Friendship Protection: A Trust-Based Shamir Secret Sharing Anti-collusion Attack Strategy for Friend Search Engines

Junfeng Tian and Yue Li(✉)

School of Cyberspace Security and Computer, Hebei University, Baoding 071000, China

Abstract. Online social networks (OSNs) provide users with applications to interact with friends or strangers. Among these applications, the friend search engine allows users to query other users' personal friend lists. However, if there is no suitable protection strategy, the application is likely to compromise the user's privacy. Some researchers have proposed privacy protection schemes to protect users from attacks that are initiated by independent attackers, but few researchers have conducted research on collusion attacks initiated by multiple malicious requestors. In this paper, we propose a resistance strategy against collusion attacks that are initiated by multiple malicious requestors in OSNs, introduce trust metrics, and limit users' ability to query through the Shamir secret sharing system (t, n) threshold function in the friend search engine to protect the user's friendships from collusion attacks by multiple attackers. The effectiveness of the proposed anti-collusion attack strategy is verified via synthetic and realistic social network datasets. Research on collusion attack strategies will help us design a safer friend search engine for OSNs.

Keywords: Friend search · Collusion attack · Threshold function

1 Introduction

Online social networks (OSNs) have increasingly become an indispensable social activity in people's lives. OSNs provide users with various applications to interact with family, friends and even strangers. Social networks provide users with a variety of services, such as interactive dating and online shopping. Users can interact, chat, and trade via social networks. As the number of social network users continues to increase and social networks continue to improve sociality, there are increasingly more attacks on users' privacy. Similar to mobile social networks, which are a social network category, the widespread nature of OSNs and their lack of effective privacy-preserving architecture make them a target for several attacks by adversaries [1]. Therefore, protecting the security of users' private information is important.

Social networks tend to display as many friend lists of users as possible, and a new social network application, the friend search engine, serves existing users in the social network and can be employed to attract potential users to join the network. The search

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 367–385, 2021.
https://doi.org/10.1007/978-3-030-89814-4_27

engine allows ordinary users to query a list of friends of individual users so that users can find friends of friends. People may join a social network because their friends have joined the social network, thereby attracting more users, which improves the sociality of social networks and expands the use of social networks. However, with the increasing number of social networks and the increasing number of users in social networks, some researchers have observed that friend search engines may expose more friendships than users are willing to show. The friends that users are not willing to show comprise private data of users, referred to as friendship privacy. The privacy-aware friend display scheme [2] can not only successfully protect the friendship privacy of users but also improve the sociality of OSNs. However, researchers studying collusion attacks in OSNs considered a collusion attack that was launched by multiple malicious requestors in coordination with each other and a collusion attack that can destroy the privacy of users' friendships and cause users to expose more friendships than they are willing to share [3]. And the data sharing framework can resolve potential data leakage [4]. We focus on the design of anti-collusion attack strategies that are aimed at the privacy of users' friendships in OSNs. The major contributions of this paper are presented as follows:

First, we analyse the collusion attack method in the friend search engine [3], in which multiple malicious requesters coordinate with each other to initiate queries to different but related users by designing a query sequence, thereby destroying the privacy of the friendships of the target user.

Second, we propose methods that can resist collusion attacks on friend search engines in OSNs. For resistance, we design a strategy for these collusion attacks, introduce trust metrics [5] to restrict access to requestors in friend search engines, and use the Shamir Secret Sharing (SSS) system (t, n) threshold function [6] to limit queries. In this paper, a trust-based (t, n) threshold anti-collusion attack strategy is proposed to prevent collusion attacks launched by multiple malicious requestors who coordinate the query sequence and query targets.

Third, to evaluate the effectiveness of the proposed anti-collusion attack strategy, we implement the strategy in a synthetic dataset and three large-scale real-world datasets. The experimental results show that the proposed anti-collusion attack strategy works effectively on large-scale datasets. By comparing the probability of a successful attack by a malicious inquirer using the proposed anti-collusion attack strategy with the probability of a successful attack by a malicious inquirer without the anti-collusion attack strategy, we determine that under the same attack conditions, the anti-collusion attack strategy in this work can reduce the probability of a successful attack, thereby protecting users' friendship privacy.

2 Related Works

2.1 Attacks Against Friendship Privacy

The number of users in OSNs continues to grow. Tens of thousands of users search for new friends and establish new contacts every day. Therefore, the privacy problem in friend search engines has attracted the attention of many researchers. Attacks against the privacy of friendships in OSNs can be divided into two categories, that is, attacks initiated by independent attackers and by colluding attackers.

Regarding independent attacks, research on modeling malicious attacks in OSNs shows that malicious individuals use the actual trust relationship between users and their family and friends to spread malware via OSNs [7]. By changing the display of malicious posts and personal information and hiding him/herself to avoid detection, an attacker in a chameleon attack, which is a new type of deception based on OSNs, is able to destroy users' privacy [8]. Studies have also shown that when the topology of OSNs does not contain cycles, malicious entities will violate users' privacy via active attacks if the network structure is not carefully designed [9]. Due to the rapid development of convolutional neural networks in recent years, applying them to social networks can result in very effective reasoning attacks and make high-precision predictions about private data [10]. In addition, by using the friend search application programming interface (API) to randomly grab data in OSNs, independent attackers can collect users' friendships. For some location-based social networks (LBSNs), despite the privacy protection strategies of location confusion and relative location, attackers can still perform a series of attacks by running LBSN apps and easily inferring the user's location [11].

Collusion attacks involve multiple malicious entities with the aim of launching a malicious attack through the coordination of multiple malicious entities to obtain more private information than is obtained in independent attacks. When users publish personal information in OSNs, attackers can launch inference attacks based on non-sensitive attributes and social relationships. An attacker utilizes users' profiles and social relationships in a collective manner to predict the sensitive information of related victims in a social network dataset that has been released [12]. Multiple malicious entities can be fake accounts that are created by a single attacker or different real attackers [13–15]. The router and users can maliciously collude to perform a collusion name guessing attack to compromise people's privacy [16]. Compared with independent attacks, collusion attacks are more complex and often exploit system vulnerabilities that independent attacks cannot detect. There is a complex collusion attack strategy, in which multiple malicious users coordinate their queries, share the query results, and dynamically adjust their queries based on the system's feedback to other malicious requestors [4].

