Social Authentication Identity: An Alternate to Internet Real Name System

Chengcheng Shao(✉), Liang Chen, Shuo Fan, and Xinwen Jiang

National University of Defense Technology, Changsha 410073, Hunan, China
sccotte@gmail.com, chl160@163.com, {fanshuo_ben,xinwenjiang}@sina.com

Abstract. Rumors and defamation are now becoming a main threat to Online Social Networks (OSNs). To prevent them, Real Name System (RNS) was proposed, but has been proved vulnerable by the data leakage in South Korea. In this paper, we propose a new identity model, Social Authentication Identity (SAI), to trace rumor-makers. In SAI, only a small number of users (called *root*s) are required to be authenticated by RNS. And the others are authenticated by vouching of friends, called social authentication. We evaluate factors that affect the efficiency of SAI. Results show that selecting *root*s in communities are the best strategy, comparing with random and maximum degree strategies. We also provide an social tracing mechanism to trace down rumor-makes. Analysis shows our social tracing is robust enough to defend Sybil attacks.

Keywords: Online social network · Social authentication · Real name system · Sybil defend · Network surveillance

1 Introduction

Recent years have witnessed the explosion of Online Social Networks (OSNs). According to Statistic Brain [1]: facebook now owns more than 1.31 billion users. While we have seen the power of OSNs in the fields of information sharing and social media, it is noticeable that baseless rumors, personal defamation and privacy invasion are becoming an emergent threat to our life. Anonymity, once was considered as the essential nature of the Internet, now becomes a nightmare to the security of OSNs. When attacking, attackers try to register virtual identities or stealing others' identities. So it's quit difficult to trace them down. Worse more, new security threats are coming along with the booming of OSNs. Large degree nodes are tricked to send rumors or distribute viruses. Well organized nodes act as Sybil nodes [2] to guide or distort opinions of polls or reviews of products.

Therefore, the Real Name System (RNS) was proposed. RNS performs like an map that maps national identity (offline) with virtual identities (online). Often RNS works as a center server, adopting a schema called 'anonymity in foreground

Project supported by the National Natural Science Foundation of China (Grant No. 61272010).

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 132–140, 2015.
DOI: 10.1007/978-3-319-23829-6_10

and real-name in background', meaning virtual names are used to surf the OSN, while real names must be provided when registering. It's quit reasonable to use RNS in financial transaction where high security are required. However, it sounds harsh to submit real names in OSN. Nevertheless, South Korea becomes the first to try RNS, which unfortunately ends with leaking more than 35 million identities and being forced terminated. Many studies are done on the effect of Real Name Verification Law in South Korea. The empirical analysis of Oh et al. [3] shows that the alternative RNS (i-pin) is still vulnerable to phishing attack. Findings of Cho [4] suggest that Real Name Verification Law has a dampening effect on overall participation in short-term, but not in long term. Again Cho et al. find that RNS has significant effect on reducing uninhibited behaviors at the aggregate level, but no significant impact on behavioral shift of a particular user [5]. Though it is not certain whether RNS has the capability to defend rumors, it's quite clear that RNS is vulnerable to protect personal information.

Verifying a user through his national identity is actually a kind of identity authentication. Traditionally three factors, including something you have (e.g., a hardware token) [6], something you are (e.g., a fingerprint) [7,8], and something you know (e.g., a password) [9] are used in computer authentication. Brainard et al. [10] introduce the fourth factor, somebody you know, known as social authentication. Following works are: Schechter et al. [11] build a backup authentication among trustees and Zhan et al. [9] enhance social authentication by divide social relations apart. However, all these works are base on offline relations, where people have face-to-face contact. And then, we're wondering is it viable to applying social authentication in OSN, where 'no one knows you'r a dog'. Fortunately, many studies suggest that there are enough trusted online relations. In [12], Boyd and Ellison observe that most links made in OSN have offline relations. Other researches also suggest links from OSN indicate trusted relations [13]. In a word, it's quit feasible to conduct social authentication in OSN.

