Towards Active Detection of Identity Clone Attacks on Online Social Networks

Lei Jin
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
lej17@pitt.edu

Hassan Takabi
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
hatakabi@sis.pitt.edu

James B.D. Joshi
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
jjoshi@sis.pitt.edu

ABSTRACT
Online social networks (OSNs) are becoming increasingly popular and Identity Clone Attacks (ICAs) that aim at creating fake identities for malicious purposes on OSNs are becoming a significantly growing concern. Such attacks severely affect the trust relationships a victim has built with other users if no active protection is applied. In this paper, we first analyze and characterize the behaviors of ICAs. Then we propose a detection framework that is focused on discovering suspicious identities and then validating them. Towards detecting suspicious identities, we propose two approaches based on *attribute similarity* and *similarity of friend networks*. The first approach addresses a simpler scenario where mutual friends in friend networks are considered; and the second one captures the scenario where similar friend identities are involved. We also present experimental results to demonstrate flexibility and effectiveness of the proposed approaches. Finally, we discuss some feasible solutions to validate suspicious identities.

Categories and Subject Descriptors
H.2.7 [**Information Systems**]: Database Administration-*Security, integrity, and protection*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection-*Authentication*

General Terms
Security, Algorithm, Experimentation, Measurement

Keywords
Identity Clone Attack, Detection, Security, Online Social Networks, Profile Similarity

1. INTRODUCTION
The popularity of online social networks (OSNs) is growing significantly. At the same time, they are also being increasingly targeted by adversaries interested in extracting privacy sensitive information about users. The security and privacy threats on OSNs are mainly raised by the increasing amount of publicly available personal information posted by users in their profiles.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CODASPY'11, February 21–23, 2011, San Antonio, Texas, USA.
Copyright 2011 ACM 978-1-4503-0465-8/11/02...$10.00.

Recently, a new attack called Identity Cloning Attack (ICA), which focuses on forging user profiles on OSNs, has been introduced [1]. The key goal of an adversary in ICA is to obtain personal information about a victim's friends after successfully forging the victim, and to establish increased levels of trust with the victim's social circle for future deceptions. In this attack, the adversary first tries to find ways to obtain a victim's personal information, such as name, location, occupation and friends list from his public profile on OSNs or his personal homepage(s). Then, the adversary forges the victim's identity and creates a similar or even identical profile on OSN sites. Afterwards, he sends friend requests to the victim's contacts. Once the friend requests are accepted, he builds the victim's friend network and gains access to profiles of the victim's friends. In addition, he can launch ICAs on the victim's friends based on these personal data. The adversary can also implement an automated, cross-site profile cloning attack [1] in which the adversary can clone the identity of a victim in one site where the victim is registered, and forge it on another OSN site where the victim is not registered yet. After having successfully created the forged identity on the site where the victim has no account, the adversary can automatically attempt to rebuild the social networks of the victim by contacting his friends who have accounts on both sites. The experimental results show that both of these ICA schemes are effective and adversaries do not raise much suspicion in users who have been compromised [1].

The ICA on OSN sites is not only a privacy attack for the victims but also may cause potential financial loss and severely affect the trust they have built on OSNs [1]. Most users may be apt to trust their friends' activities on OSN sites more than their activities on other websites. This is because OSNs are built on the core concept of making friends and sharing information with each other. However, due to users' lack of awareness, they are more likely to trust both OSN sites and their friends there. Such trust makes it easier for adversaries to obtain a victim's personal information and then clone the identity. Thus, it is urgent to build mechanisms to detect ICAs and try to reduce the loss on OSN sites.

In regard to defending against ICAs, most solutions focus on educating users to control the distribution of their sensitive personal information and digital identities [6, 7]. *FightIDTheft* [6] and *Facebook Identity Theft* [7] focus on providing detailed suggestions to help users to define their privacy policies. Typically, well-designed OSN sites allow users to customize their privacy policies. For example, *Facebook* has a "Privacy Settings" page that allows users to specify which pieces of profile data each user is allowed to view. It also allows users to set fine-grained access policies by specifying whether a piece of profile data is

visible or not to some friends. However, configuring fine-grained privacy settings on OSNs are often complicated and time-consuming task that many users feel confused about and usually skip. Unfortunately, most of users are apt to stand in side of popularity and usability rather than security and privacy.

There are several third-party applications of OSN sites proposed and employed for protecting users against ICAs. For instance, in *Facebook*, *Identity Badge* identifies a user via a passport check [4], and *mysafeFriend* validates a user's identity by asking user's friends to verify him and doing a credit card check [5]. Although these applications may help users to validate who they are and protect their identities, they are passive protections and only used to identify users themselves but cannot defend against ICAs targeting them. The faked identities still exist on OSN sites and adversaries continue to deceive more victims using them without any restrictions. Therefore, an active security mechanism to detect faked identities on OSNs is required and urgent.

However, detection of faked identities is a challenging work. The key challenges are as follows:

- It is quite common that several people have similar names in real world; and hence their identities on OSNs may be similar. We cannot arbitrarily infer all the similar identities that have similar names as faked identities.
- The characteristics of a faked identity have not been analyzed and summarized.

In this paper, we address these challenges and propose an active detection framework to detect existing faked identities on OSNs. We first investigate and characterize a faked identity. We observe that a cleverly crafted faked identity may not only forge attributes of the victim but may also add friends into its networks that are also in the victim's networks. We propose two approaches to calculate profile similarity between two identities based on *attribute similarity* and *friend network similarity*. Based on profile similarity, we propose a framework to detect faked identities on OSNs that includes three steps. The first step is to search and filter identities in profile set where the input is a profile. The second step is to discover a list of suspicious identities related to the input profile using profile similarity schemes, and the last one is to verify the identities in suspicious identity list and remove the faked ones. In our detection process, we use a flexible set of parameters, which can be adjusted to distinguish a victim from its clones and may achieve accurate detections on different OSNs where the faked identities may have different behaviors.

The contribution of this paper is three fold. First, to the best of our knowledge, our work is the first attempt to characterize the faked identities, and detect them on OSN sites using an active approach based on profile similarity. Second, we propose two profile similarity schemes to discover suspicious identities. Third, we present experiments to demonstrate that our detection schemes are flexible and effective.

The rest of the paper is organized as follows. In Section 2, we characterize attribute and friend networks of a faked identity. In Section 3, we introduce the detection framework with two profile similarity schemes: *Basic Profile Similarity* and *Multiple-Faked Identities Profile Similarity*. Then, we build the dataset, discuss the relationships between parameters and detection results and evaluate our proposed models in Section 4. In Section 5, we discuss the related work and discuss limitations in Section 6. Finally, Section 7 concludes the paper and discusses future work.

2. CHARACTERISTICS OF FAKED IDENTITIES

In this section, we discuss the limitation of friend adding process in existing OSNs and introduce characteristics of a faked identity based on its attribute and friend network characteristics.

2.1 Friend Adding Process on OSNs

In typical existing OSN sites, when user B receives a friend request from user A, or user B gets user A's link as a result of a friend search function or friend recommender feature of OSN, there is no immediate approach for B to verify the authenticity of A. Furthermore, user B can see only the public features of user A. An adversary can create a faked identity by copying the victim's public attributes and adding the individuals in the victim's friend networks into its networks. In addition, an adversary can also first become a friend of the victim to gain access more features that are visible to his friends, which may be used to forge the victim more successfully. Due to the similarity of the public features between the victim and the faked identity, it is difficult for users to distinguish them when they receive friend requests.

limitation

2.2 Definition of a User Profile

A social network is a social structure modeled as a graph, where nodes represent users or other entities (e.g. group) embedded in a social context, and edges represent specific types of relationships between entities. Such relationship may be based on real-world friendship, common values, shared visions and ideas, kinship, shared likes and dislike, etc. [3].