2.2 Preservation of Friendship Privacy

The personalized privacy measurement algorithm can calculate the user's privacy level, thereby protecting user privacy data [17]. To protect the privacy of users' friendships in OSNs, some researchers have proposed a trust-based privacy protection friend recommendation scheme [18]. By comprehensively considering user interests and topological characteristics, a constraint traversal method is used to identify a strong trust path from trustees to trustees [19]. Research can alleviate cascading failures in trust relationships and be beneficial for the honest and free expression of opinions and experiences without users' privacy being compromised or trust relationships being affected. This research is built upon the dynamic modeling of cascading failures due to the occurrence of a trust crisis within a unidirectional or bidirectional social network [20]. The threat list and privacy protection mechanism illustrate the security requirements that OSNs should satisfy [21]. In response to privacy leakage caused by location information, researchers have proposed an environment-based system-level privacy protection solution that aims to automatically learn users' privacy preferences in different environments and provide

users with transparent privacy control [22]. As noted by [23], some researchers have conducted comprehensive surveys on credibility in OSNs, such as the credibility of information for users and the evaluation of the trust level. In addition, a series of studies have proposed an unsupervised trust inference algorithm that is based on collaborative filtering in weighted social networks and a fast and robust trust inference algorithm [24, 25] to strengthen the security of social networks via trust inference.

However, researchers rarely consider privacy leakage problems caused by the friend search service provided by OSNs. Research on these problems can address the privacy needs of users' friends while ensuring the sociality of OSNs. The solution adopted by most OSNs is to allow each individual user to choose to completely display or completely hide their entire friend list. Moreover, OSNs often default their users to expose their entire friend list, of which most users are unaware [26]. It is conceivable that this setting aims to increase the sociality of the OSN. If users set their friend lists to completely hidden to protect the privacy of their friendships, this setting will substantially affect the sociality of OSNs. There are also some OSNs that set the users' friend list display to "show only a fixed number." For example, on Facebook, the number of friends displayed is set to 8, which limits the flexibility of users in changing their personal settings. However, some researchers have discovered that randomly displaying eight friends is sufficient for third parties to obtain data to estimate friend lists [27]. Moreover, regarding the different privacy settings of users, consider the following example: if A and B are friends, even if user A hides his or her friend list and the requestor cannot query the friend list of A, if user B is set to display his or her friend list, when the requestor queries the friend list of B, the friendships of B and A will be displayed and destroy A's privacy. This problem is referred to as the "mutual effect" [2].

To better protect the privacy of users' friendships in OSNs, a privacy protection strategy in the friend search engine [2] was shown to successfully resist attacks initiated by independent attackers. However, the strategy was unable to defend against collusion attacks initiated by multiple malicious attackers. Subsequently, an advanced collusion attack strategy coordinated by multiple malicious requestors [3] showed that multiple malicious requestors with limited knowledge of OSNs can successfully destroy users' privacy settings in the friend search engine. Another study [28] implemented web applications to detect malicious behavior, such as collusion attacks in the friend search engine. However, few researchers have investigated how to resist collusion attacks initiated by malicious attackers in friend search engines.

In this paper, we propose an anti-collusion attack strategy to fill these research gaps. This strategy distinguishes trusted users from untrusted users based on the credibility among users in OSNs and uses the (t, n) threshold function to limit the querying of requestors in the friend search engine to resist malicious attacks initiated by colluding attackers in OSNs.

3 Collusion Attack Strategy

3.1 Definition

In friend search engines, to strengthen the protection of the user's friendships, a certain number of friendships, such as k, will be displayed when responding to a query request.

These k friends are defined as the most influential friends of the users in the OSN. Assume that node N_a exists in the OSN with direct friends $N_{a.i}$ and that the set is $F_a^k(i < k)$. Requestor Q_1 wants to query N_a's friendships; two nodes, N_1 and N_2, exist, and $k = 1$. N_1 and N_2 are each user's most important friends.

Occupation. If requestor Q_1 queries node N_1, based on the friend search engine display strategy, the query result is $E(N_1, N_2)$. At this time, N_1 has shown his or her most important friend N_2, and N_1 is occupied.

Passive Display. Requestor Q_1 queries the important friend list of N_1. Based on the friend search engine strategy, the query result is $E(N_1, N_2)$. The most important friend who exposes N_2 is N_1, and N_2 is referred to as a passive display.

3.2 Attack Model

Maximum Number of Friends Displayed. Due to the different personal preferences of users in OSNs, their privacy settings will also be different. The maximum number of friends displayed, k, may also be different. This strategy assumes that all nodes have the same k value.

Attackers' Prior Knowledge. Typically, the success of a malicious requestor's attack is closely related to his or her knowledge of OSNs. The attack success rate of malicious requestors who know more about OSNs is expected to be higher. This paper assumes that malicious requestors have limited knowledge of OSNs and are limited only to target nodes.

Attack Target. The goal of the malicious query is to violate the privacy of the target user in the OSN (i.e., to query the $k + 1$th friend of the target node). When the privacy of the target user is set to show a number of friends less than k, the privacy of the target user cannot be violated. Each malicious requestor's attack target is unique, and each collusion attack has only one victim node. Although malicious requestors may infringe on the privacy of other users during the query process, only when the privacy of the target node is destroyed is the collusion attack considered successful.

Attack Strategy. Colluding attackers in OSNs can query users' friendships via the friend search engine and query the relationship between users and friends by coordinating the query sequence and query targets. We take a simple scenario—the small-scale complete graph that contains the four nodes shown in Fig. 1—as an example to analyze the perpetrated collusion attacks.

The OSN has four user nodes, N_1, N_2, N_3 and N_4, and two malicious requestors, MR_1 and MR_2. Additionally, $k = 1$. Set node N_3 as the target node, and perform the following query.

First, MR_1 queries N_3. The first important friend N_1 of node N_3 can be queried. Then, MR_2 successively queries N_1 and N_3 by returning N_1's first important friend as N_2 and occupying N_1. Finally, query N_3. Simultaneously, N_3 will expose the $k + 1$ friend, N_4, and the privacy of N_3 will be violated.

