An MST–based information flow model for security in Online Social Networks

Nadav Voloch
Department of Computer Science
Ben Gurion University of the Negev
Be'er-Sheva, Israel
voloch@post.bgu.ac.il

Ehud Gudes
Department of Computer Science
Ben Gurion University of the Negev
Be'er-Sheva, Israel
ehud@cs.bgu.ac.il

***Abstract*— Online Social Network (OSN) security issues have been thoroughly researched in the past decade due to their constant increase of users and ongoing feature development. The OSN is usually described as a graph, in which the users are the vertices, and their connecting edges represent relationships such as friendships, interactions and messaging. In our previous research we have devised an Information Flow security model that gives the vertices and edges numerical attributes, based on their user credibility and connection strength, such as age of user account and the friendship duration of two users, from which we have assessed the level of information sharing willingness of an ego node to users that are not directly connected to it, thus categorizing them as adversaries or acquaintances. In this research we use this model to generate a trustworthy network of users by creating a Minimum Spanning Tree (MST) of the graph instance and iterating the removal of the weak-attributed edges and their connected vertices. We use a known MST algorithm (Kruskal's algorithm) on this unique graph, thus achieving a good Trust threshold of a user network that allows a safe information flow in the OSN. Lastly, we validate the model and show its usefulness with experimental results.**

Keywords—Online Social Networks Security, Online Social Networks Privacy, Graph Algorithms for information security, Online Social Networks user credibility

I. INTRODUCTION

Throughout the past few years Online Social Networks (OSN) privacy has been the subject of many researches. Some of the main approaches to this subject are described in [1]. One of the main problems these models deal with is the potential leakage of data to unwanted entities such as adversaries or spammers as shown in [2], where different types of information leakage scenarios are described. Most of these vulnerabilities occur as a result of discretionary privacy policies of OSN users, that create a misleading knowledge of the number and type of users exposed to the shared data. The solution to this problem either requires a change in specific policies or providing a way to control the flow of information through the network. This is needed since usually there is no actual user-awareness to the spreading of personal data throughout the network, as presented in [3]. The control of information flow through the OSN is the topic of this paper.

The flow-control problem was investigated in several recent papers. In [4] the authors suggested cutting all the edges leading to adversaries. The problem is that it also cuts edges to friends from the flow. In [5] this problem was solved by devising graph algorithms that preserve maximal flow to friends whilst keeping a minimal flow to adversaries. Their research established a sharing- habits privacy control model, where a community directed and weighted graph with an Ego-node is defined. The edges are data instances flowing from the Ego – node to the other users, and the edges' weights are sharing probabilities. The problem of data leakage was dealt with by controlling the information flow with Min-cut graph algorithms that prevent potential data leakage. A variation to this model was presented in [6], where we have used different OSN attributes to assess a user node that is not directly connected to the Ego-node. We identify the user as a potential adversary or acquaintance by the evaluation of different user and connection attributes, such as total number of friends, friendship duration, age of user account, etc. Then a trust value is computed for each of the nodes and edges, and based on a trust threshold, edges are cut, and information flow is blocked to undesired entities.

In this paper we also suggest cutting edges, but with a knowledgeable decision of whom to cut, by parameters of credibility. OSN users usually know that the information they share is exposed to their friends, but what they do not necessarily know is that if their friends act (share, like, etc.) on this data, it is exposed to a different user network, that could be unsafe or unwanted. These networks are the subject of our research, that handles this information leakage issue. We address the problem of data leakage to unknown users by using the information flow model to create a trustworthy network of users, to whom the data being spread from the Ego node is monitored. This is done using a known graph algorithm for finding a Minimum Spanning Tree (MST), that is the Kruskal's algorithm, as will be described in the following parts of the paper. The main contribution is therefore an algorithm which constructs for each ego-user a trust network which will assure that information flows only to safe nodes in this network.

The rest of this paper is structured as follows: Section II discusses the background for our work, with explanations on the related papers it relies on. Section III describes our

978-1-7281-1340-1/19/$31.00 ©2019 IEEE

previous trust model and its parameters. This trust model is the basis for our current information flow model, presented here. The section defines our model thoroughly with several examples of its operation. Section IV discusses the model's evaluation and validation using our experimental results, and Section V is the conclusion of the paper, with future prospect on further research on this subject.

