fs1; :::; sng; R = fr1; :::; rmg;

8si; rj 2 Afh(att1); :::; h(attz)g

In our scheme, friend matching in the social network should meet the following two conditions.

1. The percentage of attribute similarity between two users is greater or equal to a certain threshold given by the user.
2. Both parties’ attributes meet each other’s requirements.

In addition, SNPs do not anything about the users’ attribute sets during the process of friend matching.

**B. HIERARCHICAL BLOCKCHAIN ARCHITECTURE**

One of our major contributions is proposed a hierarchical blockchain architecture. The performance of blockchain can

be obviously improved by the hierarchical design [35]. Fig-ure 2 shows the consortium blockchain architecture. The architecture divides the consortium blockchain into three parts: platform layer, user layer and off-chain.

The platform layer is composed of various SNPs and performs friend matching by smart contracts. All SNPs are consensus on the matching results and these records are stored in the blockchain. And smart contract periodically select a PCCS from SNPs to perform outsourcing encryption of CP-ABE.

A large number of users make up the user layer, which allows users to communicate casually with others, and they can make friends who meet specific attributes via the platfor-m layer.

Off-chain stores the data that means each platform has private databases to store users’ information. It indicates the blockchain stores matching records instead of all the users’ information to reduce storage consumption of each consen-sus node. Additionally, to improve the efficiency of smart contracts, complex cryptographic calculations are also under off-chain. Each operation on the blockchain is lightweight and low storage consumption which can effectively solve the issues of low efficiency of smart contract execution and large storage burden in the blockchain.

**C. THREAT MODEL**

In our system, we consider the following threat models:

Single Point Failure: Some SNPs’ servers may be fail-ure and stop providing services, or even be hijacked by attackers, which may cause system paralysis or users’ information leakage.

External Attack: The communication of each entity in the system may be subject to external attacks, such as

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 3 |

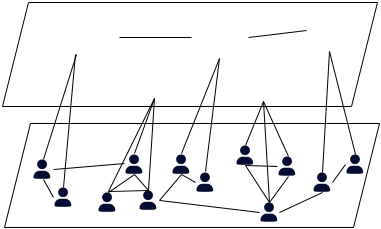
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

**Platform**  **layer** … 



**User**

**layer**

Decentralization and lightweight computing



|  |  |  |  |
| --- | --- | --- | --- |
| **Off-chain** | …… | Mass data |  |
| storage |  |
|  |  |  |
|  |  |  |  |
|  | ABE CT | CT |  |
| Complex calculations | |  |  |



**FIGURE 2.** Hierarchical Blockchain Architecture

replay attack, tampering attack, etc.

Internal Attack: Assume that entity is honest but curi-ous, but they want to infer the users’ attribute informa-tion from the obtained data. Besides, the SNP may give the wrong execution result and cause the match to fail.

Collusion Attack: SNPs may collude with some users to obtain other users’ personal information.

Based on the above threat model, our system design goals are listed as follows:

Data Confidentiality: Data privacy-preservation is the main purpose of the scheme. The users’ attribute in-formation should not be obtained by anyone under the above attacks.

Data Integrity: The data integrity should be provided in the communication between entities.

Low Storage Cost for Blockchain: The data in the blockchain is constantly added and cannot be deleted, and the data occupies the same amount of storage space of all consensus nodes. That is, if there are k consensus nodes, the storage consumption is k times of the tradi-tional scheme, so the amount of data on the blockchain should be as small as possible.

Low Computation Cost for Blockchain: Since the smart contract cannot be modified once it is deployed, and it is executed by all nodes, the algorithm in the smart contract should be simple, and the computation cost should be as low as possible.

Low Computation Cost for Users: Due to the low computing power of users, the system should have the low computation cost on the users’ side.

**IV. PRELIMINARIES**

In this section, we introduce the related cryptographic tech-nique and background knowledge.

4

**A. BILINEAR MAPS**

We define G1 as a multiplicative cyclic groups of prime order p, and the generator of G1 is g. Let a bilinear map, e : G1 G1 ! GT . And it has the following properties:

Bilinear: for all x; y 2 G1 and a; b 2 Zp, we have e(xa; yb) = e(x; y)ab.

Non-degeneracy: e(g; g) 6= 1.

Computability: for all x; y 2 G1 , there exists an algorithm to compute e(x; y).

1. **LINEAR SECRET SHARING SCHEMES(LSSS)**

A secret sharing scheme in multiple entities P is called linear if:

The shares for each entity form a vector based on Zp. The share-generating matrix M for scheme has ‘ rows

and n columns. And there is a function that associates the ith row of M to a party (i) and i is from 1 to ‘. If we want share a secret value s 2 Zp, we choose random numbers r2; ::; rn 2 Zp to generate a vector

!

~v = (s; r2; :::; rn). Then we can get the vector Mv with

‘ shares of the secret value s according to scheme .

!

That is, share (Mv)i belongs to party (i).

The secret s can be recovered as follows: Suppose that

is an LSSS according to access structure A. Define the authorized set S, and let I f1; 2; :::; ‘g be defined as I = fi : (i) 2 Sg. Then, there exist constants f!i 2 Zpgi2I such that, if f ig are valid shares of s according to scheme

P

. Then s recovered as s = i2I !i i. And s cannot be obtained for the unauthorized set S 2= A.

1. **CIPHERTEXT POLICY ATTRIBUTE-BASED ENCRYPTION**

Ciphertext Policy Attribute-Based Encryption (CP-ABE) is typical asymmetric encryption method which can provide fine-grained access control [29]. In CP-ABE, there is one trusted authority who is responsible for generation of a public key P K and a master key M K. The user can encrypt data based on P K and the designed access structure A. If the attribute sets of some users satisfy the above A, they can decrypt it with their private keys which are generated by trusted authority based on their attribute sets and M K.

1. **BLOCKCHAIN**

As the critical technology of Bitcoin, blockchain has the characteristics of distributed, immutable and traceable [36],

1. The blockchain composed of many nodes, maintaining the same database. There are mainly four concepts in the blockchain as shown below:

Transaction: It is an operation that changes the state of the blockchain.

Block: A block records a certain amount of transactions over a period of time. It represents the state change of the blockchain during this time.

Consensus: All nodes verify whether a transaction is valid and then reach an agreement.

VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

Chain: Each block stores both the hash value of the previous block and the hash value of the current block. By this means, each block has a close association with the previous block and the latter block and form a chain.

1. **SMART CONTRACT**

Smart contract is an immutable program running on the blockchain, which is called "chaincode" in Fabric, for con-sistency we use "smart contract" to represent chaincode in the paper. Once the smart contract is invoked, it is executed by a consensus node, and the execution result will be checked by all consensus nodes. After being successful deployed, the smart contract is not controlled by anyone, and is viewed by everyone via a unique address. Therefore, smart contract can be trusted by users, thereby supersede the traditional central server.

**V. THE PROPOSED SCHEME**

In this section, we present the proposed scheme. Table 1 illustrate the frequently used notations in the scheme.