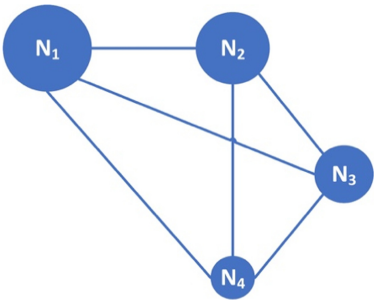

Fig. 1. Small-scale complete graph

4 Anti-collusion Attack Strategy

The adopted collusion attack strategy uses multiple malicious requesters to coordinate the query sequence and dynamically adjust the query target via the query results of other malicious requesters to query private friendships. However, if access control is given to the requestors in the friend search engine, collusion attacks can be resisted. In this section, we investigate the strategy to resist collusion attacks. In this work, the access control of requestors in the friend search engine is considered, credibility is employed as the restriction condition for requestor queries, and the SSS system (t, n) threshold function is utilized to control queries to the target user.

4.1 Credibility Calculations

Calculation of Direct Trust (DT_{ij}). For two user nodes that have historical interactions in the OSN, the credibility of the first user for the second user is referred to as direct trust. A user obtains the credibility evaluation of another user based on the historical performance of the user who has interacted with him or her. Therefore, the following factors are introduced when calculating direct trust.

Number of Interactions. The greater the number of interactions between two users is, the higher the trust between the users is.

Interaction Evaluation. After each interaction, the user gives a corresponding evaluation based on the process, results, and importance of the interaction event. The evaluation value of the *lth* interaction is recorded as $C_l \in [0, 1]$.

Interaction Time. Interaction evaluations that are similar to the current time better reflect the user's recent behavior. The closer the evaluation is to the current time, the greater is the impact on direct credibility.

Interaction Events. The weight of the event of the *lth* interaction between two users is denoted as W_l.

If node i and node j have interacted n times in the OSN, after the *lth* interaction is completed, node i evaluates node j to obtain evaluation value C_l and interaction event

weight W_l. Subsequently, the *lth* interaction time t_l, importance of the *lth* interaction event W_l, evaluation value C_l of the interaction event of node i with node j, and influence of the number n of interactions between node i and node j on the evaluation value are considered. The calculation formula of direct trust is expressed as follows:

$$DT_{ij} = \alpha \cdot \frac{\sum_{l=1}^{n} \Phi(t_l) \cdot C_l \cdot W_l}{n} \tag{1}$$

where $\alpha = n/(n+1)$ is a function of the number of interactions that is used to adjust the influence of the number of interactions on credibility. The user obtains a high degree of trust only when he or she obtains multiple satisfactory evaluation values. $\Phi\ (t_l) = exp\ (\lceil (t_n - t_l)/T \rceil)$ is the time decay coefficient, where t_n is the n th interaction time (i.e., current interaction time), t_l is the l th interaction time, and T is the time period. The evaluation of an interaction event that is more similar to the current interaction time has a greater impact on credibility. W_l and C_l are the weight of the interaction event between node i and node j and the evaluation value of node i for the event, respectively. This approach can prevent malicious requestors from interacting with the target user by using events with a low weight to gain the trust of the target user while deceiving the user during interaction events with high weights.

Recommended Trust (RT_{ij}). If node i wants to gain a comprehensive understanding of node j, node i needs to obtain the recommended trust for node j via intermediate node c, where node $c = \{c_1, c_2, c_3, ..., c_n\}$. The calculation of recommended trust is expressed as follows:

$$RT_{ij} = \sum_{c=1}^{n} \left(DT_{ic} \cdot DT_{cj}\right) \tag{2}$$

where DT_{ic} is the direct trust of user i in user c, DT_{cj} is the direct trust of user c in user j, and the direct trust of user i in user c can be regarded as a recommendation for calculating the recommended trust weights.

Comprehensive Trust (OT_{ij}). The credibility of a user in the OSN must be integrated with his or her direct trust and the recommended trust of other users, which is referred to as comprehensive trust. The weights of direct trust and recommended trust are determined by experimental calculations. In real life, people are generally more inclined to believe their judgments, and the recommendations of others serve only as a reference. Thus, the calculation of comprehensive trust is expressed as follows:

$$OT_{ij} = u \cdot DT_{ij} + v \cdot RT_{ij} (u + v = 1, u > v) \tag{3}$$

where OT_{ij} is the direct trust of node i in node j, RT_{ij} is the recommended trust of node i in node j, and u and v are the weight coefficients of direct trust and recommended trust, respectively.

4.2 Shamir Secret Sharing System

The SSS system is a specific secret sharing scheme designed by Shamir based on language interpolation polynomial theory [29, 30]. This scheme clearly illustrates how to

divide data D into n segments so that D can be easily reconstructed from t segments and so that even if all $t-1$ segments are mastered, D cannot be reconstructed.

In response to collusion attacks in OSNs, this paper uses the *SSS* (t, n) threshold function to control the querying of users' friendships. The (t, n) threshold secret sharing scheme consists of the following three stages.

System Parameter Setting. n is the number of all participants, t is the threshold, p is a large prime number, and $s \in Z_p$ is the secret to be shared.

Secret Distribution. The secret distributor D chooses a random t degree polynomial.

$$a(x) = s + a_1 x^1 + a_2 x^2 + a_3 x^3 + ... + a_{t-1} x^{t-1} \, mod \, p, \alpha_j \in_R Z_p \tag{4}$$

The condition $a(0) = s$ is satisfied. D sends $s_i = a(i)$ to participants P_i, $i = 1, 2..., n$.

Secret Reconstruction. Any number of participants can reconstruct the secret using their secret fragments. Let t participants who want to reconstruct the secret be P_i, $i = 1, 2..., t$, and let $A = |1, 2..., t|$.

λ_i is calculated based on the following formula:

$$\lambda_i = \prod_{j \in A \setminus \{i\}} \frac{j}{j - i} \tag{5}$$

The original secret is restored based on the following formula:

$$s = \sum_{i \in A} s_i \lambda_i \tag{6}$$

The security of the *SSS* depends on the assumption that the parties honestly perform the operations predetermined by the agreement. We consider reliable secret distributors and believe that the administrators of OSNs are honest in the strategy.

4.3 Friend Search Engine with the SSS System

In OSNs, users can access the friendships of other users by friend search engines. Multiple malicious requestors can share their query results with each other by coordinating the query target and query sequence, which causes the target user to expose more friends than the user is willing to display. A friend search engine that has introduced the trust metric and SSS can control the query of users. This control can guarantee that only users whose comprehensive trust reaches the trust threshold can successfully query the friendships of the target user.