In this paper, we only model users' attributes and their friend networks since the main target of ICAs is to forge the victims' attributes and rebuild the victims' friend networks to establish trust from their friends [1]. We assume that each identity may have three different lists associated with it; a *friend list* that indicates other users in his friend network, a *recommended friend list* that OSN sites generate to recommend potential new friends to users based on activities or common interests, and an *excluded friend list* that indicates people who users avoid from having in their friend network such as parents, boss, etc. We define an identity profile (or simply a profile) in an OSN site as follows:

DEFINITION 1. : *Let x be a user in an OSN system, we define public profile of x, represented as* P_x*, as a tuple* (A_x, G_x)*, where*

a) $A_x = \{(a, v) \mid a$ *is an attribute name and* v *is its value*$\}$ *represents x's public attribute-value set;*

b) $G_x = (FL_x, RFL_x, EFL_x)$ *represents x's public friend networks, where* FL_x*,* RFL_x*, and* EFL_x *are x's friend list, recommended friend list and excluded friend list respectively.*

We assume that friendship relationships in the *friend list* are bidirectional while relationships in the *recommended friend list* and *extended friend list* may not be bidirectional.

2.3 Attribute as Target

In most OSN sites, *Name* of a user, *e.g.* a full name "*Martin Luther King*", is usually public to everyone and is treated as a key piece of identification information. Consequently, as the first step of ICA, an adversary usually creates a faked identity that has a name that is same or similar to that of the victim, such as "*Martin L. King*" and "*Martin King*". Besides *Name*, we note that a faked identity may have several attributes whose values are similar to those of the victim, if not the same. In this paper, "similar" means "equivalent" when values of these attribute are numeric, such as

Birthday and *Age* while it means "equivalent", "synonymous" or "abbreviation" when the values are strings. Some of attributes of a victim may be easily obtained from his homepage or his public profile on OSNs. Sometimes, an adversary may not obtain such information easily when a user does not have a homepage or is careful enough to publicly disclose as little personal information as possible. However, even if a victim's public information is little, it cannot prevent an adversary from implementing ICAs. He can arbitrarily set the values of fake identity's attributes when he does not know the victim's corresponding attribute value, and then hide them as private attributes. As users not connected to the faked identity only see the public attributes, they may not be able to distinguish the faked identity from the genuine identity.

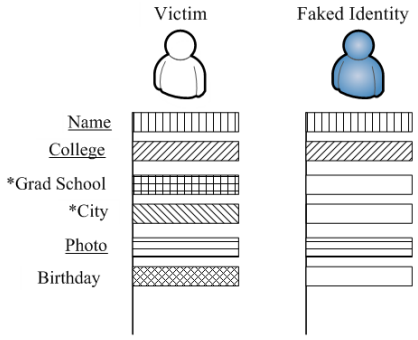

Figure 1. Attribute characteristics of a faked identity

An experienced adversary may also be able to easily guess the victims' privacy settings by checking the blank values in the victim's public profiles. By doing this, the adversary can forge not only the values of the attributes they do not know, but also hide some information that is regarded as private in general but is set as public by the victims. This activity may make the faked identities more genuine. In Figure 1, the left part represents the victim's profile and the right part is the faked identity created by an adversary. In the victim's profile, the *Grad School* and *City* are set to be private (indicated by an *) and other attributes are public. The adversary, if he is not a friend of the victim, can obtain the values of the victim's *Name*, *College*, *Photo* and *Birthday*, but he may not know the values of *Grad School* and *City*. However, he can create a faked identity which has the same *Name*, *College* and *Photo* as the victim. For *Grad School* and *City*, he may simply configure the privacy settings similar to that of the victim, sets random values for these attributes and hides them. For *Birthday* which is usually sensitive and is set to be private by most of users, the adversary can set it as private, while the victim in Figure 1 has it set as public. Such an activity may make the victim's friends believe the faked one more than the real one because it complies with the general trends regarding privacy settings.

Based on the above analysis, we summarize the key targets while creating attributes of a faked identity as follows:

- ***Attribute value* as target**: An adversary creates a faked identity that has several similar attributes to those of the victim.
- ***Privacy setting* as target**: An adversary creates a faked identity that has several similar attributes to those of the victim but also follows his privacy settings.

2.4 Friend Networks as Target

The main goal of ICA is to add victim's friends and build the trust relationship with them [1]. Therefore, after an adversary creates a faked identity based on attributes similar to that of the victim, his next step is typically to forge friend networks of the victim. Earlier we introduced the friend networks in definition 1. We elaborate on these lists below.

Friend List (FL)

At first, an adversary may try to obtain the *friend list* of a victim's profile. Then, he may send friend requests to all of these friends in order to forge the victim more accurately. Experimental results reported in [1] suggest that a typical user tends to accept a friend request from a fake identity although its real counterpart is already in his *friend list*. Thus, the adversary may add many friends of the victim successfully in his networks. After he gets enough friend requests accepted, he can forge the victim successfully.

Recommended Friend List (RFL)

A cautious user may check his friend list before he makes a decision to accept a suspicious friend request. However, an experienced adversary can still forge the victim successfully even when he cannot add enough existing friends of the victim. Many OSN sites generate a list of recommended friends for every identity based on some of his attributes, such as Educations and Interests. These recommended friends are probably friends of the victim or the ones who are familiar with him, but they are not in the *friend list* of the victim. The victim may add these people as friends in future. However, the recommended friends generated for a faked identity may be same as those of the victim, since the adversary has successfully forged related attributes of the victim. Then, an adversary can send friend requests to these recommended friends before the victim does. After the friend requests are accepted, it may make the faked identity more genuine than the victim and makes it difficult for the victim to add these friends. Therefore, besides the existing friends of the victim, the recommended friends of the victim should be considered in characterizing the friend networks of a faked identity.

Excluded Friend List (EFL)

In order to avoid social embarrassments within OSNs, some users may be less likely to add some individuals that they are very familiar with into their friend list. For instance, teenagers may want to make sure they do not add their parents, elder relatives and tutors to their *friend list*. These people may or may not be listed in his *recommended friend list*. In this paper, we add these people into a special group, called *excluded friend list*. It is also possible that an adversary, such as a neighbor or a colleague of the victim, may be familiar with the victim and his relatives who may be in his *excluded friend list*. The adversary can forge the victim's identity on OSN sites and send friend requests to these users who are in the victim's *excluded friend list*. If such a friend request is accepted, this faked identity may appear to be more authentic and hence it may be easier for the adversary to add the victim's other friends who do not know the victim's potential *excluded friend list*.

Based on the above analysis, we characterize the friend networks of a faked identity as follows:

- ***Friend List* as target**: An adversary creates a faked identity that has several common friends with the victim. F1 in Figure 2 belongs to this type. In F1, all of friends of the faked identity are also the friends of the victim.
- ***Recommended Friend List* as target**: An adversary creates a faked identity and adds some friends who are in the victim's *recommended friend list*. An example is shown in F3 in Figure 2.

- ***Excluded Friend List* as target**: An adversary creates a faked identity and some friends of this faked identity are the users in the victim's *excluded friend list*. An example is shown in F6 of Figure 2.
- ***Combined Friend List* as target**: An adversary creates a faked identity that has some friends, who are in *friend list* of the victim, some friends who are in *recommended friend list* of the victim, some friends who are in *excluded friend list* of the victim, and some friends who are not connected to the victim. Examples of these attacks are shown in F2, F4, F5, F7, F8, F9 and F10 of Figure 2. For instance, F10 shows the most complicated *friend list* of a faked identity that has four mutual friends with the victim, and has one *recommended friend* and one *excluded friend* of the victim. It also has one friend who does not connect with the victim. Specially, in F9, the faked identity does not connect with the victim directly but there is one *recommended friend* and two *excluded friends* of the victim in its *friend list*.

Note that these characteristics introduced above are basic and used for one of our profile similarity schemes: *Basic Profile Similarity*. For *Multiple-FID Profile Similarity*, we introduce several additional characteristics in Section 3.2.4.

3. PROFILE SIMILARITY SCHEMES AND DETECTION PROCESS

Based on the characteristics of a faked identity, we introduce our profile similarity schemes and detection process in this section.