1.1 Contribution and Organization

Contribution. In this paper, we introduce an online identity model called Social Authentication Identity (SAI) by exploiting online social relations. In SAI, only a small number of nodes are required to be authenticated by RNS and the others are authenticated by friends, so that it's quit appropriate to replace RNS in network surveillance. Firstly, we proposed a simple vouching protocol to implement authentication between friends (social authentication). Then we discussion how to select *root*s and evaluate factors that affect the efficiency of the SAI. Our results show that selecting *root*s in communities are the best strategy, comparing with strategies like selecting by random and selecting by maximum degree. And lastly, we provide an social tracing mechanism to trace down rumor-makes. Analysis shows our social tracing is robust enough to defend Sybil attacks.

Organization. Our SAI model is introduced in Sect. 2, including how to authenticate (Sect. 2.1), how to select *root*s (Sect. 2.2) and how to build an identity

(Sect. 2.3). In Sect. 3, we discuss the prorogation of authentication and find that selecting *root*s in community is the best strategy. In Sect. 4, we propose a social tracing mechanism and analysis its capability to defend Sybil attack. And in the last Sect. 5, we make a conclusion of our work.

2 Social Authentication Identity Model

In this section we introduce our Social Authentication Identity (SAI) model. First, let's pay attention to the following two common characters in social networks. (a) Your friends could identify you (a local view). (b) You would tend to trust the one who is a friend of your friend, even though your know nothing about him (highly relies on (a)). SAI model takes idea from both of them and neither is dispensable. In SAI, we first establish strong ties: edges that both ends could identify each other through social knowledge will be keep, otherwise be removed. This step takes ideas from character (a) and is accomplished in Sect. 2.1. Then we establish strong paths and build social authentication identity. This step takes ideas from character (b) and is accomplished in Sects. 2.2 and 2.3.

In SAI, there a small number of special *root*s which are mainly authenticated by RNS. Others are authenticated by social authentication. In a view of management, this is a kind of distributed authentication where only *root* are authenticated by center server. Compared with RNS, personal information now stores in the brain of the friends of everyone. The name social authentication comes from the fact that friends authenticate each other using their social knowledge. In social authentication, each user selects friends from his neighbors, then he exchanges and verifies social knowledge between his friends. If a couple of friends could identify each other by social knowledge, we say they pass social authentication. Through social authentication, these authenticated nodes and edges become reliable.

2.1 Social Authentication Between Friends

Social authentication is used to establish strong ties in OSN. To determine who is your best friends, $Server$ (that provides social network service) first filters neighbors of u by their daily behaviors denoted as $neighbors_{server}(u)$, and then u choose friends from $neighbors_{server}(u)$, denoted as $friends(u)$. Our Social Authentication is implemented by vouching, a peer-level human-intermediate authentication. The following part provides a simple vouching protocol.

Authentication Parties. The principal parties involved in the social authentication are $Asker$, $Helper$ and the $Server$. (a) $Asker$ is the invoker of the authentication. (b) $Helper$ is responsible to authenticate $Asker$. (c) $Server$ is responsible to arbitrate the authentication. Both $Asker$ and $Helper$ should be valid $User$ and they almost play the same role. $Asker$ can be authenticated by $Helper$, if and only if $Helper$ can be authenticated by $Asker$. The reasons

Table 1. *PRI* data item

Visible	Invisible
Name: your real name	Personal Q&A About Yourself: age, gender, favorite and etc.
Relation Type: the relation type between you and the receiver	Social Q&A of The Type Specified Relation: e.g. for schoolmate relation, question may be school, major and etc.

that we distinct *Asker* and *Helper* apart, one is that it's convenient to describe the protocol, and the other one is that some appropriate incentive mechanism can be applied to *Asker* to stimulate more invokers and eventually speedup the authentication process of the whole network. Additionally, we call an party as sender if it sends data and receiver if receives. All parities can be act as senders or receivers.