Assume that secret distributor D is honest and that each anonymous requestor $P_i (i = 1, 2, ..., n)$ can obtain a correct secret fragment from D. The number of requestors is higher than the trust threshold for querying the friendships of the target user each time $n_A \geq 2$. The access control process of this solution is described as follows:

Obtain Comprehensive Trust. Requestors $P_i(i = 1, 2..., n)$ request querying the friendships of target user n_a, obtaining comprehensive trust T_{ai} of P_i, and sorting the results in descending order by value based on the interaction between target user n_a and requestor P_i in the OSN.

Classify the Query. Based on trust threshold TR, the requestors are divided into categories A and B. Category A: $T_{ai} \in [TR, 1]$ and category B: $T_{ai} \in [0, TR]$. The number of requestors in the two categories is denoted as n_A and n_B.

Confirm Threshold t. According to the definition of the (t, n) threshold function and the requirements of access control security, requestors who have not reached the system trust threshold cannot successfully query the target user's friendships. Since $T_{ai} < TR$, it is necessary to ensure that requestors in category B cannot successfully query the friendships of the target user. Thus, in each query process, $t = n_B + 1$.

Secret Distribution. The secret distributor D chooses a random t degree polynomial $a(x) = s + a_1x^1 + a_2x^2 + a_3x^3 + ... + a_{t-1}x^{t-1} mod\ p, \alpha_j \in_R Z_p$, $a(0) = S.D$ sends $s_i = a(i)$ to participants P_i, $i = 1, 2..., n$.

Secret Reconstruction. n_B requestors in category B, who are arranged in descending order of comprehensive trust, submit the secret fragments s_i obtained in reverse order, and n_A requestors and n_B requestors are divided into n_A groups for secret reconstruction.

Assume that requestor $P_i(i = 1, 2..., n)$, who queries the friendships of the target user, is arranged in descending order based on the comprehensive trust of the target usern_a. Category A is $P_1, P_2, P_3, \ldots, P_m$, and category B is $P_{m+1}, P_{m+2}, \ldots, P_n$. Threshold $t = n_B + 1$. Category A can be divided into m groups to reconstruct secrets (Table 1).

Table 1. Groups to reconstruct secrets

Group number	Group member
1	$P_1, P_n, P_{n-1}, \ldots, P_{m+1}$
2	$P_2, P_n, P_{n-1}, \ldots, P_{m+1}$
3	$P_3, P_n, P_{n-1}, \ldots, P_{m+1}$
…	……
a	$P_a, P_n, P_{n-1}, \ldots, P_{m+1}$

The comprehensive trust of the first requestor among the m groups of requestors who participate in the secret reconstruction is greater than the trust threshold set by the target user (i.e., only users trusted by the target user can successfully query the target's friendships). During each secret reconstruction process, the users P_{m+1}, P_{m+2}, …, P_n, who have not reached the comprehensive trust level threshold, must submit their secret fragments s_{m+1}, s_{m+2}, …, s_n obtained from D. Users P_1, P_2, P_3, …, P_m will submit s_{m+1}, s_{m+2}, …, s_n. The secret fragment $s_i(i \in [1, m])$ is secretly reconstructed. The threshold $t = n_B + 1$ can ensure that even if P_{m+1}, P_{m+2}, …, P_n constitute the group of submitted secret fragments, the secret cannot be successfully reconstructed.

5 Experiment

In this section, we experimentally verify the effectiveness of the proposed anti-collusion attack strategy. Our experimental research includes synthetic datasets to verify the validity of the credibility calculations and three large-scale real-world datasets to verify the security of the anti-collusion attack strategy.

5.1 Datasets

We generate random numbers that satisfy the previously described conditions of the credibility calculation method, including data on 1000 groups of user interactions, and verify the correctness of the trust calculations. In addition, we use three real-world social network datasets to verify the security of the anti-collusion attack strategy.

Synthetic Dataset. A random probability function is used to fit users' interactions in OSNs. The setting standards for the time interval of interactions between users and the weights of the interaction events are different for each OSN. We select the interaction data within the time interval ($\Phi(t_l) = 0.367879$) among users in the synthetic dataset. The number of interactions is set to 50; the weights of the interaction events take values in the range [1, 20]; and the interaction evaluation takes values in the range (0, 1] as an example to verify the rationality of the trust calculations, that is, $W_l \in [1, 20]$, $C_l \in (0, 1]$, and $n \in [1, 50]$. The trust between users may exceed 1 and should be normalized.

Facebook Dataset [31]. The data from Facebook.com capture the friendships among users, which can be modeled as undirected graphs.

Slashdot Dataset [32]. Slashdot is a technology-related news website and a specific user community, where users can submit and edit news about the current main technology. In 2002, Slashdot launched the Slashdot Zoo function, which enables users to mark each other as friends or enemies. The network establishes links between two friends or enemies among Slashdot users. Therefore, the data in this dataset are directional. This article uses 2009 Slashdot data, and the Slashdot dataset is converted to an undirected graph to reflect users' friendships. Regardless of the direction of the connection between two nodes in the network, an edge is created in the undirected graph for these two nodes.

Gowalla Dataset [33]. Gowalla is a location-based social networking site, where users share their location by signing in. The friendships collected from Gowalla are undirected. The complete dataset consists of 19,591 nodes and 950,327 edges. Due to data size limitations, this program selects only a portion of the data for testing.

We list the main attributes of each dataset in Table 2. The synthetic dataset is used to verify the rationality of the credibility calculations, and the remaining three datasets are used to verify the security of the proposed anti-collusion attack strategy.

Table 2. Social network dataset property

Dataset	Synthetic dataset	Facebook	Slashdot	Gowalla
Vertices	1000	63731	82168	196591
Edges	8997	817090	948464	582533
Average degree	____	25.773	12.273	9.668

5.2 Comparison Plan

As the collusion attack problem in friend search engines is more advanced than other attack problems, we have not obtained any relevant solutions to resist these collusion attacks. Therefore, we applied this strategy to the collusion attack and compared it with the original collusion attack. For the collusion attack strategy [3], we applied the proposed anti-collusion attack strategy. It is assumed that colluding attackers can successfully destroy the friendship privacy of users in every query if (t, n) threshold function access control is not adopted (i.e., probability that the colluding attackers successfully destroy the user's privacy is 1). The cases of using and not using (t, n) threshold function access control is compared to analyze the performance of the anti-collusion attack strategy proposed in this paper.