3.1 Profile Similarity

We first introduce definitions of *attribute similarity* and *friend network similarity* in two identity profiles. Several different similarity measures exist in the literature which may be useful to define these two similarity measure such as *overlap similarity*, *Jaccard similarity, dice coefficient* and *cosine similarity*. We adopt the *cosine similarity* [8, 9, 14] in this work because of the following reasons:

- The vectors in our framework are binary vectors and *cosine similarity* is especially suitable for binary data [8]. For computing *attribute similarity*, the vector we expect is the value of *similar* or *not similar* for the same attribute between two profiles. For computing *friend network similarity*, the vector is the value of *mutual friend* (Y) or *not mutual friend* (N) between two sets for *Basic Profile Similarity*, while it is the value of similar friend (Y) or not similar friend (N) for *Multiple-Faked Identities Profile Similarity*.
- The equations using *Cosine Similarity* have special representations in these definitions. In *attribute similarity*, the equation represents the portion of similar attributes in the average size of two sets, while the other equations in *similarity of friend networks* represents the portion of mutual and similar friends in the average size of two sets.

Based on *attribute similarity* and *friend network similarity*, we introduce two profile similarity measures: *Basic Profile Similarity* (BPS) and *Multiple-Faked Identities Profile Similarity* (MFIPS). We consider both of similarities of attributes and friend networks in these schemes for two reasons. First, after an adversary creates a faked identity by forging attributes similar to that of the victim, some victim's friends may add the faked one as their friend without noticing any suspicion, while some of them may notice such attacks and add the genuine one. Thus, some of the victim's friends are in the *FL* of the faked identity, whereas some of his friends are in *FL* of the authentic identity of the victim. When only considering *friend network similarity*, we may overlook the faked identities that have few mutual friends with the identity of the victim but have victim's many friends that do not add the victim as friend in their profiles. Second, some users that have similar names may be cautious enough to not disclose enough attributes to the public. When only considering similarity of attributes, we may make mistakes and identify some of these genuine and similar identities as faked ones.

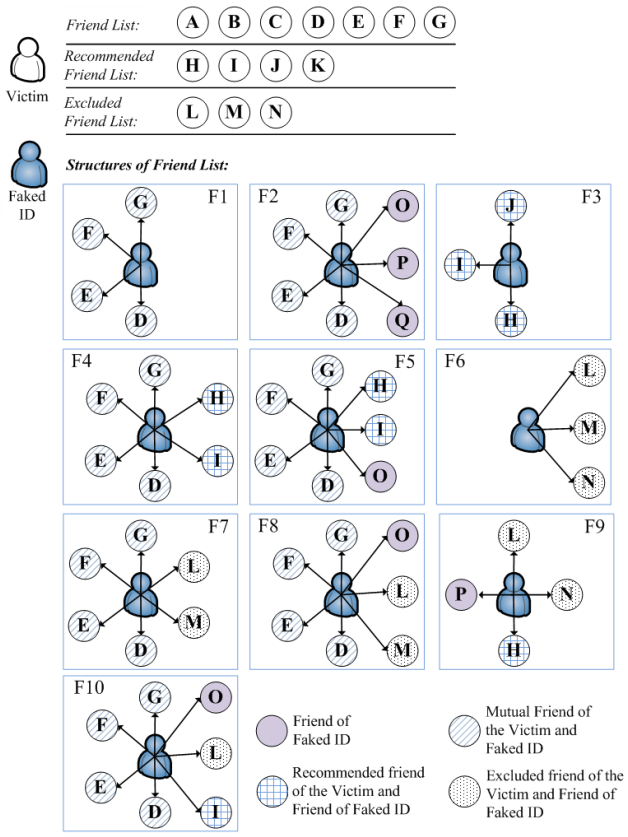

Figure 2. Friend networks of a faked identity

3.1.1 Attribute Similarity Measure

The *attribute similarity* calculates the similarity between two profiles' attributes and its value is based on similar values of the attributes in two profiles. The attribute similarity based on *cosine similarity* is formalized as follows:

DEFINITION 2. *Let P_c be the public profile of a candidate identity c and P_v be the public profile of a victim v. Let SA_{cv} denote the number of the attributes for which P_c and P_v have similar values. We define the attribute similarity of two profiles, S_{att}, as*:

$$S_{att}(P_c, P_v) = \frac{SA_{CV}}{\sqrt{|A_c| \times |A_v|}} \quad (1)$$

where $|A_c|$ and $|A_v|$ represents the number of attributes in P_v and P_v, respectively.

For instance, for the victim in Figure 1, *Name*, *College*, *Phone* and *Birthday* are public attributes ($|A_v|$=4). For the faked identity, *Name*, *College* and *phone* are public and $|A_f|$ = 3. The values of

Name, *College* and *Phone* are same for the victim and the faked identity. Thus, SA_{fv}=3 and $S_{att}=\frac{3}{\sqrt{3\times4}} \approx 0.866$.

3.1.2 Friend Network Similarity Measure

As another key component of our detection models, *friend network similarity* calculates the similarity of two identities' friend networks. As discussed earlier, we should count not only the victim's and the faked identity's *friend list* but also their *recommended friend list* and *excluded friend list*. For the *friend list*, it can be obtained directly from user's profile, when it is set to be public. Users can set their friend network as private and such activity may defend against ICAs. However, most of users do not prefer this activity as it also makes their profiles less popular. The *recommended friend list* is usually generated by an OSN system. This list may be dynamic. In our detection framework, we only consider the *recommended friend list* at the time the detection process runs. For the *excluded friend list*, we assume that a user creates them. For example, a user is able to input a list of e-mail addresses as the identifications of individuals in his *excluded friend list*. Also, a user can input user IDs or profile links on OSN sites to identify his *excluded friend list*.

In order to calculate the of *friend network similarity* between a candidate identity and a victim identity, we first define similarities with respect to *FL*, *RFL* and *EFL* of the victim:

DEFINITION 3. *Let P_c be the public profile of a candidate identity c and P_v be the public profile of a victim v. We define the similarity between the FLs in two identities as S_{ff}, similarity between FL of P_c and RFL of P_v as S_{frf}, and similarity between FL of ~~P_v~~ and EFL of P_v as S_{fef}:*

$$S_{ff}(P_c, P_v) = \frac{|MFF_{cv}|}{\sqrt{|F_c| \times |F_v|}} \quad (2),$$

$$S_{frf}(P_c, P_v) = \frac{|MFRF_{cv}|}{\sqrt{|F_c| \times |RF_v|}} \quad (3),$$

$$S_{fef}(P_c, P_v) = \frac{|MFEF_{cv}|}{\sqrt{|F_c| \times |EF_v|}} \quad (4),$$

where

- *MFF_{cv} denotes the set of mutual friends common in the FLs of P_c and P_v*
- *$MFRF_{cv}$ denotes the set of mutual friends common in FL of P_c and RFL of P_v*
- *$MFEF_{cv}$ denote the set of mutual friends common between FL of P_c and EFL of P_v*
- *$|X|$ represents the number of elements in the set X.*

We define the overall *friend network similarity* for Basic Profile Similarity approach as follows:

DEFINITION 4. *Given a public profile P_c of a candidate identity c and a public profile P_v of a victim identity v, we define the friend networks similarity of these two identities for* BPS *as S_{bfn}:*

$$S_{bfn}(P_c, P_v) = (\alpha S_{ff} + \beta S_{frf} + \gamma S_{fef}),\ \alpha + \beta + \gamma = 1 \quad (5),$$

where α, β and γ are parameters that are used to balance the weights of similarities related to FL, RFL and EFL for the overall similarity of the friend networks in two identities.

We assume that the similarity of *FL*s between two identities is the most important component and the similarity between *FL* of the candidate and *RFL* of the victim is the second most important element. Thus, we set $\alpha > \beta > \gamma$. These three parameters can be estimated by distributions of dataset before the detection process runs. For example, we can estimate their values based on average values of S_{ff}, S_{frf} and S_{fef}. In Section 4, we determine these values by estimating the minimum values of S_{ff}, S_{frf} and S_{fef}.

Note that definition 4 is only used for *Basic Profile Similarity*. We will update this equation for *Multiple-Faked Identities Profile Similarity* in definition 12.

3.1.3 Basic Profile Similarity (BPS)

In the BPS scheme, we assume that the adversary does not create faked identities that forge friends of the victim. In this case, all of the friends in friend networks of both victim and faked identity are assumed to be authentic identities. Therefore, BPS is based on the number of similar attributes and the number of mutual friends in the two identities. We formalize the BPS in two identities as follows:

DEFINITION 5. *Given a public profile P_c of a candidate identity c and a public profile P_v of a victim v, we define the Basic Profile Similarity of these two identities as S_{BPS}:*

$$S_{BPS}(P_c, P_v) = \frac{\sqrt{(\kappa S_{att})^2 + (\chi S_{bfn})^2}}{\sqrt{\kappa^2 + \chi^2}} \quad (6),$$

where κ and χ are the parameters to balance the effect of attribute similarity and friend network similarity on the BPS..