Authentication Data Items. Authentication between *Asker* and the *Helper* is based on their social knowledge about each other. We define a data item called Person & Relation Information (*PRI*) to describe it. *PRI* is a list of Questions and Answers (Q&A), where questions are always visible but answers are divided into visible and invisible part (see Table 1). Answer visible part is used to make the receiver identify the sender. Answer invisible part works as 'challenge and response': receiver has to answer questions with his social knowledge and *Server* is responsible to check the answer. When *Asker* or *Helper* passes the challenge, *Server* sends each of them a security *code* as another challenge, and they must exchange their own *code* and submit to *Server* to verify the challenge.

Simple Vouching Protocol. The vouching protocol shows in Fig. 1. (a) *Asker* sends his *PRI* to *Helper* and *Server*. (b) *Helper* answers the *PRI* and sends result back to *Server*. (c) *Server* checks the received answer and if passed, sends a security code $code_1$ to *Helper*. For *Helper*, it requires the similar operations, showing in Fig. 1 (d), (e), and (f). (g) Then *Asker* and *Helper* should exchange the security code. (h) After exchange, *Asker* and *Helper* send exchanged security code to *Server*. (i) *Server* determines whether *Asker* and *Helper* get correct security code. If yes, *Sever* confirms that the authentication between *Asker* and *Helper* has passed.

Note that If *Helper* forget something about *Asker*, so that he cannot answer the received *PRI*. At this moment, *Helper* could get help from *Asker* through social contact stealthily (step (h)). To gain the verification from server, users have to collect a certain number of passed vote from friends.

2.2 Select Root Nodes

When discussing behavior tracing, we need to identify the online user that commits the malicious behavior firstly, and then trace down the real identity of

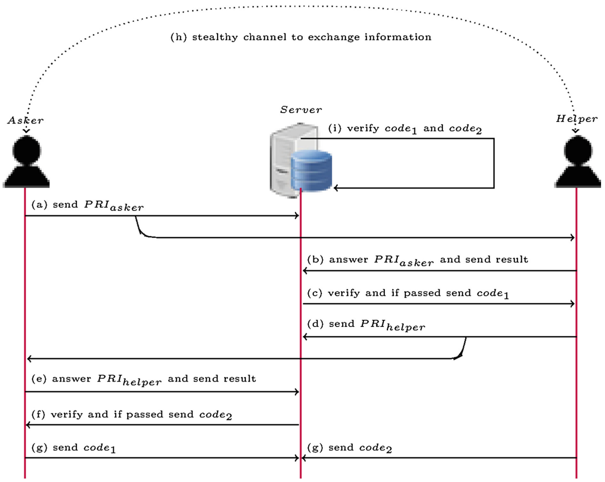
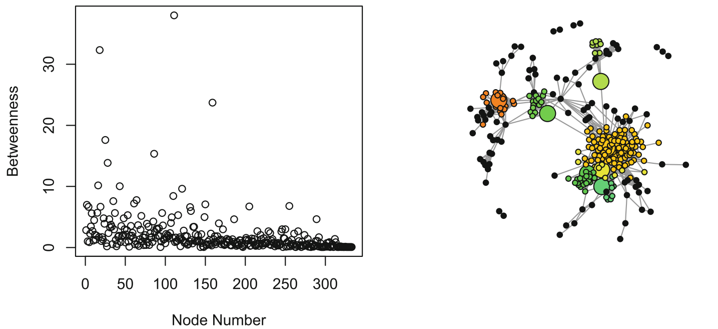

(h) stealthy channel to exchange information
Asker
Server
Helper
(i) verify $code_1$ and $code_2$
(a) send PRI_{asker}
(b) answer PRI_{asker} and send result
(c) verify and if passed send $code_1$
(d) send PRI_{helper}
(e) answer PRI_{helper} and send result
(f) verify and if passed send $code_2$
(g) send $code_1$
(g) send $code_2$

Fig. 1. Simple vouching protocol

that guy. The real identity is an offline identity that government can use it to catch the certain person. Here we refer it to national id, denoted as RI. Our SAI cannot identify who commit malicious behaviors, but can track down the RI of the bad guy by social tracing (Sect. 4). To achieve this purpose, we trace along the paths called $path_{root}$ from $root$ to the bad guy to get the RI of the bad.