**TABLE 1.** The Notations Frequently Used in the Scheme

|  |  |
| --- | --- |
| Notations | Description |
| ; ; z; t | Random numbers. |
| MK;PK | The master key and the public key of CP-ABE. |
| SK;TK | User’s secret key and transformation key of CP-ABE. |
| Ka | The symmetric secret key generated by Alice. |
| S; S0 | User’s private attribute set and the corresponding ciphertext. |
| R; R0 | User’s requirements for new friends and its ciphertext. |
| Psc | The unique address of the smart contract. |
| Pa; Pb | Users’ contact address in the social network. |
| P Ubc | The public key of the platform layer in the blockchain. |
| P Rbc | The private key of the platform layer in the blockchain. |
| CTabc | The ciphertext (from Alice to blockchain). |
| CTapc | The ciphertext (from Alice to PCCS). |
| CTpcb | The ciphertext (from PCCS to Bob). |
| CTbsc | The ciphertext (from Bob to the smart contract). |

**A. SYSTEM INITIALIZATION**

In the system initiation phase, TA generates various keys for the system, and transmits them to the corresponding entities via secure channel. The asymmetrical key (P Ubc; P Rbc) are generated for blockchain consensus nodes to communicate with users.

Then TA generates the master key M K and the public key P K. Transformation key T K and secret key SK are generated for each user based on their attribute sets S. The outsourcing decryption of CP-ABE is constructed based on scheme [38]. In addition, the smart contract for periodically selecting PCCS from SNPs is deployed on the blockchain.

1) CP-ABE INITIALIZATION

TA selects a security parameter and generates a universe description U = f0; 1g . Then it chooses a bilinear group

G1 of prime order p, a generator g. TA then selects two random numbers ; 2 Zp. Besides, it needs to choose a hash function F that maps U to G1. After that, the public key is published:

|  |  |
| --- | --- |
| P K = fg; g ; e(g; g) ; F g | (1) |

TA secretly stores the master key M K = (P K; g ) and exposes P K to SNPs and users, respectively.

2) USER REGISTRATION

The system sets the attribute space A that contains a large amount of attribute information such as gender, income, age, sports preference, education background, etc. When a new user joins the system, he/she first selects his/her attribute sets S and R from the attribute space A, then uploads S to TA to get transformation key T K and secret key SK.

TA picks a random number t0 2 Zp to create a temp key as:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| K0 = g g t0 ; L0 = gt0 ; | 8 | x | 2 | S K0 | = F (x)t0 |  |
|  |  | x | (2) |  |

Then it chooses another random number z 2 Zp and sets

* = t0=z. Then it generates the transformation key T K as:

P K; K = K01=z = g( =z)g (t0 =z) = g( =z)g t;

|  |  |  |
| --- | --- | --- |
| L = L01=z = gt0 | (3) |  |
| =z = gt; fKxgx2S = fKx01=zgx2S |  |

The T K is sent to the PCCS. The private key SK = fz; T Kg is sent to the user. Noted that when the user’s personal attribute set changes, T K and SK should be regen-erated.

3) SMART CONTRACT DEPLOYMENT

In the system initialization phase, the rules for electing PCCS need to be written into the smart contract. The PCCS is periodically elected from SNPs. The election algorithm is depicted in Figure 3. The smart contract containing the election algorithm is deployed on the blockchain, and all SNPs comply with the smart contract.

**B. USERS COMMUNICATION**

As demonstrated in Figure 4, when the initiator Alice wants to make friend discovery, she needs to execute three steps: matching preparation, smart contract initialization and data upload. The blockchain first checks the data. Then PCCS partially decrypts the data, and send it to the users who meet the requirement. If the receiver Bob also wants to execute friend discovery, he performs two steps: simple decryption and smart contract invocation.

1) MATCHING PREPARATION

In the preparation phase of matching, Alice first uses sym-metric key Ka to encrypt each element of her private attribute set Sa to get Sa0.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S0 | = | f | s0 | ; :::; s0 | j 8 | s | ai 2 | S | ; s0 | = E | Ka | (s | ) | g (4) |  |
| a |  | a1 | an |  | a | ai |  |  | ai |  |

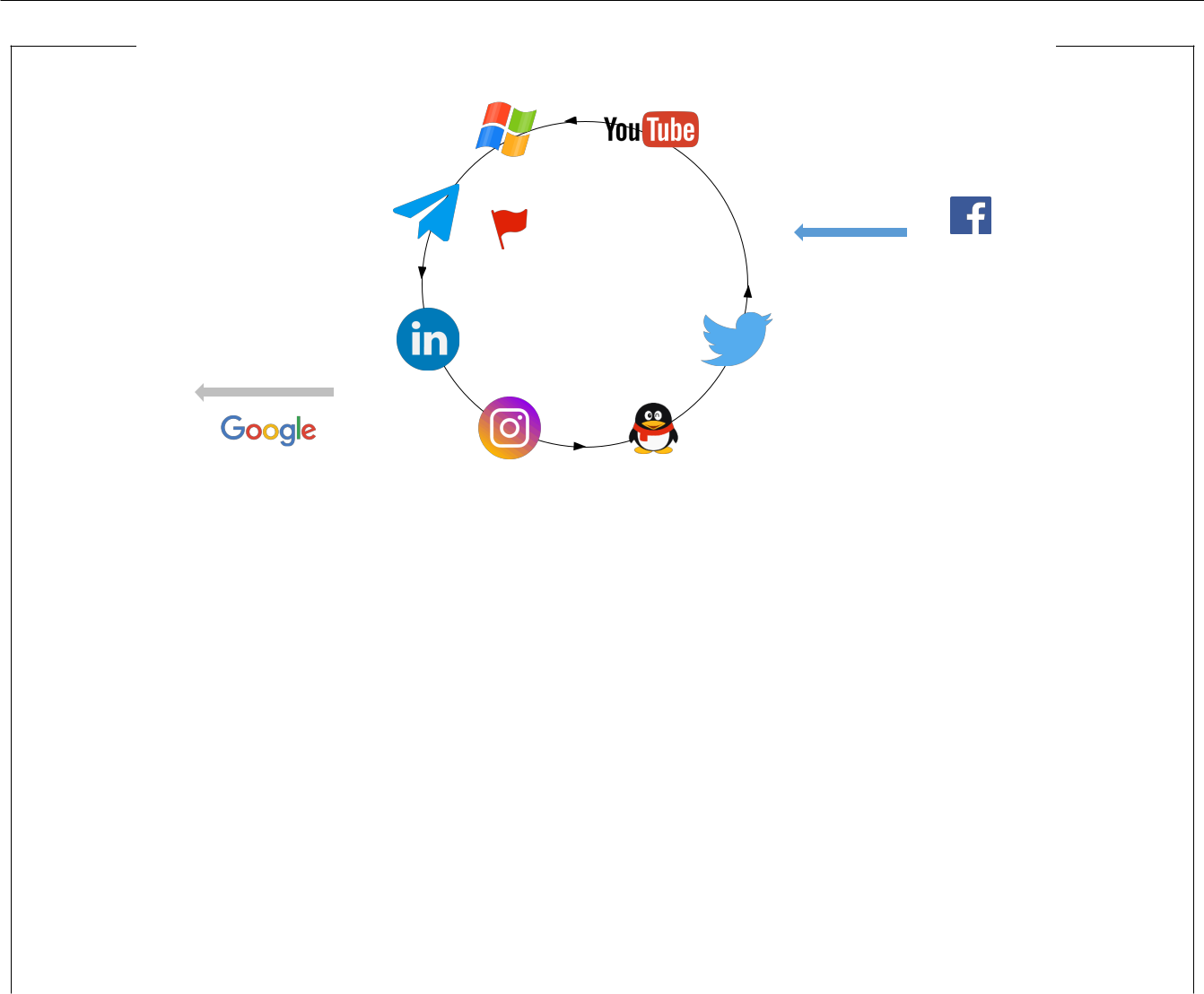
|  |  |
| --- | --- |
| VOLUME 4, 2016 | 5 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks



id = 1

id = 2

id = 3

Old Node Exit

id = 4

id = 0

..... New Node Addition

.