The values of κ and χ can be determined based on the pre-calculated results of *attribute similarity* and *friend network similarity*. Usually, we adjust κ and χ to make similarities of attributes and similarity of friend networks contribute equally to the overall similarity. However, their values can be adjusted based on requirements of the detection. In particular, when $\kappa = 0$ and $\chi = 1$, BPS is switched to the case that only similarity of the friend networks contributes to its result. We specially use this case in our experiments in Section 4 to compare the differences of our two schemes of *profile similarity*.

3.1.4 Multiple-Faked Identities Profile Similarity (MFIPS)

In previous sections, we characterized the friend networks of a faked identity based on assumption that an adversary does not create faked identities that forge friends of the victim. However, an intelligent adversary can create multiple faked identities related to a single victim. In addition to forging a victim, the adversary may also fake the identities of the victim's friends. It is possible that some of the victim's cautious friends notice friend requests from a potentially faked identity and reject such requests. In such scenarios, this attack of faking friends of the victim can be especially effective. In particular, after multiple faked identities related to a victim are created and added as friends of the faked identity targeting the victim, an adversary can forge friend networks of the victim more successfully. Figure 3 illustrates two examples of such ICAs.

As shown in F11 of Figure 3, in order to get *FL* of the victim, an adversary first creates a faked identity *E*' that forges identity *E*. Then he sends friend request from *E*' to the victim. After the friend request is accepted and then *FL* of the victim is exposed, the adversary creates a faked identity to forge the victim. In *FL* of the faked identity, we see identity *K*, who is a *recommended*

friend of the victim, and identity *L*, who is an *excluded friend* of the victim. In addition to these friends related to the victim, the adversary also creates two faked identities - *C*' and *D*' - to forge victim's friends. Lastly, although the faked identity targeting the victim has only one mutual friend with the victim, it has four indirect friends: *C*', *D*', *K* and *L*. In F12 of Figure 3, it is a more successful example of a faked identity whose *FL* is almost the same as that of the victim. In this graph, the faked identity has connected and forged all of the friends of the victim (friends *D*, *E*, *F*' and *G* are connected, friend *B* and *C* are forged by *B*' and *C*'), except friend *A*. Such attack makes it more difficult for the victim's friends to notice the attacks.

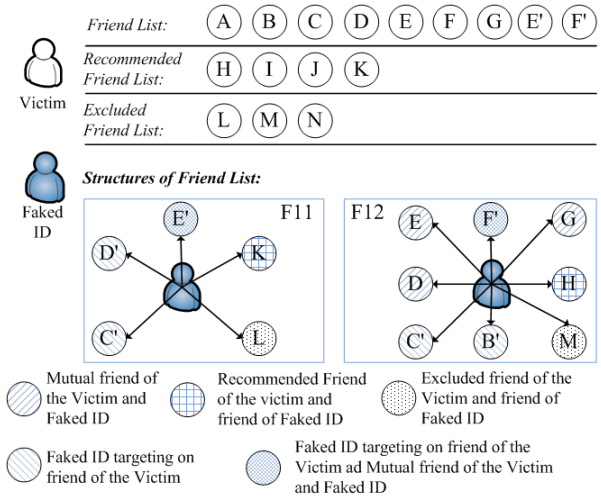

Figure 3. e.g. 1: Multiple faked identities in friend networks

It is worth noting that some features on OSNs may help defend against such an ICA. For instance, in *Facebook*, showing mutual friends between two identities when a user is looking at another user's public profile may help to defend against such ICA that is creating multiple faked identities of the victim's friends. However, these features do not work for the victims who have a large number of friends (e.g. the number of friends larger than 100). This is because it is difficult for a user to remember all the friends he added. Furthermore, an adversary can first create a faked identity of the victim, send friend requests to all of the victim's friends and may successfully add some of the victim's friends who are less cautious. Then, he can create some faked identities that are interesting for the victim, and try to make them as the victim's friends. After that, he can create some faked identities of the victim's friends, make all of them as the friends of the identity that forges the victim and then try to add these faked identities into the victim's friend list. He can repeat these three steps to succeed in adding and forging a large number of the victim's friends. At this point, the number of the mutual friends between the friend list of the faked identity targeting the victim and the victim may be large and complex enough that makes it difficult for the victim's friends to distinguish which one is real. In addition, recent statistic from *Facebook* supports the possibility of our proposed arguments since the average user in *Facebook* has more than 130 friends [2]. Therefore, attacks by creating multiple identities forging the victim as well as his friends cannot be neglected and should be actively detected.

In order to detect multiple faked identities targeting one victim, we update BPS and design a similarity scheme called MFIPS to detect such complicated attacks. In MFIPS, we apply the same definition of attribute similarity but update the equations to calculate the similarity of friend networks. Here, we focus on similar friends between *FL* of the candidate identity and friend networks (*FL*, *RFL* and *EFL*) of the victim.

In the following, we categorize types of attacks related to multiple faked identities targeting one victim while introducing new definitions to capture them.

TYPE A: The adversary first creates identities that forge the identities in the victim's *FL*, *RFL* and *EFL*. Then, these faked identities are added as friends of a faked identity targeting the victim. For instance, in F11 of Figure 3, the faked identity has friends of C' and D'. They are the faked identities of C and D, who are the friends of the victim. However, potential friends of the victim may not discover such attack and add the faked identity of the victim as friend since friend networks of faked identity are similar to those of the victim. As more friends of the victim make wrong decisions and add faked identity as friend, the friend networks of the faked identity become more complex: both authentic friends and faked friends of the victim exists. It may confuse the future friends of the victim and make it more difficult for them to distinguish which one is genuine.

In order to detect such attack, we introduce the following definitions in which similar friends between *FL* of the candidate identity and the friend networks of the victim are captured.

DEFINITION 6. *Let P_x and P_y be the public profiles of two identities. We define P_x and P_y as similar if and only if $S_{BPS}(P_x,P_y)>\mu$, where μ is a threshold of profile similarity.*

DEFINITION 7. *Let P_c be the public profile of a candidate identity c and P_v be the public profile of a victim v. We update the similarity between the FLs in two identities as $S_{s\text{-}ff}$, similarity between FL of P_c and RFL of P_v as $S_{s\text{-}frf}$, and similarity between FL of P_v and EFL of P_v as $S_{s\text{-}fef}$:*

$$S_{s-ff}(P_c,P_v)=\frac{|SFF_{cv}|}{\sqrt{|F_c|\times|F_v|}} \quad (7),$$

$$S_{s-frf}(P_c,P_v)=\frac{|SFRF_{cv}|}{\sqrt{|F_c|\times|RF_v|}} \quad (8),$$

$$S_{s-fef}(P_c,P_v)=\frac{|SFEF_{cv}|}{\sqrt{|F_c|\times|EF_v|}} \quad (9)$$

where,

- *SFF_{cv} denotes the set of similar friends between FLs of P_c and P_v.*
- *$SFRF_{cv}$ denotes the set of similar friends between FL of P_c and RFL of P_v.*
- *$SFEF_{cv}$ denotes the set of similar friends between FL of P_c and EFL of P_v.*

TYPE B: We also consider the influences of similar identities which are in the friend networks of the victim and are also mutual friends of both the faked identity and the victim. In this case, an adversary, who creates a faked identity targeting a victim, can create multiple faked identities corresponding to those in the victim's friend list, add them as friends of the faked identity and try to add them into the friend networks of the victim. After

enough mutual faked identities are added into the victim's friend networks, the faked identity forging the victim will appear authentic to the victim's friends.