We hope $root$s have these properties. (a) High reliability, implying less likely to be Sybil nodes. Metric to measure it is online behaviors. (b) High influence, implying faster propagation of authentication. Centrality (e.g. degree, closeness, betweenness and etc.) could be the metric. (c) Low sensitive to RI. Since $root$s are mainly authenticated by RNS, they face the risk of information leakage. This property means less problems will be caused to the user when his RI is leaked. It is hard to quantify the property, we assume nodes owned by famous persons are low sensitive to RI leakage, because most of these people's information have already been dug to public.

Since $root$ is authenticated RI, the less number of $root$, the less risk of information leakage. However, the less number of $root$ means the longer of $path_{root}$, causing SAI less reliable. To balance them, we must consider the distribution of $root$s. An useful method is to choose $root$s from different communities. Set $num_{community}$ as the number of communities in G, $size_{community}$ as the average size of communities. We can simply select one $root$ in small or medium community and two or more $root$s from large community. By this way, we could control percentage of num_{root} in whole network by community amount and size. Figure 2 is a example of how to select $root$s according to their importance in communities.



(a) Distribution of D+B

(b) Root Set Selection by Communities

Fig. 2. Example of *roots* selection in communities (Color figure online)

The graph is sampled from facebook and communities are detected by walk trap algorithm [14]. For simpleness we defined the importance of nodes u as

$$db(u) = degree(u)/degree(G) + betweenness(u)/betweenness(G). \qquad (1)$$

Degree is capable to measure the power of a node in a local effect and betweenness can measure both global and local impact. So we use $db(u)$ as a mixed metric to measure the importance of u (db distribution Fig. 2(a)). Communities are separated by different color. We ignore communities with size less than 10 (black color) which take 94/333 = 28 % part (94, ignored part and 333, total amount of nodes). The selected *roots* are determined by maximum value of db in each community and highlighted with large size (Fig. 2(b)).

2.3 Building the Social Authentication Identity

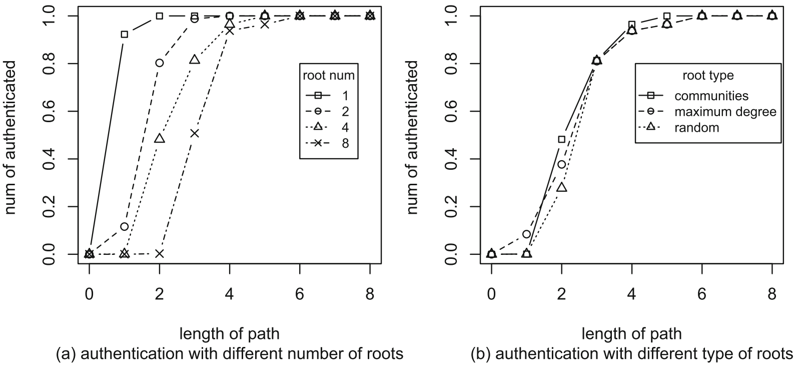
When building the SAI, we should keep the capability of tracing with essential information, that is to say we can recreate $path_{root}$ from SAI with limit information. Here is a very simple schema of SAI we design.

$$SAI(u) = [\sum_{k=1}^{2}(friends_{step=k}(u))][depth][root_{rch}][root_{min\_num}] \qquad (2)$$

The symbol '[ ]' is used to separate SAI. (a) Part $[\sum_{k=1}^{2}(friends_{step=k}(u))]$ works as a local view of u. $friends_{step=k}(u)$ refers to friend nodes who have a shortest path length k to u. So part (a) means $friends(u) \cup friends(friends(u))$. (b) Part $[depth][root_{rch}][root_{min\_num}]$ works as global view of u. From u goes *depth* steps to collect *root* as $[root_{rch}]$ while the number of $[root_{rch}]$ should be large than $[root_{min\_num}]$. When tracing, $[depth]$ indicates the depth, $[root_{rch}]$ indicates the ending *roots*, and $[root_{min\_num}]$ indicates the strength. For *roots* themselves, we can ignore other parts except (a), since they have already been authenticated by RNS.