= 2

total\_ = 7

id = 6

= \_ −1

id = 5

Each SNP has a unique serial number Pid, and the serial number increases monotonically from 0.

total\_number represents the current total number of SNPs.

currentID is the serial number of the current PCCS.

Server Election. The function is automatically executed when one of the following three conditions is met. First, the current PCCS’s term ends. Second, other nodes do not receive the heartbeat from PCCS within Tss. Third, other nodes do not receive the CTbsc from any users within Tss.

{currentID = (currentID + 1) mod total\_number;}

New Node Addition. If a new SNP joins the consortium blockchain. The function will be executed automatically. {P id = total\_number ;

total\_number = total\_number + 1}

Old Node Exit. If an SNP exits from the consortium blockchain. P id0 represents the sequence number of the SNP. The function will be executed automatically.

{total\_number = total\_number 1 ;

if P id0 == currentID: function "Server Eelection" is invoked;

if P id0 < currentID: currentID = currentID 1;

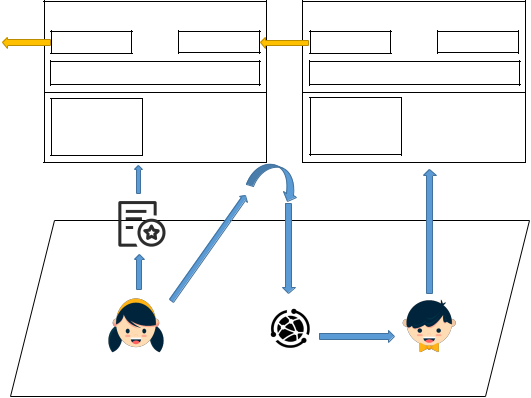
All serial numbers larger than P id0 are subtracted by one;}

**FIGURE 3.** Smart Contract for Proxy Server Election

**Consortium blockchain**

**Off-chian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Block N** | | **Block N+1** | |  |
| Prev Hash |  | Hash | Prev Hash | Hash |  |
| Block Header (Timestamp…) | | | Block Header | (Timestamp…) |  |
| Matching record |  | Smart contract 1 | Matching record | Smart contract 1 |  |
|  | **…** | **…** |  |
| H(T|| || || ′ | | | H(T|| || || |  |
|  |  |  |
| | ′ || ′ ) |  | Smart contract M | ′ || ′ || ′ ) | Smart contract M |  |
| 2. deploy |  |  | 4.Checking and |  |  |
|  | 3. Message | forwarding |  |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  | uploading |  |  |  |
|  |  |  |  | 7.Interaction |  |
| 2.Create |  |  |  |  |  |
| smart contract |  |  | 6. Simple | |  |
|  |  |  |  |
|  |  | 5.ABE partial | decryption | |  |
|  |  |  |  |  |
| 1. Matching |  | decryption |  |  |  |
|  |  |  |  |  |
| preparation |  |  |  |  |  |



**Alice** **PCCS** **Bob**

Then she encrypts her social network address Pa, Pa0 =

EKa (Pa). Additionally, Alice constructs the Bloomer filter.

She first initializes an array B of m bits with 0, and selects

* hash functions fh1; h2; :::; hkg. Alice uses these k hash functions to hash each value of Sa0. The hash value of s0ai is its position in the array B, and the value of this bit in array B is set to 1. Figure 5 shows the procedure of Bloom filter when k = 3.

2) SMART CONTRACT INITIALIZATION

Alice initiates a transaction that creates a smart contract for friend matching. The smart contract is created here but not performed, so we elaborate its details in Section V-C. Note that the smart contract indicates that Alice is looking for friends. Alice needs to upload the following data to the platform layer including the instruction I, the ciphertext of

**FIGURE 4.** Users Communication

the attribute set Sa0, the Bloom filter B, the ciphertext of the

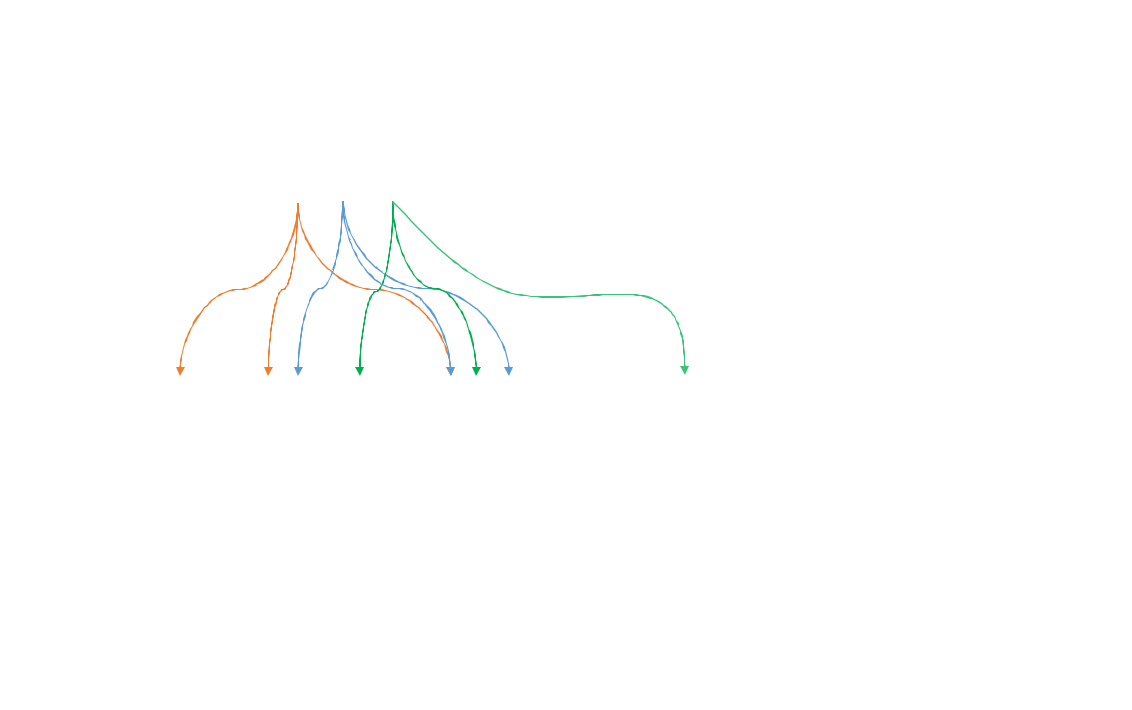
contact address Pa0, and the number of friends max\_num.