II. Background and related work

Using Information Flow control for preserving privacy in OSN was investigated by several early papers on OSN. [7] presents a privacy architecture that reduces potential information leakage threats whilst preserving good accessibility to the user's important data. Another early paper that handles these issues is [8], that presents a heuristic method for network security based on user identification and shows a novel method of basing the credibility of a certain user on its relationship with other users. Changing a graph for anonymization purposes is discussed in papers such as [9] and [10], where the anonymization of the OSN is done by sequential clustering.

In [11] the user-relationship approach for information flow security in OSN is further developed. The OSN is portrayed in a modular manner, in which a deeper resolution of the graph is given, as the vertices represent different data instances whilst the edges are connections between them, such as friendships between users, sharing of posts or pictures, etc. The model is dynamic and fits a real-life applicative form as it shows the different graph instances in several timestamps, monitoring the changes over time. This model specifically uses the Facebook jargon of OSN activities such as Wall posts, sharing, tagging, etc. This choice is well justified having Facebook a multi-functional OSN, serving as a professional and social OSN, as well as serving other purposes and including numerous additional features (the Facebook model is also used in our paper). [12] presents a model named IMPROVE-Identifying Minimal Profile Vectors for similarity-based OSN privacy control. It elaborates on the importance of user and connection attributes for setting a credibility level of an OSN data instance by giving ranking to these attributes. This ranking is based on information gaining from each attribute, assessing their importance in the closeness approximation between users and evaluating their information sharing willingness.

The above approaches and researches contributed to our current research. Our goal is to create a trustworthy solution for OSN users in the aspect of their data privacy and unwanted information leakage to unknown users that might take advantage of the user's private data instances. In the following parts of this paper, we base our model on attributes which were investigated in the mentioned above approaches but use them to construct a comprehensive information flow control model that creates a trusted user network, by using a Minimum Spanning Tree algorithm for an informed choice reduction of relatively non-credible nodes and edges.

III. The secure network information flow model

A. Preliminary motivation

We present the social network as an undirected graph, where nodes are the OSN users, and edges represent relations between them such as friendship relations. We designate the user for which we like to control its information flow as the Ego-user. The issue we address is whether the ego-user is willing to share information with another user. This specific point is quite obscure in terms of OSN users. The definite knowledge users have is that their information instances (such as pictures, posts, personal details, etc.) are revealed to their direct OSN friends. In the OSN graph terms, these are the Ego-node's adjacent vertices. The information leakage problem begins when one of these friends acts upon this information instance, meaning comments on a post, likes or shares a picture, or any other form of OSN action. As this occurs, his friends, which are mostly not direct friends of the Ego-node, can see this information instance. In the OSN graph, any other vertex that is in a distance > 1 is of an unknown definition at the time to the Ego-node, and its exposure to the Ego-node's information could be harmful or just unwanted for the Ego-node user. We can see this situation in Fig. 1 that describes information leakage of the Ego-user's data. The goal of our model is to create a new Ego-user's graph, which best represents, its trusted network and prevents leakage to undesired users. This process is composed of two phases:

- In the first phase, we assign trust values to both nodes and edges, based on attributes that were examined in our previous information flow control model. This phase will be described briefly in the following section.
- The second phase, which is also described in the next section, removes edges from the graph by constructing a Minimum Spanning Tree.

The result of this two-phase process is a new trusted network for the ego-user with minimal undesired information flow.

Although social networks may contain millions of nodes and tens of millions of edges, the Ego-node's trustworthy network is usually at a distance of two or three, which limits the number of nodes and edges involved, therefore makes the running of the MST algorithm feasible.

B. Grenrating the graph's trust values

For the main problem handled by the model, we need to assess the users' level of credibility in terms of OSN parameters that include both personal attributes and connection attributes in relation to their relevant adjacent users (direct friends). These attributes were handled in our previous information flow model mentioned above. In that research, more attributes were used but some of their threshold values were difficult to evaluate. In this paper we use only a small part of those attributes but validate their threshold values using a user study we conducted on a survey of 282 OSN users that were asked for the importance of various attributes in their decisions to grant various permissions to their private data. The survey included the quantifiable attributes of user credibility and its connection strength to their adjacent users. The survey and its results are further explained in the following parts of this paper. Setting the values for the Trust variables is done in this model in a probability scale of 0 to 1, since the decision of sharing information with a certain user is defined as a probability variable, 0 being no sharing willingness at all, 1 being definite sharing willingness.