3 Propagation of Social Authentication

Before a $path_{root}$ could be established, all edges on the path should be already. As we have discussed, a SAI could be built only when satisfying the requirement of $depth$, $root_{rch}$ and $root_{min\_num}$. In order to estimate the effectiveness of SAI, we conduct an experiment called propagation of social authentication where propagation starts from all $root$s, we count these authenticated nodes as num_{auth} that could be reached by $root_{min\_num}$ of $root$s within $depth$ steps (result see Fig. 3). The Data set is got from snap [15], a project of Stanford, which originally is sampled from facebook with 3964 nodes and 88159 edges. The diameter is $d = 8$ and average length is $l = 3.68$.



(a) authentication with different number of roots

(b) authentication with different type of roots

Fig. 3. Propagation of social authentication

Figure 3(a) shows the number of authenticated nodes with different minimum number of vouchers (same as $root_{min\_num}$). There are four different value of $root_{min\_num}$ which are 1, 2, 3 and 4. We find that (a) The less $root_{min\_num}$ required, the more nodes could be authenticated. (b) Most of the node will be authenticated when $path_{root}$ reaches around the average path length. For example, here l is 3.68 in G, and when $depth = 4$, more than 90 % nodes are authenticated regardless of $root_{min\_num}$. However, when less of $root_{min\_num}$ required, the authentication start faster.

Figure 3(b) shows the number of authenticated nodes with different type of roots. Three types of root selection strategy are taken: random, maximum degree and community. Communities are detected by fastgreedy [16] as it's faster then walktrap algorithm. The number of community is 13, so we select 13 $root$s in all three strategies. The results are (a) $roots$ of communities is the first to authenticate all of nodes. (b) $roots$ of maximum degree starts faster. In summary, select $root$ by community strategy is the best choice to satisfy requirements of less $depth$ with greater $root_{min\_num}$. We can also infer that networks with small world property (smaller l) are more easily to be authenticated.

4 Social Tracing and Sybil Defending

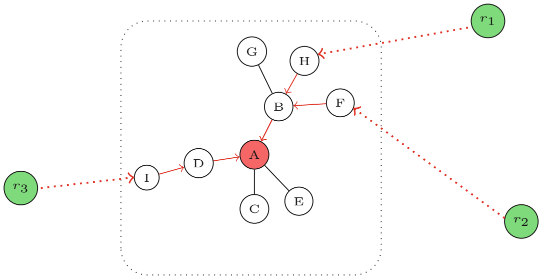


Fig. 4. Social tracing of Sybil protected node

In this section, we discuss mechanism of social tracing and its capability to defend Sybil attack. When a rumor spreads on OSN, if we want to find the rumor-make, we first locate his online identity (this is not our work in this paper). Then a social tracing mechanism could be used to identify his RI. Figure 4 show an abnormal user A who has been identified to be a rumor-maker. And at this moment, we know these information: a friends network G_f, the SAI of A and RIs of all $root$s. When tracing the RI of a user A, we first query $friends(A)$ (here are B, D, C, E). With high probability, we should get RI of A from his friends. However, it's possible that attacker passes authentication with the help of Sybil nodes. In this worst situation, all $friends(A)$ are Sybil nodes, so that they refuse to answer our query and may be even $friends(friends(A))$ are also Sybil nodes that they refuse to answer queries about the RI of $friends(A)$. We are wondering could we find RI of A The answer is yes. Because the *depth* in SAI indicates the maximum iteration time of the query. After *depth* query, we should reach $root$s (Here $depth = 3$, $root$s are r_1, r_2 and r_3) whose RIs are known. The $root$s are responsible to answer queries about their friend, so could trace back to A eventually. Therefore, our social tracing could be used to defend sibyl nodes.