Hence, the data transmitted from Alice to the platform

6 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

After the smart contract is successfully deployed, Alice will get a unique address Psc. other users invoke the smart contract via Psc.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

′ = { s′ 1, s′a2, s′a3, … s′an }

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | for s in ′ | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | for | | i | in {1…3} | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | ℎ2( s′a3) | | |  |  |  |  | B [ℎ (s)] = 1; | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | ℎ3( s′a1) | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ℎ1( s′a1) | |  |  |  |  | ℎ2( s′a2) ℎ1( s′a3) ℎ2( s′a2) | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | ℎ2( s′a1) | | |  | ℎ3( s′a2) | | | |  |  |  |  | ℎ3( s′a3) | | |  |
| **B** | 0 | 1 | 0 |  | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |  | 1 | 1 |  | 0 | 0 |  | 0 | 0 | 0 |  | 1 | 0 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | bit 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | bit m-1 | | |  |

**FIGURE 5.** k = 3, the Generation Process of Bloomer Filter

layer to create smart contract is as follows:

Cabc = fIkT kSa0kPa0kBkmax\_numkfh1::hkgg;

(5)

CTabc = EP Ubc (CabckH(Cabc))

After verification, the current PCCS performs partial de-cryption of CP-ABE offline.

5) CP-ABE PARTIAL DECRYPTION

The PCCS uses users’ transformation key T K to decrypt CTab in Capc to get the partially decrypted data CTab0. If a user does not satisfy the requirements of Alice, that is, his/her

* and T K does not meet the access structure(M, ), the decryption fails. Suppose that S meets the access structure and let I f1; 2; 3; :::; ‘g be defined as I = fi : (i) 2 Sg. Then, let f!i 2 Zpgi2I be a set of constants such that if f ig are valid shares of any secret s according to M, then s can

P

be computed as i2I !i i = s. Therefore, the PCCS can computes:

|  |  |
| --- | --- |
| Y | Y |
| e(C0;K)=(e( Ci!i ; L) | e(Di!i ; K (i))) |
| i2I | i2I |

|  |  |  |
| --- | --- | --- |
| Y | (8) |  |
| = e(g; g)s =ze(g; g) st=( e(g; g)t i!i ) |  |

i2I

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | = e(g; g)s =z | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | Then the PCCS obtains partially decrypted data CTab0 | | | | | | | | | | | | | | | | | | | | | | | | | = |  |
| 3) DATA UPLOAD | |  |  |  |  |  |  | (C; e(g; g)s =z). It sends CTpcb to Bob. | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| The data is uploaded to the PCCS including three compo- | | | | | | | | Cpcb = fT kCTab0g; CTpcb = EP Rbc (CpcbkH(Cpcb)) | | | | | | | | | | | | | | | | | | | | | | | |  | (9) |  |
| nents: Psc, Ka and M. Psc is the access address of smart | | | | | | | |  |  |
| contract, Ka is the symmetric secret key of AES, M is | | | | | | | | The PCCS will repeat the above process until Alice makes | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| Alice’s self-introduction or any other information that allows | | | | | | | | max\_num new friends. | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bob to make a preliminary decision whether he is willing to | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| make friends with Alice. So the data needs to be encrypted | | | | | | | | 6) SIMPLE DECRYPTION | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| by CP-ABE as M0 = MkKakPsc. | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | Bob checks T and the hash value to confirm whether the | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| Then Alice converts the Ra into LSSS access structure | | | | | | | | data has been replayed or modified. After that, he computes | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| (M, ). M is an ‘ n matrix, and is a function, in which the | | | | | | | |  |
| simple decryption: | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| rows of M is associated with attributes. Alice first generates | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a random vector ~v = (s; y2; :::; yn) 2 Zpn, which is used to | | | | | | | | C=(e(g; g)s =z)z = | | | | | M | 0 |  | | e(g; g) s=e(g; g)s | | | | | | | | | = | | M | | | 0 | (10) | |  |
| share the encryption s. For i = 1 to ‘, i = ~v Mi, and the ith | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| row of M is the vector M i. Besides, Alice generates another | | | | | | | | 7) SMART CONTRACT INVOCATION | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ‘ a random sequences fr1; :::r‘ | | | 2 Zpg. Then the ciphertext | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| If Bob wants to make new friends after he obtains the | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| CTab is calculated as | | | = gs; | |  |  |  |  |
| C = | M | 0e(g; g) s; C0 |  |  |  | message M sent by Alice, he should make sure whether | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  |  |  |  |  |  |  | Alice satisfies his requirements for making friends. Firstly, | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| (C1 = g 1 H( (1))r | | | | 1 ; D1 = gr1 ; :::; | | | (6) | Bob encrypts his social network contact address Pb as Pb0 | | | | | | | | | | | | | | | | | | | | | | | | | = |  |
| (C‘ = g ‘ H( (‘))r | | | | ‘ ; D‘ = gr‘ ) | |  |  | EKa (Pb). Then he uses Ka to encrypt the attribute sets Sb | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| along with a description of (M, ). Then Alice sends CTapc | | | | | | | | and Rb to get Sb0 and Rb0, respectively. | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| S0 |  |  | s0 | ; :::; s0 |  |  |  |  |  |  | ; s0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| to the platform layer instead of directly to | | | | | PCCS | . |  | = | f | s | | bi 2 | | | S | = E |  |  | (s | |  | ) | g | | ; |  | (11) | |  |
|  |  | b | = | b1 | bn | j 8 |  | b |  | bi |  | Ka | |  |  | bi | |  |  |  |
| Capc = fT kCTabkmax\_numberg; | | | | | | |  | R0 | f | r0 | ; :::; r0 |  | r | | bi 2 | | R |  | ; s0 | = E | |  |  | (s | |  | ) | g | |  |  |  |  |
| (7) | b |  | b1 | bm | j 8 | |  | b | | ri | |  | Ka | |  |  | ri |  |  |  |  |  |
| CTapc = EP Ubc (CapckH(Capc)) | | | | | |  |  | Furthermore, Bob sets the minimum percentage of at- | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| 4) DATA VERIFICATION | | |  |  |  |  |  | tribute similarity between them to min\_percent. Bob sends | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  |  |  |  |  | data to the platform layer as follows: | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| After receiving message from Alice, SNPs in the platform | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | Cbsc = fT kSb0kRb0kPb0kmin\_percent); | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |
| layer decrypts CTapc with P Rbc to get Capc and check data | | | | | | | |  |  |  |  |  |  | (12) | |  |
| integrity and availability with T and hash values. | | | | | | |  |  |  | CTbsc = EP Ubc (CbsckH(Cbsc)) | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Then, the blockchain checks whether the data is legal, i.e. | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| whether there are symmetric secret key or plaintext of access | | | | | | | | Bob initiates a transaction to send the CTbsc to the smart | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| policies in CTapc. | |  |  |  |  |  |  | contract via the address Psc. | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| VOLUME 4, 2016 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |  |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

**C. ATTRIBUTE MATCHING**

After the transaction initiated by Bob is received, the con-sensus nodes first confirm the validity and the data integrity by T and hash value. After that, smart contract is executed by SNP, which mainly verifies whether Alice satisfies Bob’s requirements via a bloom filter. If all elements of Rb0 are in Bloom Filter B, it proves that Alice meets Bob’s needs for making friends. Besides, the smart contract also needs to calculate whether the percentage of attribute similarity between Alice and Bob is greater or equals to the threshold required by Bob.