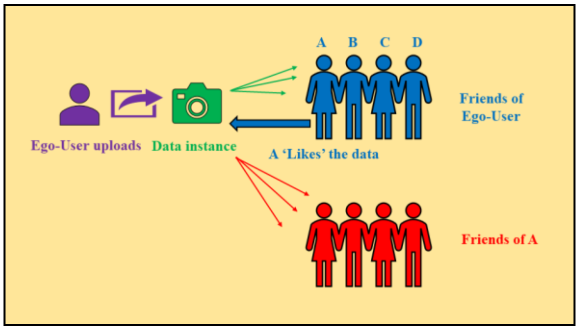

Fig. 1: Data leakage of a data instance because of a friend's activity

The attributes used here are of two types. The notation of *u* represents the user's personal attribute and the notation of *c* represents the user's connection attributes in relation to his relevant adjacent users.

These attributes and their values are as follows:

u_{TF} value is based on the Total Friends (*TF*) attribute, and the average value shown in [13], having fake profiles, social-bots, etc., with an allotted number of friends. The experimental result for this attribute was the lower bound of 244.34. A profile of 245 friends and above is with a high probability of being a genuine user profile.

$$u_{TF} = \begin{cases} \frac{TF}{245} & (TF < 245), \\ 1 & (TF \geq 245). \end{cases} \tag{1}$$

u_{AUA} value is calculated in months. It is based on the estimation of the Age of User Account (*AUA*) attribute of [14], that an active spammer profile will not remain active for a long term, due to OSN security updating policies. The experimental result for this attribute was the lower bound of 23.82. A profile with a seniority of 24 month and above is with a high probability of being a genuine user profile.

$$u_{AUA} = \begin{cases} \frac{AUA}{24} & (AUA < 24), \\ 1 & (AUA \geq 24). \end{cases} \tag{2}$$

c_{MF} value is taken from the Mutual Friends (*MF*) attribute having fake profiles, social-bots, or even adversaries, with a small number of mutual friends, if any. The experimental result for this attribute was the lower bound of 37. A profile of 37 mutual friends and above is with a high probability of being a close friend.

$$c_{MF} = \begin{cases} \frac{MF}{37} & (MF < 37), \\ 1 & (MF \geq 37). \end{cases} \tag{3}$$

c_{FD} value is calculated in months. It is based on the Friendship Duration (*FD*) attribute, having a relatively unknown user, or even a fake profile or spammer, being friends with the Ego user not for a substantial amount of time, is of an unwanted sharing willingness potential. The experimental result for this attribute was the lower bound of 17.12. A friendship of 18 months and above is most likely to be a genuine connection.

$$c_{FD} = \begin{cases} \frac{FD}{18} & (FD < 18), \\ 1 & (FD \geq 18). \end{cases} \tag{4}$$

Generating our trusted network is done in two main steps which are the preliminary creation of the graph with the trust evaluation of users and their connections, and the use of the Minimum Spanning Tree algorithm on the graph based on these values. For the first part of creating the graph, the construction's definition is as follows:

Let $G = (V, E)$ be an undirected graph that describes OSN activities, where *V* is the set of users and *E* is their connected social activities. For $1 \leq i \leq n$ and $v \in V$, the attribute u_{vi} is defined as the average of the two attributes: $\langle uTFi, uAUAi \rangle$, meaning that a vertex's u_{vi} is the user credibility attribute, that is the average of the parameters mentioned above.

For $1 \leq i \leq n$ and $e \in E$, the attribute c_{ei} is defined as the average of the two attributes: $\langle cMFi, cFDi \rangle$, meaning that an edge's c_{ei} is the connection strength attribute, that is the average of the parameters mentioned above.

In Fig. 2 we can see such an OSN graph, in its preliminary form with its raw attribute values (for MF, TF, AUA and FD), before calculations. For these calculations we can take for example the upper left part of the Ego-user's network in Fig. 2, which is the user Alice's network. The values of the Trust attributes' calculation are shown in Table I and are the base for the manifestation of the MST algorithm shown in the next part of this section and seen in Fig. 3. In general, we use a simple average for the attributes' calculation, though a weighted one can also be applied, if needed by different OSN decisions. Here we need to state and clarify that the parameter values are ones of experimental evaluation and validation, and not universal constants, that are axioms of any sort. Having the network very dynamic through time, these numbers can of course change.