First, we capture the case that at least one of the identities in the pairs of similar identities exists in *FL* of the victim. There are two attack schemes in this scenario:

- **Case 1:** Both of the similar identities exist in *FL* of the victim. One of them is a friend of the faked identity. For example, in F11 of Figure 3, E' is a faked identity of E and both of them are the friend of the victim. In addition, E' is also a friend of the faked identity.
- **Case 2:** One of the identities in the pairs of similar identities is in *FL* of the victim and is a friend of the faked identity. The other one in this pair exists in *RFL* of the victim. For instance, in F14 of Figure 4, K' is a faked identity of K that is a *recommended friend* of the victim. It is also a mutual friend of the victim and the faked identity.

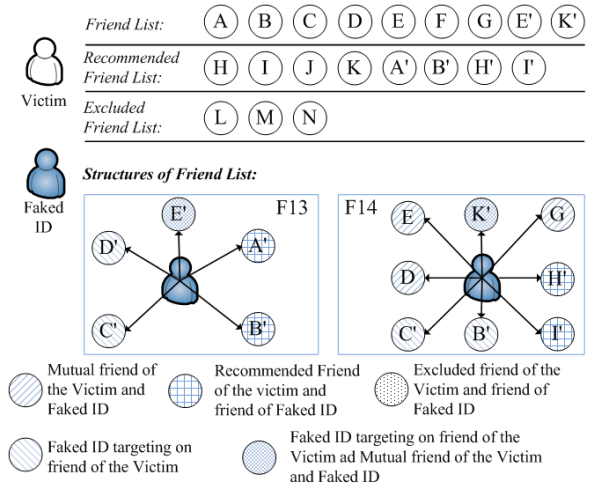

Figure 4. e.g. 2: Multiple faked identities in friend networks

However, we do not count the pairs of similar identities between the victim's *EFL* and *FL*, since identities in *EFL* are the ones that the victim never adds as friends. Thus, the common or similar identities in them do not contribute to the overall similarity.

DEFINITION 8. *Let P_c be the public profile of a candidate identity c and P_v be the public profile of a victim v. We define the similarity $S_{s\text{-}cf}$ to capture the scenarios in Case 1 and 2:*

$$S_{s-cf}(P_c, P_v) = \frac{|SCF1_{cv}| + |SCF2_{cv}|}{\sqrt{|F_c| \times |F_v|}} \quad (10),$$

where

- *$SCF1_{cv}$ denote the set of similar but no same identities between CF and FL of the victim.*
- *$SCF2_{cv}$ denotes the set of similar but no same identities between CF and RFL of the victim.*
- *$CF = F_c \cap F_v$*

Second, we capture the case where at least one of the identities in a pair of similar identities exists in the *RFL* of the victim. There are also two attack schemes in this scenario:

- **Case 3:** Both of the similar identities exist in *RFL* of the victim. One of them is a friend of the faked identity. For instance, in F13 of Figure 5, A' and B' are in *RFL* of the victim and they forge A and B respectively. Also, they are the friends of the faked identity.
- **Case 4:** One of the identities in the pair of similar identities is in RFL of the victim and is a friend of the faked identity. The other one in this pair exists in *FL* of the victim. In F13 of Figure 4, A' and B' exist in *RFL* of the victim and they forge A and B respectively. Also, they are friends of the faked identity.

As in the previous cases, we do not count the pairs of similar identities between the victim's *EFL* and *RFL*. The next definition captures these cases.

DEFINITION 9. *Let P_c be the public profile of a candidate identity c and P_v be the public profile of a victim v. We define the similarity $S_{s\text{-}cfrf}$ to capture the scenarios in Case 3 and 4:*

$$S_{s-cfrf}(P_c, P_v) = \frac{|SCFRF3_{cv}| + |SCFRF4_{cv}|}{\sqrt{|F_c| \times |RF_v|}} \quad (11)$$

where,

- *$SCFRF3_{cv}$ denotes the set of similar (but not same) identities between CFRF and RFL of the victim.*
- *$SCFRF4_{cv}$ denotes the set of similar (but no same) identities between CFRF and FL of the victim.*
- *$CFRF = F_c \cap RF_v$*

Based on definitions 7, 8 and 9, we update the definition of *friend network similarity* and then the profile similarity for MFIPS as follows:

DEFINITION 10. *Given a public profile P_c of a candidate identity c and a public profile P_v of a victim v, we define the friend network similarity of these two identities for* MFIPS *as $S_{mfn}(P_c, P_v)$:*

$$S_{mfn}(P_c, P_v) = \alpha(S_{s-ff} + S_{s-cf}) + \beta(S_{s-frf} + S_{s-cfrf}) + \gamma S_{s-fef} \quad (12).$$

DEFINITION 11. *Given a public profile P_c of a candidate identity c and a public profile P_v of a victim v, we define the Multiple-Faked Identities Profile Similarity of these two identities as S_{MFIPS}:*

$$S_{MFPS}(P_c, P_v) = \frac{\sqrt{(\kappa S_{att})^2 + (\chi S_{mfn})^2}}{\sqrt{\kappa^2 + \chi^2}} \quad (13).$$

Compared to definition 5 which is used for BPS, this definition employs refined notion of *friend network similarity* to capture the influences of multiple faked identities that forge the victim and his friends. We argue that MFIPS is more accurate and is able to discover more suspicious identities, although it increases the complexity of detection process.

3.2 The Detection Process

In order to detect faked identities on OSN sites, we design a detection process. Our detection process is mainly based on the profile similarity measures we introduced above.

3.2.1 Overview of Detection Process

As shown in Figure 5, our detection process has the following phases:

- *Discovery*: Given an *input identity* (*IID*) and a *Profile Set*, we search and collect all the identity profiles that have *Name* similar to that of IID into a *Candidate List* (*CL*).
- *Compute Profile Similarity*: After obtaining *CL*, we calculate profile similarity between each candidate identity in *CL* and

IID based on our profile similarity schemes. When the similarity of a candidate identity for *IID* is larger than a pre-defined threshold, it is added into *Suspicious Identity List* (*SIL*). *IID* is also added into *SIL* since we do not know whether *IID* itself is authentic or not.

- *Validation*: We validate every identity in *SIL* in this step. When an identity is verified to be a faked one, it is put into the *Faked Identity List* (FIL). On the other hand, when an identity is verified to be genuine, its trust value (will be explained in Section 3.5) is increased to avoid future multiple validations. Finally, the faked identities in *FIL* are temporarily closed or eliminated and the existing friends of these identities will receive notifications that their friends are determined to be faked.

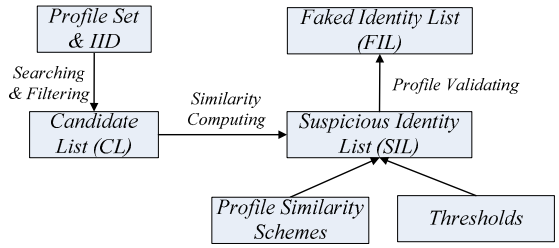

Figure 5. The detection process

Note that we do not do the similarity computation and validation in parallel since validation phase may take significant amount of time depending on the adopted approaches, some of which may involve user responding time.

3.2.2 Thresholds in Detection Process

In our detection process, we have introduced several threshold parameters in order to help to determine the suspicious identities while computing *profile similarity*.

We use threshold parameters ε and δ during *attribute similarity* computation. The ε is used as the threshold for SA_{cv} to indicate that whether we consider this measure in overall similarity. When SA_{cv} is less than ε, we believe that *attribute similarity* will not have significant influence on the value of overall *profile similarity* computed for a given candidate identity and the victim. Then, we set the *attribute similarity* between this candidate identity and the victim as δ, which is the threshold of *attribute similarity*. For example, $\varepsilon = 2$ means that we do not care about the *attribute similarity* for a candidate identity that has only two similar attributes to that of the victim. For δ, it is the minimum value of *attribute similarity*. Sometimes, due to the large number of public attributes in two identities, S_{att} may be less than δ but we set it as δ. This is because we also claim *attribute similarity* contributes less to *profile similarity* in this case.

The third threshold λ we use is to select values of *friend network similarity*. Sometimes, the number of mutual friends in two identities' friend networks may be small compared to the relatively large number of friends in each of them. In this case, similarity of friend networks does not make significant contribution to the results of *profile similarity*. Thus, we set the similarity of friend networks as λ when it is less than λ.