Algorithm 1 illustrates the attribute matching algorithm, which verifies whether Alice and Bob can be friends and matches up to max\_num friends for Alice. If the match is successful, a communication channel will be established be-tween Alice and Bob. In addition, because the smart contract will be invoked by multiple responders, bloom filter is more efficient than traversal Sa0 and Rb0. The matching records will be stored in the blockchain. Note that we store the hash value of the input parameters instead of the original data, which can achieve matching transparency while protecting user privacy. If original parameters are stored directly, Alice or Bob can decrypt them with the symmetric secret key Ka to obtain other users’ personal information. Moreover, matching records in the blockchain will not be deleted or modified by anyone, which can be managed by users.

**VI. SECURITY ANALYSIS**

In this section, we present the security of the proposed scheme under the threat model in Section III-C.

**A. SINGLE POINT FAILURE**

During the entire matching process, if an SNP loses response or is hijacked by malicious attackers, the system will not be affected and user attribute information will not be disclosed.

Challenge: SNP gets data CTabc, CTapc and CTbsc from users. We Assume that an SNP server is broken down or hi-jacked by attackers. If the entire system crash or the attackers can get the user attribute information from CTabc, CTapc and CTbsc, the attackers will have succeeded.

Proof:

The system will not crash. Owning to the consensus mechanism, the blockchain will not be effected by one single SNP failure. For example, Practical Byzantine Fault Tolerance(PBFT) can hand up to 1/3 malicious consensus nodes [39]. Besides, if the PCCS crashes, as shown in Section V-A3, the smart contract will reelect a new PCCS.

User privacy will not be disclosed. Assuming that the attacker can obtain blockchain node’s public key P Ubc and private key P Rbc, it is also impossible to obtain any user information. Because as shown in Section VI-C, even the SNP cannot obtain user information.

Algorithm 1 Attribute Matching Algorithm

Require:

The ciphertext of users’ social network contact addresses

Pa0, Pb0.

The ciphertext of users’ attribute sets Sa0, Sb0.

The Bloom filter generated by Alice B, and the corre-sponding hash functions fh1; :::; hkg.

The maximum number of friends Alice wants to make, max\_num.

The ciphertext of Bob’s requirements for new friends Rb0. The minimum attribute similarity percentage set by Bob min\_percent.

* 1. matched\_num is used to record the number of friends Alice has matched
  2. if max\_num matched\_num then
  3. return F alse
  4. end if
  5. for r in Rb0 do
  6. for (i = 1; i k; i++) do
  7. if B[hi(r)] == 0 then
  8. return F alse
  9. end if
  10. end for
  11. end for
  12. Connect Sa0 and Sb0 to get S
  13. Quicksort for S to get S0
  14. same\_num = 0
  15. for (i = 0; i < len(S0); i++) do
  16. if S0[i] == S0[i + 1] then
  17. same\_num += 1
  18. end if
  19. end for
  20. Calculate the number of attributes in Sb0 as num\_b
  21. same\_percent = same\_num=num\_b
  22. if min\_percent > same\_percent then
  23. return F alse
  24. else
  25. matched\_num += 1
  26. Send Pa0 and Pb0 to Bob and Alice respectively
  27. Hash values of all input parameters are stored in the blockchain
  28. return T rue
  29. end if

**B. EXTERNAL ATTACK**

External attackers are entities that eavesdrop or tamper with data during the communication. Our scheme can protect user attribute information from various external attacks.

Challenge: The communication data in the system in-cludes CTabc, CTapc, CTpcb and CTbsc. If external attackers can obtain users’ information or change the matching result by replaying or tampering with these data, the attackers will have succeeded.

Proof: The data flow of the proposed scheme is depicted in Figure 4. The all data has been encrypted via the public key

8 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

of the corresponding entity, so external attackers without cor-responding private key cannot decrypt the transmitted data. Except that CTpcb is encrypted by P Rbc, external attackers can use P Ubc to obtain C and CTab0. However, attackers are unable to tamper with CTab0 and cannot get the users’ information.

Attackers is unable to tamper with CT0ab. Because the CTpcb = EP Rbc (CpcbkH(Cpcb)), which contains the hash value of the Cpcb. It means Bob can calculate the hash value of Cpcb and compare it with H(Cpcb). If they are not equal, it indicates the data has been tampered. Bob will destroy the data and report it to the platform layer.

Attackers cannot get the user information. CTab0 = (C; e(g; g)s =z); C = M0e(g; g) s. And z is Bob’s private key, so the attackers cannot decrypt C to get M0 or any other information.

1. **INTERNAL ATTACK**

During the matching process, Alice, Bob, PCSS, SNP cannot know attribute information of the users.

Challenge: If internal attackers are able to obtain the user attribute information from the transmitted data, the attackers will have succeeded.

Proof: Firstly, during the matching process, Alice only receives the matching result, and Bob receives CTpcb = Enc(MkKakPsc) and the matching result. Therefore, even if both parties decrypt the received data to get fM; Ka; Pscg, they can not obtain any attribute information of the other users. Secondly, the PCCS can only know partially decrypted data, because z is Bob’s private key. Additionally, SNPs on the blockchain cannot learn users information, either. Because all attribute sets are encrypted by Ka, which is only known to Alice and Bob. Moreover, SNP can not learn anything from bloom filter B, because the attribute space A is large enough, and the SNP does not know the size of Ra.

**D. COLLUSION ATTACK**

The proposed protocol can resist collusion attacks from mul-tiple entities, and we mainly analyze the collusion between SNPs and users.

Challenge 1: SNPs on the blockchain get S0 and R0 from Alice and bob, which is encrypted by Ka. We assume that a SNP and Alice collude to decrypt Sb0 and Rb0 with Ka (the same as Bob).That is, Alice adds Ka to the uploaded data. If they could obtain Bob’s attribute set Sb, Rb, the collusion attack is successful.

Proof: Transactions in the blockchain are packaged into blocks by random consensus nodes. It means other SNPs may receive the symmetric key Ka from Alice, in which case Alice will be punished and the system will no longer provide friend matching for Alice. In addition, even in some con-sensus algorithms, the designated consensus node receives transactions. The transaction will be confirmed by all the consensus nodes and it will be detected if the transaction contains Ka.

Challenge 2: The PCCS gets CTapc from Alice. We Assume that the current PCCS colludes with Alice. And Alice sends the plaintext of access control structure or the symmetric secret key to the PCCS. If they obtain Bob’s attribute information, the collusion attack is successful.

Proof: As described in Section V-B4, before the PCCS gets the CTapc, all SNPs have recorded the CTapc and confirm whether the CTapc is legal. If it contains Ka or the plaintext of the access control structure, Alice’s operation is rejected and the match is terminated.