5.3 Performance Analysis

In this section, we analyze the rationality of the trust calculations and the security of the anti-collusion attack strategy using (t, n) threshold function access control.

Credibility Calculation Rationality. Based on random numbers, the values of direct trust and recommended trust are calculated by formula (1) and formula (2), and the value of the user's comprehensive trust is calculated by formula (3). We selected 1000 sets of data to prove the correctness of the trust calculations. The results are shown in Fig. 2.

Figure 2 shows that the results of the comprehensive trust calculations are normally distributed. In addition, they are consistent with realistic expectations.

Security Analysis. To improve the security and usability of the friend search engine, we assume that OSN administrators can be fully trusted in regard to the friend search engine. When the number of requestors is less than the number of query requests, the administrators can help the requestors complete the query. In this section, we conduct a security analysis based on four aspects: limit rate, number of users whose privacy has been violated, number of colluding attackers and success probability of collusion attacks.

Limit Rate (LR). The LR of the system is defined as the ratio of the number of users in category B to the number of all users, that is, the proportion of users who cannot successfully query in the friend search engine among all requestors. Based on formula (3) $OT_{ij} = u \cdot DT_{ij} + v \cdot RT_{ij} (u + v = 1, u > v)$, where $DT_{ij}, RT_{ij} \in [0, 1]$. The direct

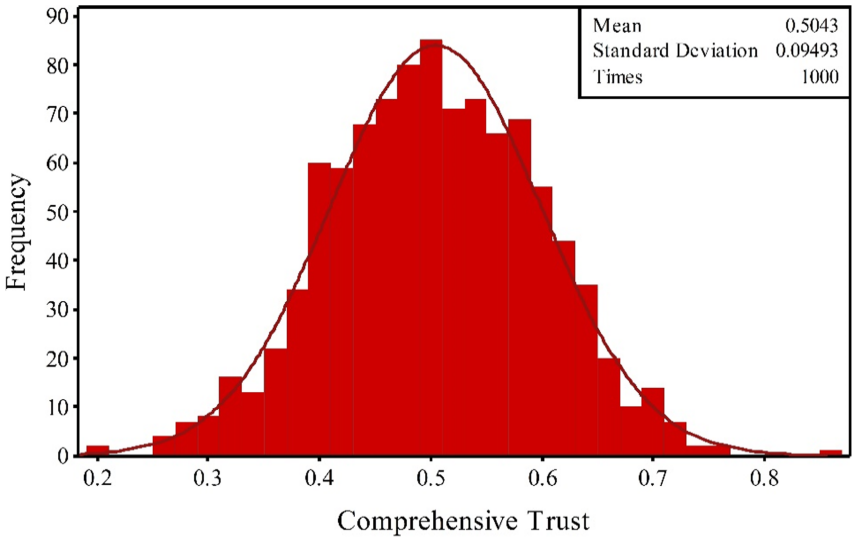

Fig. 2. Comprehensive trust values

trust weight coefficient u is set to 0.6, and the trust threshold is set to 0.5, 0.6, 0.7, 0.8, and 0.9. A total of 1000 experiments are conducted to verify the LR of the proposed strategy.

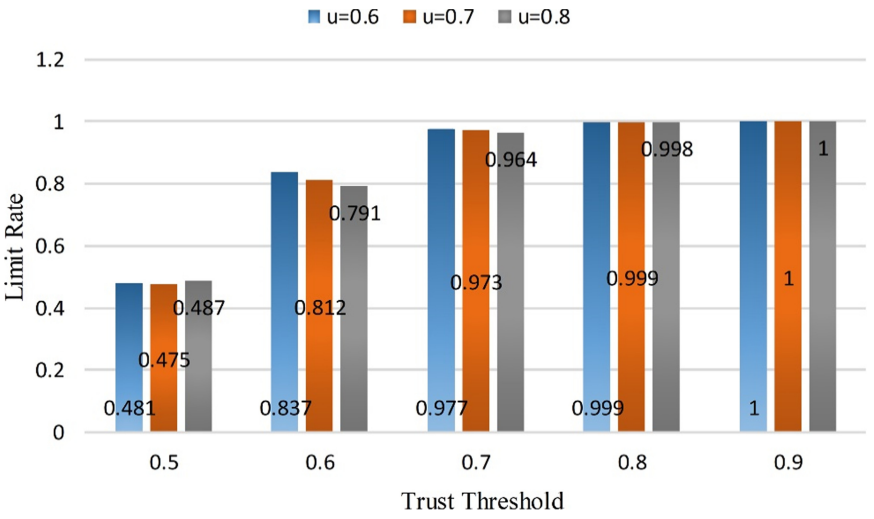

Fig. 3. Limit rate under the trust threshold with $u = 0.6$, $u = 0.7$ and $u = 0.8$

Figure 3 shows the LR and trust threshold results of this strategy. The value of the direct trust weight coefficient u are 0.6. When the trust threshold is 0.5, the LR of the strategy is approximately 40%. When the trust threshold is 0.6, the LR increases to 80%.

At 0.7, the LR increases to almost 100%; therefore, when the trust threshold is 0.7, almost no user reaches the trust threshold, and the friend search engine will not allow any querying. When the trust threshold is 0.6, 80% of users in the OSN cannot reach the threshold; thus, the number of requestors in the friend search engine is limited, and the safety of the friend search engine is increased.

Number of Users Whose Privacy Has Been Violated. Consider the trust threshold of 0.5 as an example. Sixty percent of users can make normal queries. In the worst case of the friend search engine query, the number of malicious requestors is not limited, and malicious requestors can destroy the privacy of the target user via a one-time collusion attack at the first layer. The probability of successfully destroying the user's privacy is $P(N_0|_{l=1}) = 1 - \prod_{i=1}^{k} \min\left(\frac{k}{d}, 1\right)$, where k is the maximum number of friends allowed to be displayed by the user and d is the degree of the node. The attack can destroy the privacy of 80% of the nodes in OSNs [3].

In a one-time collusion attack, the maximum number of malicious requestors is $n_A - 1$, and the collusion attack performs at least two queries. Thus, the probability of one collusion attack that destroys the target user's privacy at the first layer is $\left(\frac{n_A-1}{n_A}\right)^2 \cdot P(N_0|_{l=1})$. When the trust threshold is set to 0.5 (lowest threshold), 40% of users' queries will be restricted. In this case, the anti-collusion attack strategy can reduce the number of users whose privacy is breached by at least 47.9%. Accordingly, the number of users whose privacy is violated decreases. By comparing the Facebook, Gowalla and Slashdot datasets, we obtain the results shown in Fig. 4.