C. Using Kruskal's MST algorithm for securing the graph

Before describing the algorithm, we define the concept of MTV – minimum trust value which will helps us in deciding which nodes and edges we remove from the graph.

Finding the MST of a graph is a well-known problem dealt with in different aspects and applications, and by many efficient algorithms as presented comprehensively in [15]. One of the better-known algorithms is Kruskal's algorithm that's was first shown in [16] and its description is as follows:

- Given an undirected connected weighted graph G create a sorted set S containing all the removed edges in the graph, ordered by weights.
- While S is non-empty, and G is not yet spanning:
  - Remove an edge E with minimum weight from S.

At the termination of the algorithm, it forms an MST of the graph. The meaning of this algorithm to our OSN graph is that it finds the weakest (thus, insecure) connections of the graph, and following this notion we have devised the following algorithm:

The Trustworthy network algorithm:

- If $u(v_j) \geq MTV$ then e_j, remains.
- If the removed edge E does not create a circle, put it in G.

- Let $G = (V, E)$ be an OSN graph as described in the previous subsection (3.3.1).
- Define a Minimal Trust Value (MTV) for which a deletion candidate vertex will be measured.
- For $1 \leq i \leq n$, representing every friend from the OSN Ego-user's friend list:
  - Let $G_i = (V_i, E_i) \subseteq G$ an undirected weighted connected graph that represents the friend's network.
  - Let $G_i' = (V_i', E'_i) \subseteq G_i$ be the MST result of $Kruskal\ (G_i)$.
  - For $e_j, \in E'_i$ where $1 \leq j \leq m$:
    - If removing e_j, does not disconnect v_j, v_{j+1} (its connecting vertices): $G_i = G_i - e_j$, (the edge is removed).
    - Else check $u(v_j)$:
    - If $u(v_j) < MTV$ then $G_i = G_i - e_j$, (the edge is removed).

A simple explanation for the algorithm is that for every Ego-user's friend graph (as seen in Fig. 2 with 4 friends), we use the Kruskal algorithm to find the MST- this is the weak (insecure) part of the network.