The last threshold we adopt in the detection process is μ for both BPS and MFIPS. We use μ to directly determine whether a candidate identity is suspicious or not. When the *profile similarity* value calculated via BPS or MFIPS is less than μ, we consider this candidate identity as not suspicious identity to the victim; otherwise, we consider it as suspicious and add it into *SIL*.

The values of these thresholds should be estimated before running the detection process. We show how to determine them based on the distribution of the data set in Section 4. Note that there are other thresholds that used in various definitions but we do not discuss those it in this paper.

3.3 The Validation Process

We discuss several possible approaches for validation of the identities once they have been identified as suspicious.

An intuitive approach to identify users is to ask users to input a unique real world ID. For example, *Identity Badge* requires users to input a passport in order to verify them [4]. However, this kind of approach may be difficult to apply for all of the users in the world. Furthermore, since user's unique ID in the real world may be difficult to change, it is hard for the victim to recover from ICA when the adversary knows the victim's unique ID and uses it to pass the identity validation on OSN sites. Another approach currently applied for identifying users is proposed by *mysafeFriend* [5], which is a third-party application in Facebook. It sets five levels of trust for one identity. At the lower levels (level 1 to level 3), it asks an identity to choose his friends to verify himself. The more friends verify this identity, the more points it gets and eventually it gets promoted to a higher level. At the high levels (level 4 and 5), this application checks the identity's credit card to verify it. However, we argue that this is not secure enough to ask user's friends directly verify whether a user's identity is genuine no not. It may be better to ask user's friends to design questions for a user and validate the answers from him. The validation of an identity should be based on the percentage of questions that a user answers right. In addition, we claim that asking users to input their private information, such as credit card, may not be effective as users are likely to avoid them.

Social authentication, which is similar to the identification method of *mysafeFriend*, is proposed by *Schechter et.al* [10]. In this approach, when the user has answered most of the questions from his friends correctly, he is verified to be valid. We believe that social authentication is the ideal approach for identifying users on OSNs because this method does not request any personal information for validation. However, the current approaches of social authentication are not secure enough and have not been well tested. There are several problems of these identifying approaches that need to be solved before they can be used in practical environments:

- A mechanism is required for an identity to choose his appropriate friends, not suspicious identities, to design or select questions to identify him.
- A research on how to design or select questions to identify identities effectively on OSN systems.
- This identifying approach based on the questions should not expose more private information of identities. Otherwise, the adversaries may get more private information of identities when faked identities created by them are chosen by the validation service.

Another potential approach we consider is to monitor and analyze activities of the faked identities, such as their login times and times of checking friends' posts. However, we cannot guarantee that it works until a practical research is applied and we can monitor suspicious identities for substantially long period of time.

Finally, the validation process may verify the duplicated identities of a user as the faked identities when the user refuses to respond to the validation processes for his multiple identities. In addition, there is a possible case that some real identities may fail during the validation process for various reasons, such as not responding to validation questions within a certain time period. As a result, an appeal mechanism should be applied for the identities that have been erroneously flagged as a faked identity.

3.4 Detection Resistant for Future Attack

It is possible that the faked identities may claim they are authentic and request validation many times. In addition, as the candidate identity of the validation process, the victim may be added into *SIL* and need to be verified. Therefore, the genuine identities may be asked to validate too many times, which is inconvenient. In order to prevent too many validations for identities, we define the number of times a user passes the validation as his trust value. After *SIL* is generated, we refine the similarity of every identity in this list as per definition below:

DEFINITION 12. *Given a public profile P_c of a candidate identity and its result of profile similarity ω_c (calculated by BPS or MFIPS) to a public profile P_v of a victim, we refine it as ω_c^r by counting the times of validation the candidate identity has passed:*

$$\omega_c^r = \omega_c * a^{K_c}, a \in (0,1), K_c \geq 0, K_c \in N \quad (14)$$

where a is a pre-defined parameter to reduce the value of the similarity between P_c and P_v, and *K_c is the times of successful validations (trust value) of P_c.*

The value of *a* needs to be determined empirically to suit the requirements of the detection phase. Based on this value, when a user has been validated enough number of times, its profile similarity value with respect to a victim will be reduced to be less than a threshold μ, and will be removed from *SIL*. We claim that his trust value will be reset to zero once a user alters any value of his attributes. This setting is used to avoid an attack where an adversary creates a genuine identity and wins enough trust value at first and then he changes some values of his attributes to forge the victim and clone the victim's friend networks as well.

4. SIMULAITONS AND EXPERIMENTS

In this section, we present our experimental results to validate out detection models. We used offline dataset of *Facebook* [16] and added *recommended friends* and *excluded friends* of users in them for our detection process. We do not validate our detection framework in a real system because it is difficult to find verified identities and their suspicious identities on OSN sites. Also, we cannot create faked identities into the real system as such activity interferes with OSN sites [1]. In the offline dataset, we assumed a special set of identities as faked identities, in order to verify our detection framework.

4.1 Data Initialization

In the original *Facebook* data, the total number of users is 63,731 (User ID is from 1 to 63,731), the total friend links are 1,634,115, with the average user's friend links of 25.6 [16]. Specially, there are 60,102 users that have the whole friend networks, with the remaining 3,629 users who may not show their full friend links in this dataset because they appear in the friend list of users but do not exist in the profile ID index.

In order to implement our detection schemes, we have updated this data by creating user's attributes, recommended friends and excluded friends for every user:

- **Attribute Generation**: in our updated data, we have ensured that every user has 2 to 10 non-hidden attributes.
- **Recommended Friend Generation**: the number of the recommended friends for a user is a random number between 10 and 42 and these friends are random users whose IDs are from 1 to 63,731, excluding the users that are already in his friend list.
- **Excluded Friend Generation**: they are also random users from 1 to 63,731 with the exclusion of the existing users in this user's *friend list* and *recommended friend list*. The total number of the excluded friend for a user is a random number between 5 and 40.

We also need to choose the victims and assume faked identities in this data set.

Victim Selection

We have randomly selected 179 users (1%) from 17,880 users (excluding the 3,629 users who may not show their full friend links), who have more than 25 friends, as the victims. These victims have 11,723 friend links in total and have average friend links of 65.5 with the minimum and maximum friend links of 26 and 266 respectively. The reason we selected such victims is that adversaries usually like to forge victims who are popular on OSN sites [1].

Faked Identity Assumption

We assumed that the 3,629 users, who do not show their full friend links in original data, are the faked identities that randomly forge the 179 candidates (average faked identities for a victim is 20.3). The reason we set these users as faked identities is that they may not show their full friend links and then it may not affect a lot when friends of victims are added into their friend lists, in order to make them forge the victims.

We also assume that similar attributes between a victim and a faked identity is a random number between 2 and the minimum number of public attributes of the victim and the faked identity. We also randomly created 25 to 50 friend links and added into *friend list* of each faked identity. These links are chosen from corresponding victim's *friend list*, *recommended friend list* and *excluded friend list*. Besides 88,494 friend links that already exist in the faked identities (we infer this number based on the fact that A must be in B's friend list if B is in A's friend list), we have created 136,527 faked links with an average faked identity's friend links of 62.0 (minimum and maximum friend links of 26 and 759 respectively). Table 1 shows the statistic of our data.

Table 1. Statistics of the dataset

Total Users	**Original Friend Links**	**Original Average Friend Links**
63731	1634115	25.6
Victims	**Victims' Friend Links**	**Victims' Average Friend Links**
179	11723	65.5
Faked Identities	**Friend Links of Faked Identities**	**Average Friend Links of Faked Identities**
3629	225021	62.0
Victims' Public Attributes	**Public Attributes of Faked Identity**	**Similar Attributes**
2 to 10	2 to 10	2 to min (V's attributes, F's attributes)

4.2 Initialization of Parameters

Before running experiments, we should determine the appropriate parameters in the detection framework. In this section, we set and estimate these parameters by computing their average values and minimum values.

Setting parameters for *attribute similarity*

In our experiments, we first set the minimum value of ε as 2. Based on the attribute distribution in data initialization – an identity has 2 to 10 public attributes, we infer the minimum value of S_{att} is 0.2 (refer to equation 1). We also set the minimum value of δ is equal to 0.2.