**VII. PERFORMANCE EVALUATION**

In this section, we first introduce the simulation experiment environment, then we analyze the performance of the scheme based on Hyperledger Fabric. Finally, the function of our scheme is compared with the existing schemes.

**A. SIMULATION EXPERIMENT PREPARATION**

The simulation experiment is implemented in C/C++ on the Ubuntu 16.06 virtual machine under Windows 10 with Inter Inter(R) Core(TM) i5-7500 CPU @ 3.40GHz. Our experiment focused on the implementation of AES, outsour-ing decryption of CP-ABE and consortium blockchain. We ignore the underlying communication consumption between entities.

Firstly, the traditional AES with 128-bit key was imple-mented in C and the bloom filter was constructed in C++ based on Open Bloom Filter library. Afterwards, the outsour-ing decryption of CP-ABE was conducted based on Stanford Pairing-Based Crypto (PBC) library. It should be noted that the main computational consumption in CP-ABE is caused by bilinear operation, so we ignore the arithmetic operation on Zp. In addition, we deployed a consortium blockchain on Hyperledger Fabric which is constructed based on Go and Docker. We set all SN P s as endorsing peers. And the smart contract was implemented in Go.

We choose Fabric instead of Ethereum, the reasons are listed .

Ethereum is a public blockchain, which is almost im-possible for SNPs to trust it and use it.

The transaction in Ethereum requires an additional fee called "Gas", which users are unwilling to bear.

Ethereum’s consensus mechanism leads to low transac-tion execution efficiency, which does not apply to friend matching in social networks.

We focus on the relationship between the performance of the scheme and the number of attributes, so we assume that Alice and Bob have ‘ private attributes and ‘ preference attributes, respectively. And ‘ = f5; 10; :::; 95; 100g. The experimental results are depicted in Figure 6, all the data were the average of 20 experiments.

**B. PERFORMANCE ANALYSIS**

Figure 6(a) shows the time consumed by AES encryption for different numbers of attributes. Each attribute in R and

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 9 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access

Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 4 | 4 |  |  | AES\_Enc |  |
|  |  | 4 |  |  | AES\_Enc |  |
| 3.5 | |  |  |  | AES\_Enc |  |
|  |  |  | Bloom\_Gen |  |
|  | 3.5 | |  |  |  |
|  |  | 3.5 |  |  | Bloom\_Gen |  |
| **(ms)** | 3 |  |  |  | Bloom\_Gen |  |
| 33 |  |  |  |  |
|  |  |  |  |  |  |
| **(ms)(ms)**2.5 | | |  |  |  |  |
| 2.5 | | 2.5 |  |  |  |  |
|  | 2 22 | |  |  |  |  |
|  | 1.5 | |  |  |  |  |
| 1.5 | | 1.5 |  |  |  |  |
| **Rruntime** |  | 11 |  |  |  |  |
| **RruntimeRruntime** | |  |  |  |  |  |
|  | 1 |  |  |  |  |  |
|  | 0.5 | |  |  |  |  |
| 0.5 | | 0.5 |  |  |  |  |
| 0 |  |  |  |  |
|  |  | 0 | 5 |  | 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95100 |  |
|  | 0 |  | 5 |  |
|  |  |  | 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95100 |  |
|  |  |  | 5 | 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95100 | |  |
|  |  |  |  |  | **of** |  |
|  |  |  |  |  | **Number of Attributes** |  |

**Number of Attributes**

(a) The Performance of AES and Bloom Filter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 400400 | |  | ABEABEEnc\_Enc |  |
|  | 400 | |  |  |
|  |  | ABE\_Enc |  |
|  | 350 | |  | ABEABEDec**\_**PCCSDecPCCS |  |
|  |  | 350 |  | ABE\_Dec\_PCCS |  |
|  | 350 | |  | ABE DEC\_Local |  |
|  |  |  |  | ABE\_DEC\_Local |  |
| **(ms)** | 300 | |  | ABE\_DEC\_Local |  |
| 300300 | |  |  |  |
| 250 | |  |  |  |
| **(ms)**250 | | |  |  |  |
| **Rruntime** |  | 250 |  |  |  |
| 200 | |  |  |  |
| 200 | |  |  |  |
|  | 200 |  |  |  |
| 150 | |  |  |  |
| 150 | |  |  |  |
|  | 150 |  |  |  |
| 100 | |  |  |  |
| 100 | |  |  |  |
| **Rruntime** | | 100 |  |  |  |
|  |  | 50 |  |  |  |
|  |  | 50 |  |  |  |
|  |  | 050 |  |  |  |
|  |  | 0 | 5 | 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95100 |  |
|  |  | 0 5 | | 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95100 |  |
|  |  |  |  | 5 1 0 1 5 2 0 2 5 3 0 3 5 4 **of**04 5 5 0 5 5 6 0 6 5 7 0 7 5 8 0 8 5 9 0 9 5 1 0 0 |  |
|  |  |  |  | **Number of Attributes** |  |

**Number of Attributes**

(b) The Performance of Outsouring Decryption of CP-ABE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1200 |  |  |  |  |  |  |  |  |  |  |  |  | 1200 |  |  |  |  |
|  | 1200 |  |  | SC\_initiation | | | |  |  |  |  |  | **(ms))(** | 1200 |  | Alice | Bob |  |
|  |  |  |  |  |  |  |  |  |  |  | Alice | Bob |  |
|  |  |  |  | SC\_initiation | | | |  |  |  |  |  |  |  |  |  |  |
|  | 1000 |  |  | Attribute matching | | | | |  |  |  |  | 1000 |  | Blockchain | PCCS |  |
|  | 1000 |  |  |  |  |  |  |  | Blockchain | PCCS |  |
| **(ms)** |  |  |  | Attribute matching | | | | |  |  |  |  |  |  |  |  |  |
| 8000 |  |  |  |  |  |  |  |  |  |  |  | 800 |  |  |  |  |
| **(ms)** |  |  |  |  |  |  |  |  |  |  |  |  | **Timeconsuptionm** |  |  |  |  |  |
| **Rruntime** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Rruntime** | 6000 |  |  |  |  |  |  |  |  |  |  |  |  | 600 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 4000 |  |  |  |  |  |  |  |  |  |  |  |  | **400** |  |  |  |  |
|  | 20**0**0 |  |  |  |  |  |  |  |  |  |  |  |  | **200** |  |  |  |  |
|  | 00 |  |  |  |  |  |  |  |  |  |  |  |  | **0** |  |  |  |  |
|  | 11 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99 | 1100 1111 1122 1133 1144 1155 1166 1177 | 1188 | 119 20 |  | **5** | **1 0** | 15 20 25 3300 3355 4400 4455 5500 5555 6600 6655 7700 7755 8800 8855 9900 9955110000 | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Numberber of Attributest s** | |  | **Number of Attributes** |  |
|  |  | (c) The Performance of Blockchain | | (d) Total Time Consumption of Each Entity | | |
|  |  |  | **FIGURE 6.** Performance Evaluation of the Scheme | |  |  |
| **TABLE 2.** Comparisons of Functionality | | |  |  |  |  |
|  |  |  |  |  |  |  |
|  | Scheme | Fine-Grained Matching | Low User Consumption | Bidirectional Matching | Collusion Resistance | Blockchain-Based |
|  | Ref. [31] | Y | N | N | N | N |
|  | Ref. [13] | Y | Y | Y | N | N |
|  | Ref. [10] | N | Y | N | ? | N |
|  | Ref. [25] | N | Y | N | Y | N |
|  | Ref. [11] | Y | Y | N | Y | N |
|  | Ref. [32] | Y | Y | N | ? | N |
|  | Our scheme | Y | Y | Y | Y | Y |

* is encrypted separately. It can be seen from the figure that the time consumption has not changed significantly with the increase of the number of attributes. Besides, we set the tolerable false positive probability of bloom filter as 0:0001. We can find that the bloom filter has extremely high efficiency, the time consumption is increasing with the number of attributes, but is not more than 0:2ms.