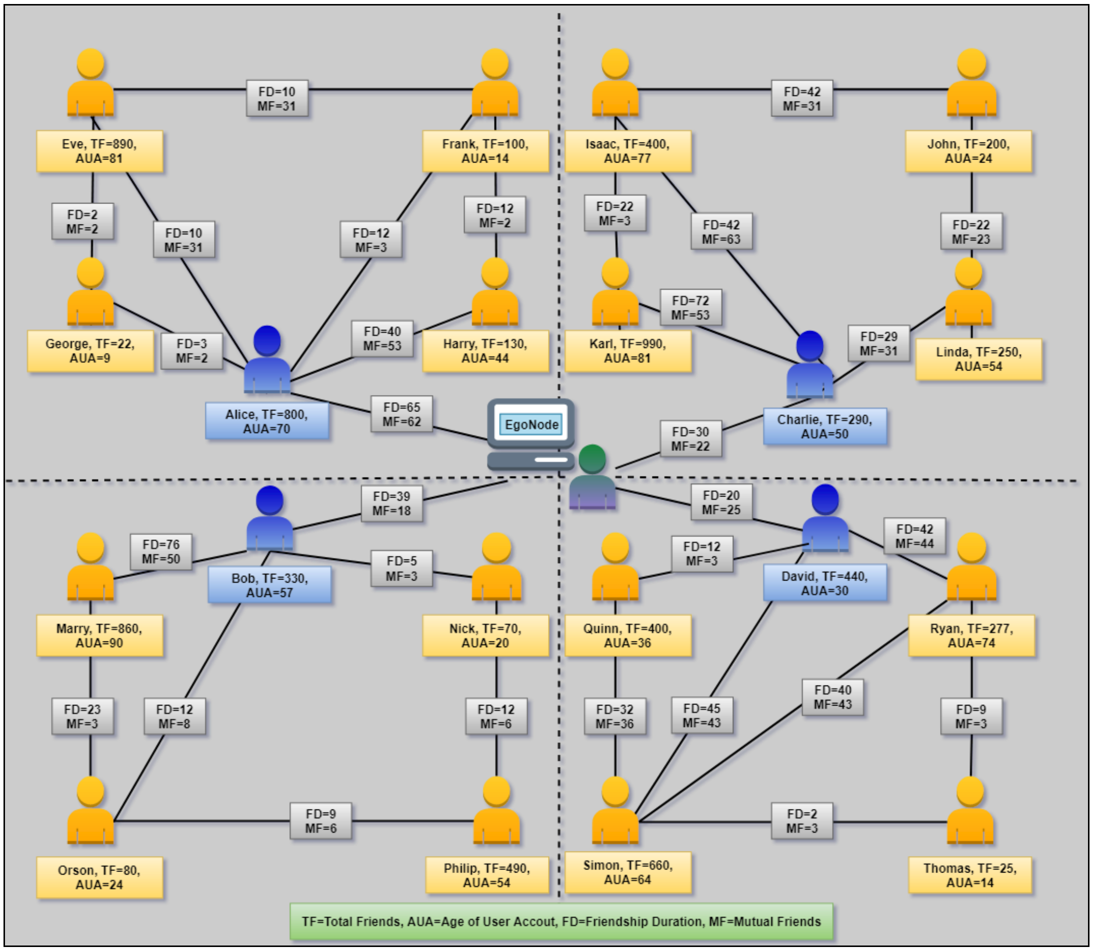

Fig. 2: OSN user graph with the model's trust attributes for the users and their connections

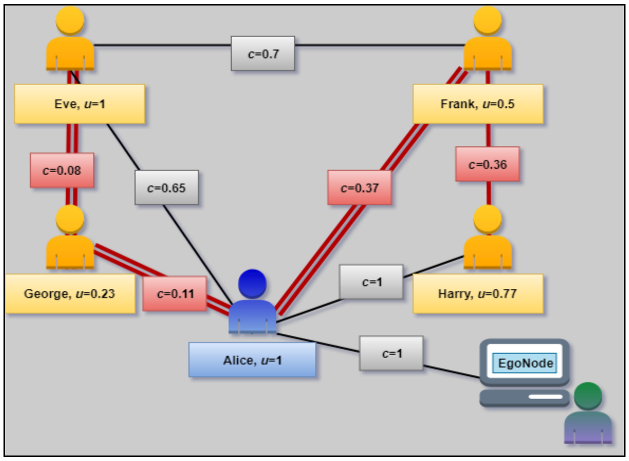

Fig. 3: Creating the MST in the Ego-user's sub-graph by the evaluation of trust attributes

After finding the MST we remove its edges one by one, and if a certain user (vertex) is going to be disconnected from the graph, we first check its trust value (u), and if it is equal or higher than our preliminary defined threshold (*MTV*), we do not remove it, if it's lower, we do remove it.

After the action of this algorithm, that is the disconnection of the weak edges and vertices from the OSN graph, we get an Information-flow Trust Network, where the information is freely shared.

We can assess the overall network's trust value by averaging all the edges' and nodes' trust values, and if we desire having a higher overall value – the algorithm action can be executed iteratively, until a desired threshold is reached.

TABLE I: TRUST ATTRIBUTES' CALCULATION FOR THE USER ALICE'S NETWORK.

User/Connection	Attribute calculation
Eve	$u = \langle uTF, uAUA \rangle = \langle 1,1 \rangle = 1$
George	$u = \langle uTF, uAUA \rangle = \langle \frac{22}{245}, \frac{9}{24} \rangle = 0.23$
Frank	$u = \langle uTF, uAUA \rangle = \langle \frac{100}{245}, \frac{14}{24} \rangle = 0.5$
Harry	$u = \langle uTF, uAUA \rangle = \langle \frac{130}{245}, 1 \rangle = 0.77$
Alice	$u = \langle uTF, uAUA \rangle = \langle 1,1 \rangle = 1$
Eve - George	$c = \langle cMF, cFD \rangle = \langle \frac{2}{37}, \frac{2}{18} \rangle = 0.08$
Eve - Frank	$c = \langle cMF, cFD \rangle = \langle \frac{31}{37}, \frac{10}{18} \rangle = 0.7$
Eve - Alice	$c = \langle cMF, cFD \rangle = \langle \frac{11}{37}, 1 \rangle = 0.65$
George - Alice	$c = \langle cMF, cFD \rangle = \langle \frac{2}{37}, \frac{3}{18} \rangle = 0.11$
Frank - Alice	$c = \langle cMF, cFD \rangle = \langle \frac{3}{37}, \frac{12}{18} \rangle = 0.37$
Frank - Harry	$c = \langle cMF, cFD \rangle = \langle \frac{2}{37}, \frac{12}{18} \rangle = 0.36$
Harry - Alice	$c = \langle cMF, cFD \rangle = \langle 1,1 \rangle = 1$
Ego - Alice	$c = \langle cMF, cFD \rangle = \langle 1,1 \rangle = 1$