Estimating parameters for *friend network similarity*

We first infer α, β and γ in the equations 5 and 12. The values of these parameters can be estimated by the minimum sizes of friend networks of faked identities. According to data initialization, the average size of a faked identity's *FL* is 62, while the average sizes of its *RFL* and *EFL* are 26 and 22.5 in our simulation. Thus, we get $\alpha:\beta:\gamma=\sqrt{62}:\sqrt{26}:\sqrt{22.5}\approx 1.7:1.1:1$ (refer to equation 5 or 12), in order to ensure that *FL*, *RFL* and *EFL* equally contribute to the *friend network similarity* in two identities. In our experiments, however, we assume that the contribution of *FL* to *friend network similarity* is the largest while contribution of *RFL* is the second largest. Then, we set $\alpha:\beta:\gamma$=5:3:2 and get α=0.5, β=0.3 and γ=0.2.

Next, we estimate the minimum value of *friend network similarity* in two identities is approximately $0.2\times 26/\sqrt{40\times 759}\approx 0.03$, where 26 is the minimum value of mutual friends in two identities, 40 is the largest size of *EFL* for a victim and 759 is the largest size of *FL* for a faked identity (refer to equation 5). Thus, λ should be no less than 0.03.

Estimating parameters for *profile similarity*

Based on the previous settings of *attribute similarity* and estimations of *friend network similarity*, we calculate *attribute similarity* and friend *network similarity* for all pairs of faked identity and victim based on equations 2 and 5. We then get the average values of *attribute similarity* and friend *network similarity* for an identity as 0.56 and 0.31 respectively. Since 0.56: 0.31≈1.8:1 (based on equation 6), we set κ=1 and χ=1.8 in order to balance the contribution of the similarities of attributes and friend networks to the overall *profile similarity* in two identities. Finally, we get the minimum value of μ as 0.1 by using minimum values of *attribute similarity* (0.2) and friend *network similarity* (0.03) in equation 6.

4.3 Experimental Results and Analysis

In order to demonstrate the effects of the key parameters in detection results, we first design four experiments for the BPS scheme to illustrate the sensitivity of μ, ε, δ and λ under the given values of κ and χ.

Relationship between μ and detection results

In Figure 6, with ε = 2, δ = 0.2 and λ = 0.03, the detection result drops quickly from 100% to 11.13% with μ increasing from 0.1 to 0.5. It shows that our detection results are very sensitive to μ.

Relationship between ε and detection results

Figure 7 demonstrates the sensitivity of ε for δ = 0.2, λ = 0.03 and μ = 0.2. The experimental result illustrates that the detection result drops slowly when ε is larger than 6. Thus, we infer that the sensitive values of ε are the integers between 2 to 5 (20% to 50% in total number of attributes).

Relationship between δ and detection results

The experiment that shows the trend of detection results with the values of δ under the condition of ε=2, λ=0.03 and μ=0.4 is illustrated in Figure 8. This figure shows that detection result increases gradually with the increasing of δ from 0.2 to 0.8.

Relationship between λ and detection results

Figure 9 illustrates the influence of λ in detection. With ε = 2, δ = 0.3 and μ = 0.3, the detection result increases slowly for λ from 0.12 to 0.27. However, in the range of λ between 0.27 and 0.3, the result increases sharply. This is because current minimum value of *friend network similarity* is 0.3. This makes the minimum value of the overall *profile similarity* to be larger than 0.3, which is the current threshold of overall *profile similarity*.

Contributions of friend network similarity to detection results

Next, we design an experiment to demonstrate influence of only the *friend network similarity* on the detection result. This experiment can be regarded as a supplementary to compare detection results between BPS and MFIPS.

To do this, we first filter the candidates by setting ε = 2. When the number of similar attributes between a candidate identity and the victim is not less than 2, we set it as the suspicious identity and calculate its similarity to the victim. Then, we remove the influence of *attribute similarity* in the detection results by setting κ=0 and χ= 1. The result of this experiment is illustrated in Figure 10. The detection result decreases slowly with μ increasing from 0.1 to 0.2. At μ beyond 0.2 the detection result drops quickly to a value around 10%. The comparison of this experiment with the first experiment presented above is also demonstrated in the same figure. After comparing their differences, we find that *friend network similarity* influences the overall *profile similarity* significantly when the detection results are from 90% to 10%. Thus, we may only consider *friend network similarity* for *profile similarity* in MFIPS when comparing its detection results to that of BPS.

Comparisons of detection results of MFIPS and BPS

In this experiment, we only consider *friend network similarity* for *profile similarity* in both of BPS and MFIPS .We adopt ε = 2, δ = 0.3, λ=0.03 and apply the same values of α, β, γ for both MFIPS and BPS. Our experimental results are illustrated in Table 2. The MFIPS approach detect more faked identities than BPS, e.g. it detects 114 more faked identities than BPS when μ=0.5. However, MFIPS does not change the percentages of the detection results significantly, compared to the detection results of BPS. The main reason is that we randomly chose 179 candidates as the victims from 17,880 users and there are fewer similar and faked identities in friend networks of these victims. Considering the key difference between BPS and MFPIS that we count similar identities rather than mutual identities in MFIPS when computing *profile similarity*, these experimental results are expected. This is because the number of similar and faked identities are not enough large that it cannot update the percentages significantly, compared to the larger denominator (3,629).

5. RELATED WORK

The clone attack (or node replication attack) in sensor networks is similar to ICA on OSNs [11]. In this attack, an adversary captures only a few of nodes, replicates them and then deploys arbitrary number of replicas throughout the network. Then, the adversary can carry out many internal attacks through these compromised nodes.

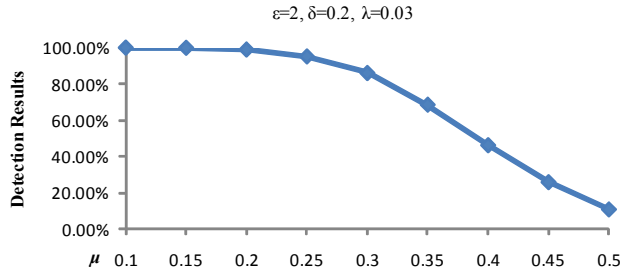

Figure 6. Sensitivity of μ and detection results

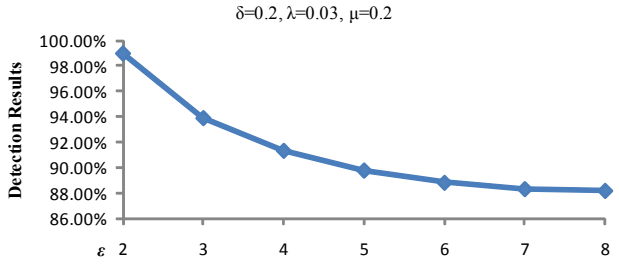

Figure 7. Sensitivity of ε and detection results

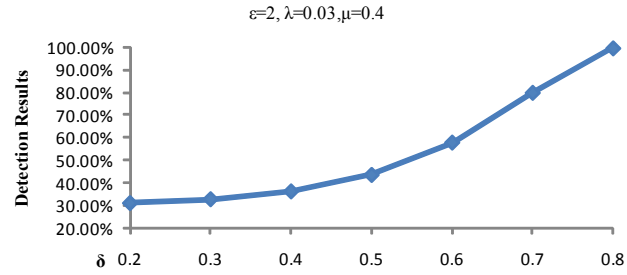

Figure 8. Sensitivity of δ and detection results

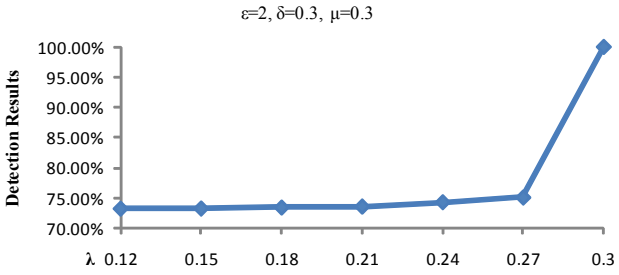

Figure 9. Sensitivity of λ and detection results

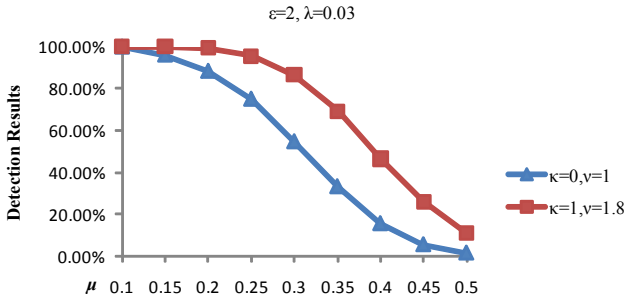

Figure 10. Comparisons of with and without counting attribute similarity

Table 2. Comparisons of BDM and MFDM

BDM(ε=2, δ=0.2, λ=0.03), MFDM(ε=2, δ=0.2, λ=0.03,η=1)									
μ	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Detected ID (BDM)	3629	3627	3592	3455	3124	2492	1684	947	404
Detected ID (MFDM)	3629	3629	3599	3469	3149	2532	1742	1029	518
Differences	0	2	7	14	25	40	58	82	114