5 Conclusion

In this paper, we introduce a new online identity, SAI, which is capable to trace the real identity of user without real name information. In SAI, some user are select as informers called $root$s and authenticated by RNS. Others are authenticated by their friends, called social authentication. We done an experiment to discussion factors that affect SAI. Our result shows that select $root$ in communities are the best strategy to meet requirement of shorter authentication length and more $root$s as voucher, comparing with strategies like selecting by random and selecting by maximum degree. We also provide social tracing mechanism to trace down rumor-makers. Analysis shows that our social tracing mechanism is robust enough to defend Sybil.

References

1. Statistic Brain. http://www.statisticbrain.com/
2. Danezis, G., Mittal, P.: Sybilinfer: detecting sybil nodes using social networks. In: NDSS (2009)
3. Oh, Y., et al.: Empirical analysis of internet identity misuse: case study of South Korean real name system. In: Proceedings of the 6th ACM Workshop on Digital Identity Management. ACM (2010)
4. Cho, D.: Real name verification law on the internet: a poison or cure for privacy? In: Schneier, B. (ed.) Economics of Information Security and Privacy III, pp. 239–261. Springer, New York (2011)
5. Cho, D., Kim, S., Acquisti, A.: Empirical analysis of online anonymity and user behaviors: the impact of real name policy. In: 2012 45th Hawaii International Conference on System Science (HICSS), pp. 3041–3050. IEEE (2012)
6. Mannan, M.S., van Oorschot, P.C.: Using a personal device to strengthen password authentication from an untrusted computer. In: Dietrich, S., Dhamija, R. (eds.) FC 2007 and USEC 2007. LNCS, vol. 4886, pp. 88–103. Springer, Heidelberg (2007)
7. Sarier, N.D.: A new approach for biometric template storage and remote authentication. In: Tistarelli, M., Nixon, M.S. (eds.) ICB 2009. LNCS, vol. 5558, pp. 909–918. Springer, Heidelberg (2009)
8. McCune, J.M., Perrig, A., Reiter, M.K.: Seeing-is-believing: using camera phones for human-verifiable authentication. In: 2005 IEEE symposium on Security and Privacy, pp. 110–124. IEEE (2005)
9. Zhan, J., Fang, X.: Authentication using multi-level social networks. In: Fred, A., Dietz, J.L.G., Liu, K., Filipe, J. (eds.) IC3K 2009. CCIS, vol. 128, pp. 35–49. Springer, Heidelberg (2011)
10. Brainard, J., Juels, A., Rivest, R., Szydlo, M., Yung, M.: Fourth-factor authentication: somebody you know. In: Conference on Computer and Communications Security: Proceedings of the 13th ACM Conference on Computer and Communications Security, vol. 30, pp. 168–178 (2006)
11. Schechter, S., Egelman, S., Reeder, R.W.: Its not what you know, but who you know. In: Proc. Conf. Human Factors Comput. Syst. (CHI 2009) (2009)
12. Boyd, D.M., Ellison, N.B.: Social network sites: definition, history, and scholarship. J. Comput. Mediated Commun. **13**(1), 210–230 (2007). http://dx.doi.org/10.1111/j.1083-6101.2007.00393.x
13. Xie, B.: The mutual shaping of online and offline social relationships. Inf. Res. **13**(3) (2008)
14. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: Yolum, I., Güngör, T., Gürgen, F., Özturan, C. (eds.) ISCIS 2005. LNCS, vol. 3733, pp. 284–293. Springer, Heidelberg (2005)
15. SANP of stanford (Online). http://snap.stanford.edu/data/
16. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)