An advantage of this algorithm is that highly connected vertices, that are not necessarily with a high MF (Mutual Friends) attribute will remain in this Trust Network. An example of the algorithm's manifestation in the upper left part of the Ego-user's graph of Fig. 2 (Alice's network) is seen in Fig. 3 – the MST before the removal, having the user George preliminary disconnected and then later removed since $u(v_j) < MTV$.

Note also that Frank had two edges disconnected but can still issue a sharing request if Eve acts on the data instance.

If we run the MST algorithm again, all of the remaining edges in Fig.3 are candidates for removal, but only Frank will be disconnected because of its low trust value. One can run the algorithm any number of iterations, depending on the threshold value needed.

D. *Real OSN implementation of the model*

In a real OSN, the graph's edges are not permanently cut, because the algorithm we have described here is done once and then the result is saved for each Ego-user. At runtime, every sharing request is checked against the saved trusted network, thus deciding which node will be exposed to this data.

The actual implementation of the model is done in a finite distance of users, that can be set by the network administrator. This distance makes the network safe in a given perimeter, given the assumption that an exposure to the Ego-user's data of an unknown user in a great distance (meaning no direct connection to the Ego-user's close environment) is generally unwanted.

IV. THE MODEL'S VALIDATION BY EXPERIMENTAL RESULTS

For the model's validation we have conducted an experimental survey that included 1968 users' evaluations done by 123 participants. Every participant evaluated the 16 users of the graph presented in Fig. 2, thus we get trust values for 1968 users. This gives us the option to compute the values and importance of each of the parameters defined above (clearly this experiment can be conducted for more users and other graphs for different networks and may result with different parameters values).

The purpose of this evaluation was to estimate the correlation between the model's cutting decisions and the general trust estimation and, more importantly, sharing willingness, of real OSN users. The average SP (Sharing Probability) results are as follows (with respect to the Ego user):

{{**John:0.414,** Isaac:0.483, Linda:0.541, Karl:0.528},
{Frank:0.428, Eve:0.497, Harry:0.559, **George: 0.386**},
{Ryan:0.534, Quinn:0.51, **Thomas:0.366**, Simon:0.466},
{Nick:0.428, Marry:0.503, Philip:0.462, Orson:0.434}}.

The model's algorithm results on the same graph (of Fig. 2) are presented in Table II, where the MST's of the friends' networks are shown, as well as the removal candidates and their trust values.

The juxtaposition of the two gives us the same lower bound candidates for each network, meaning the users that are cut from the graph in the model, because of the MST result and low trust values, were the ones that also got the lowest SP in every one of the 4 networks.

The algorithm result of Charlie's network (John) seems a bit high (0.91), but in relation to the other users of the network, it is the lowest since it is a very strong network.

A certain weakness of this and any Trust related model is that new OSN users "fall through the cracks" since their Trust parameter values, such as *AUA* and *TF*, are very low, even though they could be legitimate users that will be mistaken for fake profiles or spammers.

For these specific cases of new users, we can remedy the problem by giving extra weight to other attributes, such as *MF*, and an exception can also be made by using the ration of *MF/TF*, which will hold for cases of new genuine users, that have multiple connections to the Ego-node's network.

V. Conclusions

In this paper we have presented a model for creating a trustworthy network of users, that gives a good privacy infrastructure for such a solution. The need for such a solution is the growing amount of personal information in OSN, and with it, grows the need for its privacy.

We have used a known graph algorithm (Kruskal for finding the MST), along with the combination of several user and connection trust attributes, to find the weak security leaks in such an OSN graph, and to create a stronger, more viable trusted sub-graph, in which users can be relatively safe in terms of information sharing.

These MST edges are the weak links of the network. Removing them substantially improves the trustworthiness of the network, potentially removing hazardous adversaries. The model handles the problem of distributing the Ego-user data to relatively unknown users, that are connected to his friends. This implementation reduces the data leakage by removing potentially harmful users from his extended network. As explained, the number of disconnected edges can be increased by running the MST algorithm multiple times, depending on the needed trust value threshold.