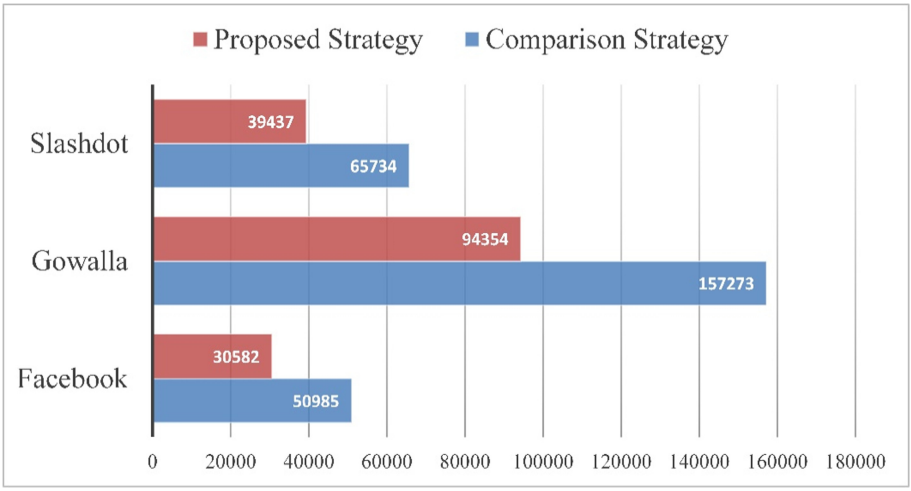

Fig. 4. Comparison of the number of users with a compromised strategy

Due to the limitation of the trust threshold, the number of users whose privacy is breached is significantly reduced. The number of users whose privacy is breached in the Facebook and Slashdot datasets is reduced by approximately 20,000, while the number

of users whose privacy is breached in the Gowalla dataset is reduced by approximately 60,000. In the three datasets, the number of users whose privacy has been violated will be reduced by at least 40%. The proposed strategy greatly reduces the number of users whose privacy is violated, which improves the privacy security of users in OSNs.

Number of Colluding Attackers. Based on the (t, n) threshold function, in the query process of the friend search engine, n inquirers are required to participate in the query, and at least t requestors are required to perform secret reconstruction. Therefore, in a single query process, to ensure that malicious requestors can successfully query, it is necessary to ensure that t requestors are malicious requestors and that the comprehensive trust is higher than the trust threshold. In the best situation, two malicious requestors can destroy the privacy of the target user by making two queries. The total number of attackers required is $2n$, while in the comparison strategy, the number of inquirers required is only 2. Therefore, when the value of n set by the system is larger, more malicious attackers will be needed.

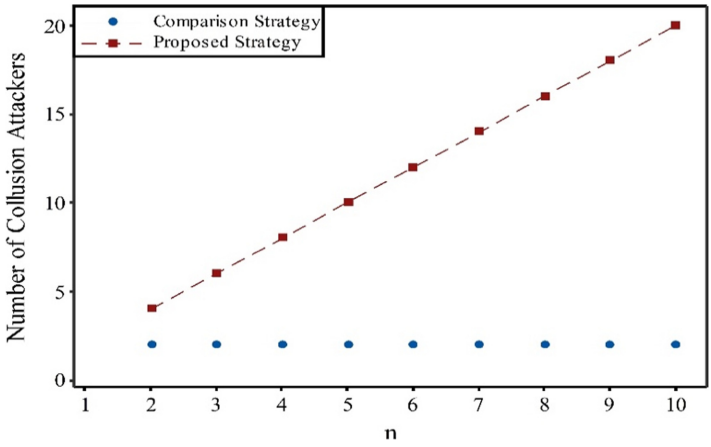

Fig. 5. Number of colluding attackers

Figure 5 shows that the number of colluding attackers varies with the number of queries n. The number of attackers in the proposed strategy is twice that of the comparison strategy. Under the same conditions, the colluding attackers will need more entities or accounts to make queries with the proposed strategy.

Probability of a Successful Collusion Attack. Assume that malicious requestors who have not interacted with the target user in the OSN want to query the target's friendships. First, long-term excellent interactions with the target are needed to obtain the trust of the target. A successful collusion attack requires multiple malicious requestors to cooperate to coordinate their query order and target, and each malicious requestor can successfully query the friend list of the query target. Therefore, multiple malicious requestors need to maintain excellent interactions with users in the OSN, which will require colluding malicious requestors to spend a substantial amount of time disguising their intentions to obtain the trust of the target user.

Consider the following attack [3] as an example of a successful collusion attack.

MR_1: Query N_0—> Retrieve $E(N_0, N_{0.1})$, $E(N_0, N_{0.2})$, $E(N_0, N_{0.3})$
MR_2: Query $N_{0.2}$—> Retrieve $E(N_{0.2}, N_{0.2.1})$, $E(N_{0.2}, N_0)$, $E(N_{0.2}, N_{0.2.3})$
MR_3:
Query $N_{0.2.1}$—> Retrieve $E(N_{0.2.1}, N_{0.2.1.1})$, $E(N_{0.2.1}, N_{0.2})$, $E(N_{0.2.1}, N_{0.2.1.2})$
Query $N_{0.2.2}$—> Retrieve $E(N_{0.2.1}, N_{0.2.1.1})$, $E(N_{0.2.1}, N_{0.2.2.2})$, $E(N_{0.2.1}, N_{0.2.2.3})$
// $N_{0.2.2}$ is "occupied"
Query $N_{0.2}$—> Retrieve $E(N_{0.2}, N_{0.2.1})$, $E(N_{0.2}, N_0)$, $E(N_{0.2}, N_{0.2.3})$
// find new friends:$N_{0.2.3}$
MR_4:
Query $N_{0.2.3}$—> Retrieve $E(N_{0.2.3}, N_{0.2.3.1})$, $E(N_{0.2.3}, N_{0.2})$, $E(N_{0.2.3}, N_{0.2.3.2})$
// $N_{0.2.3}$ can be passive show $N_{0.2}$
Query $N_{0.2}$—> Retrieve $E(N_{0.2}, N_{0.2.1})$, $E(N_{0.2}, N_{0.2.2})$, $E(N_{0.2}, N_{0.2.3})$
// $N_{0.2}$ is occupied
Query N_0—> Retrieve $E(N_0, N_{0.1})$, $E(N_0, N_{0.3})$, $E(N_0, N_{0.4})$
This collusion attack successfully destroyed N_0's privacy.