Figure 6(b) presents the performance of outsouring de-cryption of CP-ABE. We use EG; ET ; P to denote the aver-age time to complete an exponentiation in G1, exponentiation in GT and a paring respectively. The average time they consume is 1:3953ms, 0:9876ms, and 0:1839ms. From the Section V, we know the encryption time consumption of CP-ABE is fP + ET + (2‘ + 1)EGg. And the partial decryption time consumption is f(2 + ‘)P + 2‘EGg. Most importantly, the local decryption time is ET , which does not change with the number of attributes increasing.

Figure 6(c) depicts the relationship between the running time of the designed blockchain and the number of attributes, including the time that it takes to initiate a transaction to generate a smart contract, and the time that it takes to execute attribute matching. We record the full running time of the transaction from initiation to store in the blockchain. Compared with the time consumed by the blockchain, the time consumption caused by the changes in the number of attributes can be negligible.

The total time consumption of each entity is shown in Figure 6(d). Alice performs encryption of CP-ABE, encryp-tion of AES and Bloom filter generation. For Bob, he needs to complete local decryption of CP-ABE and encryption of AES. For the blockchain, it performs smart contract ini-tialization and attribute matching. For PCCS, it performs outsourcing decryption of CP-ABE. From the figure, the blockchain operation and outsourcing decryption of CP-

10 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

ABE take the most time. It demonstrates the necessity of transferring complex calculations from the blockchain to the off-chain, which can significantly reduce the computation cost on the blockchain. What’s more, an initiator will have multiple responders, so the reduction of time spent by the responder can effectively decrease the consumption of the entire matching process. The experimental result in Figure 6(d) demonstrates that Bob’s time consumption is negligible, which proves the effectiveness of our scheme.

In summary, experimental results show that the scheme is effective and feasible.

1. **FUNCTIONAL COMPARISON OF SIMILAR SCHEMES**

We compare the proposed scheme to some similar advanced schemes introduced in Section II, and the comparison re-sults are demonstrated in Table 2. All the schemes can achieve privacy preservation, so it is not displayed in the table. To achieve privacy preservation, there are five key factors are considered: Fine-Grained Matching, Low User Consumption, Bidirectional Matching, Collusion Resistance and Blockchain-Based. The Y in the table indicates that this feature is available in this scheme and the N represents that it is not. And ? means the scheme is a point-to-point attribute matching scheme, so collusion attack is not considered.

It can be concluded that we proposed a novel and com-plete friend matching protocol based on the new blockchain architecture with the above five characteristics.

**VIII. CONCLUSION**

In this paper, we present a privacy-preserving attribute matching scheme under multiple semi-trusted servers. In our scheme, we utilize CP-ABE and bloom filter to conduct bidi-rectional attribute matching and relieve the computation cost of users by outsourcing decryption. In addition, we design a novel hierarchical blockchain architecture, which massively reduces the storage consumption of the blockchain and im-proves operating efficiency. Security analysis and experiment results demonstrate that our scheme can resist single point failure attack, collusion attack, internal attack and external attack, and also provide effectively friend matching for users.

In the future, we consider using blockchain instead of the trusted third party to initialize CP-ABE, which is a challenge issue. In addition, we plan to analyze the security and efficiency of blockchain consensus and friend matching in practical application.

**REFERENCES**

1. S. Oukemeni, H. Rifà-Pous, and J. M. M. Puig, “Privacy analysis on mi-croblogging online social networks: A survey,” ACM Computing Surveys (CSUR), vol. 52, no. 3, pp. 1–36, 2019.
2. Y. Lin, Z. Cai, X. Wang, and F. Hao, “Incentive mechanisms for crowd-blocking rumors in mobile social networks,” IEEE Transactions on Vehic-ular Technology, vol. 68, no. 9, pp. 9220–9232, 2019.
3. Y. Pu, T. Xiang, C. Hu, A. Alrawais, and H. Yan, “An efficient blockchain-based privacy preserving scheme for vehicular social networks,” Informa-tion Sciences Journal, vol. 540, pp. 308–324, 2020.
4. Z. Cai, Z. He, X. Guan, and Y. Li, “Collective data-sanitization for preventing sensitive information inference attacks in social networks,”

IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 4,

pp. 577–590, 2016.