There are various approaches proposed to detect such clone attacks. One of the significant mechanisms is proposed by *Parno et.al* [11]. They propose two protocols to provide the globally-aware distributed node-replication detection systems. Their first protocol distributes location claims to a randomly selected set of witness nodes while the second protocol exploits the routing topology of the network to select witnesses for a node's location and utilizes geometric probability to detect replicated nodes. Another significant detection method is proposed by *Zeng et.al* [12]. They update the *Randomized Multicast* to propose two non-deterministic and fully distributed protocols to reduce the cost of communication based on random walk.

However, the detection process in sensor network is different from the detection of faked identities in OSNs. The nodes in sensor network are usually active in order to communicate with each other, while the identities in social networks may not be such active - an identity does not need to interact or cooperate with others all the time. For social network sites, the first step for detecting faked identities is to discover similar identities, while similarity discovering in sensor network may be negligible because the cloned nodes are same as the victim nodes. However, the detection approaches in sensor networks may inspire us to design the validation process to verify faked identities in OSN sites, e.g. how to select users' friends for validation.

On the other hand, the approaches of searching and matching profiles in social networks may be similar with the detection of similar identities on social sites. In this scenario, *Bilge et.al* present an algorithm for matching individuals across social networks to conduct cross-site identity clone attack, based on attribute similarity, such as the similarities of name and education [1]. *Motoyama et.al* develop a system for searching and matching users on Facebook and MySpace via a boosting algorithm based on users' attributes [13]. *Markines et.al* summarize several folksonomy-based similarity measures and evaluate them in finding similar social tags [9]. *Xiao et.al* propose a *positional filtering* principle, which exploits the ordering of tokens in a record and leads to upper bound estimates of similarity scores, to achieve better qualities and improve the runtime efficiency in detecting near duplicate Web pages [15]. Compared to these approaches, our method is based on not only the similarities of attributes or items in entities but also the impact of friend network similarity when discovering the similar profiles.

Our proposed detection process may be extended by an algorithm proposed by *Bayardo et.al* [14]. In the beginning of detection process, we filter profiles by their similar names. However, we may overlook some ICAs where an adversary creates faked identity of the victims using their nicknames. It may be a more secure approach to calculate profile similarities of each profile with the victim and then select the profiles whose similarities with the victim is above the threshold. However, this is a time consuming task. Based on *Bayardo*'s novel indexing and optimization strategies that solve this problem without relying on approximation methods or extensive parameter tuning, we may achieve this goal in an acceptable time.

6. LIMITATIONS

In this section, we discuss some of the limitations of our proposed detection mechanisms:

- Some individuals never use the social network systems. When adversaries get enough personal information of these individuals, they can create faked identities without any misgiving to forge them and deceive their friends. Even, adversaries can create faked identities to forge the individuals who do not exist in the real world. After they successfully add some friends into the faked profiles and get their personal information, these friends probably become the victims in future. Our detection process cannot detect such attacks, as the victims do not register accounts on OSNs.
- Our detection framework can detect the existing faked identities on the OSNs but cannot defend against ICAs in future. This is because the smart adversary can first set up a genuine identity that never uses social networks or an identity that never exists in real world. After having achieved enough trust, he can alter attributes of this identity, such as name and gender, to forge a victim. But we argue that our framework can detect such attacks if OSN sites execute our detection process at the moment when a new profile is created or the sensitive attributes of an existing profile are updated.
- Although we validated our detection schemes by offline *Facebook* data, the detection schemes have not been tested on real world OSN systems. The main concern is that it is difficult to find known faked identities on OSN sites and use them to validate our models. In addition, creating some faked identities on OSNs for the validation will bring big damages to the sites. We plan to develop a Facebook application, implement our detection process and make it popular in *Facebook* users. After that, we can validate our models in the real system.

7. CONCLUSIONS

Identity clone attacks are becoming a significant threat in OSNs. It can severely affect the trust relationships among users in the OSN sites and expose users' personal information. In this paper, we have proposed two detection schemes to discover the potential faked identities. Through our experiments, we have demonstrated that our detection schemes are feasible and effective. We have also shown that our detection schemes can be easily adapted to other data with the different distributions by tuning various parameters. In addition, we have discussed several approaches to validating suspicious identities.

As future work, we plan to address the limitations and develop a *Facebook* or *Linkedin* third-party application to implement our proposed detection schemes in a more real OSN environment. We also plan to investigate the social authentication mechanism and the validation and the appeal mechanisms discussed earlier.

8. ACKNOWLEDGMENTS

This work has been supported by the US National Science Foundation award IIS-0545912. We would also like to thank *Alan Mislove* and his group for providing us the original *Facebook* data.

9. REFERENCES

[1] L. Bilge, T. Strufe, D. Balzarotti and E. Kirda, "All your contacts are belong to us: automated identity theft attacks on social networks", In Proceedings of the 18th international conference on World Wide Web, Madrid, Spain, 2009.

[2] Facebook Factsheet [Online]. http://www.facebook.com/press/info.php?statistics.

[3] Wikipedia. Social network [Online]. http://en.wikipedia.org/wiki/Social_network.

[4] Identity Badge [Online]. http://apps.facebook.com/identity_badge

[5] mysafeFriend [Online]. http://apps.facebook.com/mysafefriend

[6] FightIDTheft [Online]. http://www.facebook.com/FightIDTheft

[7] Facebook Identity Theft - Don't be a Victim! [Online]. http://www.facebook.com/group.php?gid=7086517815

[8] E. Spertus, M. Sahami and O. Buyukkokten, "Evaluating similarity measures: a large-scale study in the orkut social network", In Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 678 – 684, Chicago, IL, USA, 2005.

[9] B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme, "Evaluating similarity measures for emergent semantics of social tagging", In Proceedings of the 18th international Conference on World Wide Web, pp. 641-650, Madrid, Spain, 2009.

[10] S. Schechter, S. Egelman, and R.W. Reeder, "It's not what you know, but who you know: a social approach to last-resort authentication", In Proceedings of the 27th international conference on Human factors in computing systems, pp. 1983-1992, Boston, MA, USA, 2009.

[11] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks", In Proceedings of the IEEE Symposium on Security and Privacy, pp. 49–63, Oakland, CA, USA, 2005.

[12] Y.Zeng, J.Cao, S. Zhang, S.Guo, and L. Xie,"Random-walk based approach to detect clone attacks in wireless sensor network," IEEE Journal on Selected Areas in Conmmunications, Vol. 28, No. 5, pp. 677 – 691, 2010.

[13] M. Motoyama and G. Varghese, "I seek you: searching and matching individuals in social networks", In Proceeding of the 11th international workshop on Web information and data management, pp.67-75, Hong Kong, China, 2009.

[14] R. J. Bayardo, Y. Ma, and R. Srikant, "Scaling up all pairs similarity search", In Proceedings of the 16th international Conference on World Wide Web, pp. 131-140, Banff, Alberta, Canada, 2007.

[15] C. Xiao, W. Wang, X. Lin and J. X. Yu, " Efficient similarity joins for near duplicate detection", In Proceeding of the 17th international Conference on World Wide Web, pp. 131-140, Beijing, China, 2008.

[16] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in Facebook", In Proceedings of the 2nd ACM Workshop on online Social Networks, pp. 37-42, Barcelona, Spain, 2009.