The collusion attack was coordinated by four malicious requestors. MR_1 makes the first request, and MR_2 determines the target to be queried based on the query results of MR_1. MR_3 queries based on the query result of MR_2. Thus, user $N_{0.2.2}$ will be "occupied," and the new friend $N_{0.2.3}$ of user $N_{0.2}$ can be queried. MR_4 makes a query based on the query result of MR_3 and obtains the $k + 1th$ friend of N_0, i.e., fourth friend $N_{0.4}$. The privacy of the friendships of user N_0 is destroyed.

Under (t, n) threshold function access control, four malicious requestors, i.e., MR_1, MR_2, MR_3, and MR_4, want to complete this query. First, they need to obtain the high trust of the target nodes, i.e., N_0, $N_{0.2}$, $N_{0.2.1}$, $N_{0.2.2}$, and $N_{0.2.3}$, and all four malicious requestors must have long-term excellent interactions with the target. If a malicious requestor cannot obtain the trust of the target, then $T_{ij} < T_t$, and the previously described attack cannot be successfully carried out. Therefore, a successful malicious attack by colluding attackers requires that all malicious requestors reach the trust threshold.

If malicious requestors already exist in the OSN and have interacted with the target user, this strategy restricts requestors whose trust level is below the trust threshold. A requestor cannot query the target user's friend list under (t, n) threshold function access control. Therefore, when the trust threshold is 0.5, 40% of users who do not reach the trust threshold will not be able to query. As described in the second part of this section, for the collusion attack strategy in [3], if (t, n) threshold function access control is not adopted, the probability that colluding attackers will successfully destroy a user's privacy is 1 for each query. In the (t, n) threshold secret sharing anti-collusion attack strategy combined with trust, the comprehensive trust of the requestors who can successfully query the friendships of the target user must be higher than the trust threshold, that is, malicious requestors need to be in category A. Next, we take the trust threshold of 0.5 as an example to discuss the probability that colluding attackers will successfully destroy the privacy of a user's friendships under (t, n) threshold function access control.

If there is a collusion attack, the worst case is that there are enough colluding attackers, and the privacy of the target user is destroyed by just two queries. During a single query, the maximum number of malicious requestors is $n_A - 1$. The maximum probability of malicious requestors who make two requests is $[0.6^{(n_A-1)} \cdot (\frac{n_A-1}{n_A})]^2$.

In Facebook, Gowalla and Slashdot, we observe that regardless of whether a popular node or an unpopular node is considered, the number of malicious requestors required to conduct a successful collusion attack can reach 10,000, which is the best case of a successful collusion attack in the three datasets. Therefore, in the case of $n_A \geq 2$, when there are at most $n_A - 1$ malicious requestors, the probability of a successful collusion attack $p \leq 0.09$.

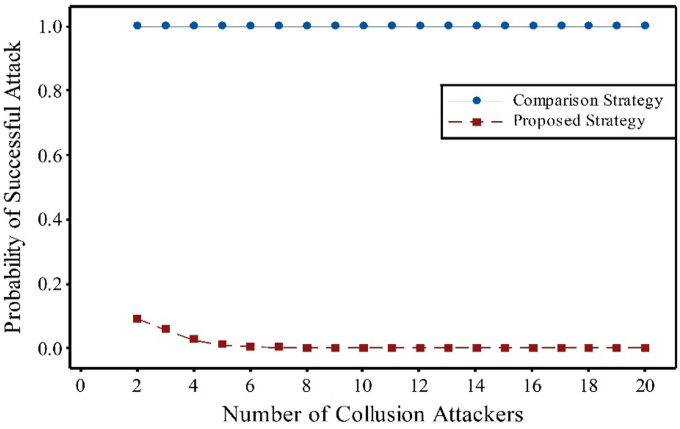

Fig. 6. Comparison of the probabilities of a successful collusion attack based on the number of colluding attackers

Figure 6 shows that when the number of malicious requestors is 2, the anti-collusion attack strategy based on (t, n) threshold secret sharing can reduce the probability of a successful collusion attack from 1 to 0.09. When the number of malicious requestors increases to 18, the anti-collusion attack strategy reduces the probability of a successful collusion attack to 0. When the system trust threshold is higher, it is more difficult for malicious requestors to conduct collusion attacks.

Therefore, the trust-based SSS anti-collusion attack strategy proposed in this work can substantially reduce the number of users whose privacy is compromised by means of credibility calculations, the trust threshold and the (t, n) threshold function. This strategy restricts user queries based on trust and uses the (t, n) threshold function of the SSS system for access control. This strategy can also reduce the probability of successful collusion attacks, which has a significant effect on resisting collusion attacks and can protect the friendship privacy of users in OSNs.

6 Conclusion

In this work, we propose an anti-collusion attack strategy based on trust and the SSS system (t, n) threshold. By analyzing the behaviors of users in OSNs, a method for calculating the credibility between two users is proposed. Comprehensive trust is calculated based on direct trust and recommended trust, and the comprehensive trust degree is employed as a control condition for querying in the friend search engine. The (t, n) threshold function is used for access control in the friend search engine. For users that already exist in the OSN, only requestors who satisfy the trust threshold can successfully query the friendships of the target user. The experimental results show that the proposed strategy can ensure that general users in the friend search engine can query normally and can greatly reduce not only the number of users whose privacy is destroyed but also the probability of successful collusion attacks initiated by malicious requestors by coordinating the query order and sharing the query results.

Theoretically, this work simplifies the complex privacy protection of a user's friendships to the user's access control strategy in the friend search engine. This research starts by theoretically analyzing the calculation of trust between two users and applies the (t, n) threshold function to control querying in the friend search engine to protect the privacy of the users' friendships.

Overall, the proposed strategy can successfully decrease the probability of collusion attacks in friend search engines. Specifically, attacking the same number of users requires more attackers, and the number of users who violate the same number of attackers is greatly reduced.