1. X. Yi, E. Bertino, F.-Y. Rao, and A. Bouguettaya, “Practical privacy-preserving user profile matching in social networks,” in 2016 IEEE 32nd international conference on data engineering (ICDE). IEEE, 2016, pp. 373–384.
2. P. Chaudhary and B. Gupta, “A novel framework to alleviate dissemination of xss worms in online social network (osn) using view segregation,” Neural Network World, vol. 27, no. 1, p. 5, 2017.
3. M. Li, N. Cao, S. Yu, and W. Lou, “Findu: Privacy-preserving personal profile matching in mobile social networks,” in 2011 Proceedings IEEE INFOCOM. IEEE, 2011, pp. 2435–2443.
4. Y. Ishikuro and K. Omote, “Privacy-preserving profile matching protocol considering conditions,” in International Conference on Network and System Security. Springer, 2016, pp. 171–183.
5. D. He, Z. Cao, X. Dong, and J. Shen, “User self-controllable profile matching for privacy-preserving mobile social networks,” in 2014 IEEE International Conference on Communication Systems. IEEE, 2014, pp. 248–252.
6. E. Luo, Q. Liu, J. H. Abawajy, and G. Wang, “Privacy-preserving multi-hop profile-matching protocol for proximity mobile social networks,” Future Generation Computer Systems, vol. 68, pp. 222–233, 2017.
7. C.-z. Gao, Q. Cheng, X. Li, and S.-b. Xia, “Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network,” Cluster Computing, vol. 22, no. 1, pp. 1655–1663, 2019.
8. G. Sheng, T. Wen, Q. Guo, and Y. Yin, “Privacy preserving inner product of vectors in cloud computing,” International Journal of Distributed Sensor Networks, vol. 10, no. 5, p. 537252, 2014.
9. T. Li, X. Li, and X. Liu, “An efficient privacy-preserving bidirectional friends matching scheme in mobile social networks,” in 2019 International Conference on Networking and Network Applications (NaNA). IEEE, 2019, pp. 51–57.
10. W. Wang, F. Qi, X. Wu, and Z. Tang, “Distributed multi-authority attribute-based encryption scheme for friend discovery in mobile social networks,” Procedia Computer Science, vol. 80, pp. 617–626, 2016.
11. F. Qi, X. Chang, Z. Tang, and W. Wang, “Searchable attribute-based encryption protocol with hidden keywords in cloud,” in International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications. Springer, 2019, pp. 80–92.
12. W. Cui, C. Du, and J. Chen, “Cp-abe based privacy-preserving user profile matching in mobile social networks,” PloS one, vol. 11, no. 6, 2016.
13. T. Guo, K. Dong, L. Wang, M. Yang, and J. Luo, “Privacy preserving pro-file matching for social networks,” in 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD). IEEE, 2018, pp. 263–268.
14. B. Chandrasekaran, Y. Nogami, and R. Balakrishnan, “An efficient hi-erarchical multi-authority attribute based encryption scheme for profile matching using a fast ate pairing in cloud environment,” Journal of Communications Software and Systems, vol. 14, no. 2, pp. 151–156, 2018.
15. S. He, Q. Wu, X. Luo, Z. Liang, D. Li, H. Feng, H. Zheng, and Y. Li, “A social-network-based cryptocurrency wallet-management scheme,” IEEE Access, vol. 6, pp. 7654–7663, 2018.
16. S. Zhu, W. Li, H. Li, L. Tian, G. Luo, and Z. Cai, “Coin hopping attack in blockchain-based iot,” IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4614–4626, 2018.
17. Y. Pu, C. Hu, S. Deng, and A. Alrawais, “Rpeds: A recoverable and revocable privacy-preserving edge data sharing scheme,” IEEE Internet of Things Journal, 2020.
18. Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, “Cost-efficient strategies for restraining rumor spreading in mobile social networks,” IEEE Trans-actions on Vehicular Technology, vol. 66, no. 3, pp. 2789–2800, 2016.
19. S. Hasavari and Y. T. Song, “A secure and scalable data source for emergency medical care using blockchain technology,” in 2019 IEEE 17th International Conference on Software Engineering Research, Manage-ment and Applications (SERA). IEEE, 2019, pp. 71–75.
20. S. Zhu, Z. Cai, H. Hu, Y. Li, and W. Li, “zkcrowd: a hybrid blockchain-based crowdsourcing platform,” IEEE Transactions on Industrial Infor-matics, vol. 16, no. 6, pp. 4196–4205, 2019.
21. R. Li, H. Li, X. Cheng, X. Zhou, K. Li, S. Wang, and R. Bie, “Perturbation-based private profile matching in social networks,” IEEE Access, vol. 5, pp. 19 720–19 732, 2017.
22. C. Li, Z. Zhang, and L. Zhang, “A novel authorization scheme for multi-media social networks under cloud storage method by using ma-cp-abe,” International Journal of Cloud Applications and Computing (IJCAC), vol. 8, no. 3, pp. 32–47, 2018.

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 11 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

PLACE

PHOTO

HERE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3016986, IEEE Access



Yang et al.: An Efficient Blockchain-Based Bidirectional Friends Matching Scheme in Social Networks

1. Q. Huang, L. Wang, and Y. Yang, “Secure and privacy-preserving data sharing and collaboration in mobile healthcare social networks of smart cities,” Security and Communication Networks, vol. 2017, 2017.
2. X. Sun, Y. Yao, Y. Xia, X. Liu, J. Chen, and Z. Wang, “Towards efficient sharing of encrypted data in cloud-based mobile social network.” KSII Transactions on Internet & Information Systems, vol. 10, no. 4, 2016.
3. J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in 2007 IEEE symposium on security and privacy (SP’07). IEEE, 2007, pp. 321–334.
4. B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in International Workshop on Public Key Cryptography. Springer, 2011, pp. 53–70.
5. E. Luo, Q. Liu, and G. Wang, “Hierarchical multi-authority and attribute-based encryption friend discovery scheme in mobile social networks,” IEEE Communications Letters, vol. 20, no. 9, pp. 1772–1775, 2016.
6. H. Li, X. Cheng, K. Li, and Z. Tian, “Efficient customized privacy preserving friend discovery in mobile social networks,” in 2015 IEEE 35th International Conference on Distributed Computing Systems. IEEE, 2015, pp. 225–234.
7. L. Jiang and X. Zhang, “Bcosn: A blockchain-based decentralized online social network,” IEEE Transactions on Computational Social Systems, vol. 6, no. 6, pp. 1454–1466, 2019.
8. K. Gu, L. Wang, and W. Jia, “Autonomous resource request transaction framework based on blockchain in social network,” IEEE Access, vol. 7, pp. 43 666–43 678, 2019.
9. C. Liu, Y. Xiao, V. Javangula, Q. Hu, S. Wang, and X. Cheng, “Norma-chain: A blockchain-based normalized autonomous transaction settlement system for iot-based e-commerce,” IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4680–4693, 2018.
10. M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman et al., “Blockchain technology: Beyond bitcoin,” Applied Innovation, vol. 2, no. 6-10, p. 71, 2016.
11. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” Manubot, Tech. Rep., 2019.
12. M. Green, S. Hohenberger, B. Waters et al., “Outsourcing the decryption of abe ciphertexts.” in USENIX security symposium, vol. 2011, no. 3, 2011.
13. Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in 2017 IEEE international congress on big data (BigData congress). IEEE, 2017, pp. 557–564.

FEIHONG YANG graduated from Chongqing U-niversity in China with the B.S. degree in 2019. He is currently pursuing the master degree in the School of Big Data & Software Engineering, Chongqing University. His research interests in-clude Iot applications based on blockchain, priva-cy and security issues in the blockchain.



YING WANG received her bachelor’s degree in Internet of Things from Chongqing University of Technology. She is now studying for mastera˛´rs de-gree at Chongqing University. Her major research interest is information security.



CHUNLEI FU is currently a senior engineer at School of Big Data & Software Engineering, Chongqing University. He received his PhD de-gree at in School of Automation, Chongqing U-niversity, China, in 2014. He received a postdoc-toral training at the school of computer science in Chongqing University, studying formal meth-ods for software engineering. His research areas include cybersecurity, blockchain, and software engineering.



CHUNQIANG HU received the B.S. degree in computer science and technology from Southwest University, Chongqing, China, in 2006, the M.S. and Ph.D. degrees in computer science and tech-nology from Chongqing University, Chongqing, China, in 2009 and 2013, respectively, and the second Ph.D. degree in computer science from The George Washington University, Washington, DC, USA, in 2016. He was a Visiting Scholar with The George Washington University from 2011 to 2012.



He is currently a Faculty Member with the School of Big Data & Software Engineering, Chongqing University. His research interests include privacy-aware computing, big data security and privacy, wireless and mobile security, applied cryptography, and algorithm design and analysis. He was honored with the Hundred-Talent Program by Chongqing University. He is a member of the ACM.

ARWA ALRAWAIS Arwa Alrawais obtained her Ph.D from the Department of Computer Science, The George Washington University, Washington DC in 2017. She also received her M.S. degree in Computer Science from The George Washington University in 2011. Recently, she obtained a patent in a˛ˇrSystem and method for remote authentication with dynamic usernamesa˛s´. Her current research interests include network security, wireless and mobile security, and algorithm design and analy-

sis. She is a professional reviewer for several conferences and journals of IEEE and ACM.



12 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.