In [17] we described an Access Control model that can deny a sharing request based on the node's trust values, and not on their edge connections.

In future work on this model, we like to extend its use with the combination of our previous access control and flow control models to create a more comprehensive and large-scale model, with a solid software implementation of the intertwined models. More experimental results for this part and other parts of the combined model are also a work in progress.

Table II: The model's results for removal candidates and their trust value, divided by friends' networks

Friend	MST of the friend's network	Removal candidates and their trust value
Alice	{Alice-George, George-Eve, Alice-Simon, Frank-Harry}	**George:** u=0.23
Charlie	{Charlie-Linda, Linda-John, John-Isaac, Isaac-Karl}	**John:** u=0.91
Bob	{Bob-Nick, Nick-Philip, Philip-Orson, Orson-Marry}	**Nick:** u=0.56
David	{David-Quinn, Quinn-Simon, Simon-Thomas, Thomas-Ryan}	**Thomas:** u=0.34

References

[1] Kayes, I., and Iamnitchi, A. "Privacy and security in online social networks: A survey". Online Social Networks and Media, 3, 1-21, 2017.

[2] Li, Y., Li, Y., Yan, Q., and Deng, R. H.. "Privacy leakage analysis in online social networks". Computers and Security, 49, 239-254, 2015.

[3] Misra, G., and Such, J. M.. "How socially aware are social media privacy controls?" Computer, 49(3), 96-99, 2016.

[4] Ranjbar, A., and Maheswaran, M.. "Using community structure to control information sharing in online social networks". Computer Communications, 41, 11-21, 2014.

[5] Levy, S., Gudes, E., and Gal-Oz, N.. "Sharing-habits based privacy control in social networks". In IFIP Annual Conference on Data and Applications Security and Privacy, pp. 217-232. Springer, 2016.

[6] Gudes, E., Voloch, N.."An Information-Flow Control Model for Online Social Networks Based on User-Attribute Credibility and Connection-Strength Factors".International Symposium on Cyber Security Cryptography & Machine Learning, pp. 55-67. Springer, 2018.

[7] Lucas, M. M., and Borisov, N.. "Flybynight: mitigating the privacy risks of social networking". In Proceedings of the 7th ACM workshop on Privacy in the electronic society (pp. 1-8). ACM, 2008.

[8] Gross, R., and Acquisti, A.. "Information revelation and privacy in online social networks". In Proceedings of the 2005 ACM workshop on Privacy in the electronic society (pp. 71-80). ACM, 2005.

[9] Das, S., Eğecioğlu, Ö., and El Abbadi, A.. "Anonymizing weighted social network graphs". In 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010) (pp. 904-907). IEEE, 2010.

[10] Tassa, T., and Cohen, D. J.. "Anonymization of Centralized and Distributed Social Networks by Sequential Clustering." IEEE Trans. Knowl. Data Eng., 25(2), 311-324, 2013.

[11] Patil, V. T., and Shyamasundar, R. K. "Undoing of Privacy Policies on Facebook". In IFIP Annual Conference on Data and Applications Security and Privacy (pp. 239-255). Springer, Cham., 2017.

[12] Misra, G., Such, J. M., and Balogun, H.. "IMPROVE-Identifying Minimal PROfile VEctors for similarity-based access control". In Trustcom/BigDataSE/I SPA, pp. 868-875). IEEE, 2016.

[13] Dunbar, R. I.."Do online social media cut through the constraints that limit the size of offline social networks?" Royal Society Open Science 3.1 :150292, 2016.

[14] Zheng, X., Zeng, Z., Chen, Z., Yu, Y., and Rong, C.. "Detecting spammers on social networks." Neurocomputing, 159, 27-34, 2015.

[15] Gabow, H. N.; Galil, Z.; Spencer, T.; Tarjan, R. E.. "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs". Combinatorica. 6 (2): 109, 1986.

[16] Kruskal, J. B.. "On the shortest spanning subtree of a graph and the traveling salesman problem". Proceedings of the American Mathematical Society. 7: 48–50, 1956.

[17] Voloch, N., Levy, P., Elmakies, M., Gudes, E. " An Access Control model for data security in Online Social Networks based on role and user credibility." International Symposium on Cyber Security Cryptography & Machine Learning. Springer, 2019.