References

1. Maradapu, A., Venkata, V., Li, Y.: A survey on privacy issues in mobile social networks. IEEE Access **8**, 130906–130921 (2020)
2. Na, L.: Privacy-aware display strategy in friend search. In: 2014 IEEE International Conference on Communications (ICC), pp. 945–950 (2014)
3. Yuhong, L., Na, L.: Retrieving hidden friends: a collusion privacy attack against online friend search engine. IEEE Trans. Inf. Forensics Secur. **14**(4), 833–847 (2019)
4. Xiong, J., et al.: A personalized privacy protection framework for mobile crowdsensing in IIoT. IEEE Trans. Ind. Inf. **16**(6), 4231–4241 (2020)
5. Tian, J., Du, R., Cai, H.: Trusted Computing and Trust Management. Science Press, Beijing (2014)
6. Qiu, W., Huang, Z., Li, X.: Basics of Cryptographic Protocol. Higher Education Press, Beijing (2009)
7. Amusan, O., Thompson, A., Aderinola, T., Alese, B.: Modelling malicious attack in social networks. Netw. Commun. Technol. **5**(1), 37 (2020)
8. Elyashar, A., Uziel, S., Paradise, A., Puzis, R.: The chameleon attack: manipulating content display in online social media. In: Proceedings of the Web Conference 2020, vol. 2, pp. 848–859 (2020)
9. DasGupta, B., Mobasheri, N., Yero, I.G.: On analyzing and evaluating privacy measures for social networks under active attack. Inf. Sci. **473**, 87–100 (2019)
10. Mei, B., Xiao, Y., Li, R., Li, H., Cheng, X., Sun, Y.: Image and attribute based convolutional neural network inference attacks in social networks. IEEE Trans. Netw. Sci. Eng. **7**(2), 869–879 (2020)

11. Muyuan, L., Haojin, Z., Zhaoyu, G., Si, C., Le, Y., Shangqian, H.: All your location are belong to us: breaking mobile social networks for automated user location tracking. In: Proceedings of the International Symposium on Mobile Ad Hoc Networking & Computing, pp. 43–52(2014)
12. Zhipeng, C., Zaobo, H., Xin, G., Yingshu, L.: Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans. Dependable Secur. Comput. **15**(4), 577–590 (2018)
13. Binghui, W., Jinyuan, J., Le, Z., Gong, N.Z.: Structure-based Sybil detection in social networks via local rule-based propagation. IEEE Trans. Netw. Sci. Eng. **14**(8), 523–537 (2018)
14. Qingqing, Z., Guo, C.: An efficient victim prediction for Sybil detection in online social network. IEEE Access **8**, 123228–123237 (2020)
15. Tianyu, G., Jin, Y., Wenjun, P., Luyu, J., Yihao, S., Fangchuan, L.: A content-based method for Sybil detection in online social networks via deep learning. IEEE Access **8**, 38753–38766 (2020)
16. Xingwen, Z., Hui, L.: Privacy preserving data-sharing scheme in content-centric networks against collusion name guessing attacks. IEEE Access **5**, 23182–23189 (2017)
17. Chen, Z., Tian, Y., Peng, C.: An incentive-compatible rational secret sharing scheme using blockchain and smart contract. Sci. China Inf. Sci. **64**(10), 1–21 (2021). https://doi.org/10.1007/s11432-019-2858-8
18. Linke, G., Chi, Z., Yuguang, F.: A trust-based privacy-preserving friend recommendation scheme for online social networks. IEEE Trans. Dependable Secur. Comput. **12**(4), 413–427 (2015)
19. Mao, C., Xu, C., He, Q.: A cost-effective algorithm for inferring the trust between two individuals in social networks. Knowl.-Based Syst. **164**, 122–138 (2019)
20. Hamzelou, N., Ashtiani, M.: A mitigation strategy for the prevention of cascading trust failures in social networks. Future Gener. Comput. Syst. **94**, 564–586 (2019)
21. Fogues, R., Such, J.M., Espinosa, A., Garcia-Fornes, A.: Open challenges in relationship-based privacy mechanisms for social network services. Int. J. Hum. Comput. Interact. **31**(5), 350–370 (2015)
22. Huaxin, L., Haojin, Z., Suguo, D., Xiaohui, L., Xuemin, S.S.: Privacy leakage of location sharing in mobile social networks: attacks and defense. IEEE Trans. Dependable Secur. Comput. **15**(4), 646–660 (2018)
23. Alrubaian, M., Al-Qurishi, M., Alamri, A., Al-Rakhami, M., Hassan, M.M., Fortino, G.: Credibility in online social networks: a survey. IEEE Access **7**(c), 2828–2855 (2019)
24. Akilal, K., Slimani, H., Omar, M.: A robust trust inference algorithm in weighted signed social networks based on collaborative filtering and agreement as a similarity metric. Netw. Comput. Appl. **126**(March 2018), 123–132 (2019)
25. Akilal, K., Slimani, H., Omar, M.: A very fast and robust trust inference algorithm in weighted signed social networks using controversy, eclecticism, and reciprocity. Comput. Secur. **83**, 68–78 (2019)
26. Friending Facebook: 13 million US Facebook users don't change privacy settings. http://www.zdnet.com/article/13-million-us-facebook-users-dontchange-privacy-settings/. Accessed 16 Mar 2019
27. Bonneau J, Anderson J, Anderson R, Stajano F.: Eight friends are enough: social graph approximation via public listings. In: Proceedings of the 2nd ACM EuroSys Workshop on Social Network Systems SNS 2009, pp. 13–18 (2009)
28. Malka, S.S., Li, N., Doddapaneni, V.M.: A web application for studying collusion attacks through friend search engine. In: Proceedings - International Computing Software Application Conference, pp. 388–393 (2016)
29. Shamir, A.: How to share a secret. Commun. ACM **22**, 612–613 (1979)
30. Dawson, E., Donovan, D.: The breadth of Shamir's secret-sharing scheme. Comput. Secur. **13**(1), 69–78 (1994)

31. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in Facebook. In: SIGCOMM 2009 – Proceedings of the 2009 SIGCOMM Conference on Co-Located Workshops, Proceedings of 2nd ACM Work Online Social Networks, WOSN 2009, pp. 37–42 (2009)
32. Leskovec, J., Lang, K., Dasgupta, A., Mahoney, M.: Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. Internet Math. **6**(1), 29–123 (2009)
33. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: The 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1082–1090. ACM (2011)