I.J. Intelligent Systems and Applications, **2015, 03, 13-20**
Published Online February 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2015.03.02

A Framework for Detecting Cloning Attacks in OSN Based on a Novel Social Graph Topology

Ali M. Meligy
Dept. of Mathematics, Computer Science, Faculty of Science, Menoufyia University, Shebien El Koom, Egypt
E-mail: meligyali@hotmail.com

Hani M. Ibrahim, Mohamed F. Torky
Dept. of Mathematics, Computer Science, Faculty of Science, Menoufyia University, Shebien El Koom, Egypt
E-mail: hanimir78@yahoo.com, mtorky86@gmail.com

Abstract— Online Social Networks (OSN) are considering one of the most popular internet applications which attract millions of users around the world to build several social relationships. Emerging the Web 2.0 technology allowed OSN users to create, share, or exchange types of contents in a popular fashion. The other hand, OSN are considering one of the most popular platforms for the intruders to spread several types of OSN attacks. Creating fake profiles for launching cloning attacks is one of the most risky attacks which target Users' profiles in Online Social Networks, the attacker seek to impersonate user's identity through duplicating user's online presence in the same or across several social networks, therefore, he can deceive OSN users into forming trusting social relations with his created fake profiles. These malicious profiles aim to harvest sensitive user's information or misuse the reputation of the legitimate profile's owner, as well as it may be used as a spy profiles for other criminal parties. Detecting these fake profiles still represent a major problem from OSN Security and Privacy point of view. In this paper we introduced a theoretical framework which depends on a novel topology of a social graph called *Trusted Social Graph (TSG)* which used to visualize trusted instances of social communications between OSN users. Another contribution is a proposed detection model that based on TSG topology as well as two techniques; Deterministic Finite Automaton (DFA) and Regular Expression. Our proposed detection model used to recognize the stranger instances of communications and social actions that performed using fake profiles in OSN.

Index Terms— Fake Profiles, Cloning Attack, Trusted Social Graph (TSG), Friend Pattern, Regular Expressions, and Deterministic Finite Automata (DFA).

I. Introduction

Online Social Networks (OSN) are considering one of the most recent internet applications which attract millions of users to setup several types of social relationships such as friendships, business or common interests…etc. Emerging the web 2.0 technologies which present a new dynamic version of World Wide Web (WWW), allowed users across online social networks to share, exchange and create several types of contents [1]. Online Social Network services range from social interactions-centered platforms such as Facebook or MySpace, to information dissemination-centric platforms such as twitter or Google Buzz, to Social interaction feature added to existing platforms such as Flicker [2]. The other hand, enhancing security considerations and protecting the OSN privacy still represent a major bottleneck and considered challenge. From a security point of view online social networks have unique characteristics. First, information access and communication models are based on the trust between parties, such that, OSN users typically share vital amount of private and personal information with their friends. This information may be shared in public mode or dedicated private mode. If it is not public, access to it must be regulated by a network of trust, in this case, a user allows to his friends to access and view information regarding him. Second, unfortunately, Online Social Networks haven't strong authentication mechanisms for protecting users' profiles. This matter provides an easy way to impersonate user's identity and to steal in a user's network of trust [3]. In recent days, users' profiles and their personal and private information are representing a main target for many OSN attacks that can exploit the weaknesses of OSN security and privacy mechanisms. These attacks include *Identity theft* [4] by which the intruder able to convince anyone about the ownership of some particular OSN profile. *Profile Hijacking* [5], the goal of the adversary is to obtain control over some existing profiles within OSN platforms. *Profiling attack* [6], by which the intruder aims to gather information about OSN activities. *Crawling and Harvesting attacks* [5], the main goal of crawling attack is to aggregate publicly available information across multiple OSN profiles and applications in an automated way. Another attack by which the attacker crawls across different OSN platforms is called *harvesting attack* by which the harvested results are storing in large dataset with larger amount of private information about OSN users. *Image Retrieval and Analysis Attack* [5], is another attack which target multimedia data such as (images, videos, etc). This attack is followed by subsequent analysis via Reverse Social Engineering attack (RSE). RSE is a specific type of social Engineering attacks by which the adversary doesn't initiate contact with the victim, rather the victim is tricked into contacting with the attacker himself [7].Special type of impersonation attacks is the *Cloning Attack* by which the intruder creates some fake profiles

for one or more users. These fake profiles are claiming the same identity as the genuine profiles in a given OSN. The cloning attack makes the fake profiles able to setup friendships with the victim's friends and eventually mounting malicious activities with them [8]. In other words, the intruder seeks to finds user's personal information such as (name, job, photo of his profile …etc) then forges user's identity and creates a similar profile as a fake profile in OSN. Finally, he sends friend requests to victim's friend list. As soon as these friend requests are accepted and confirmed, a new friend list network is built. Hence, the intruder become able to access friend's profiles of the victim and steal sensitive information or mount other malicious activities such as posting malicious photos or sending spam messages, etc [9].In response to this type of attacks, this paper introduces a novel framework for detecting fake profiles in OSN. This paper involves two contributions; first, we introduced a novel graph topology called *Trusted Social Graph (TSG).* TSG model used to visualizes the instances of trusted communications between OSN users in a novel way which facilitate detecting the un-trusted or fake communications between OSN users. Second, we present a detection model that depends on two terms: *Regular Expressions* and *Deterministic Finite Automaton (DFA)* for detecting fake profiles in Online Social Networks.

The rest of this paper is organized as follows; *section II* presents and discusses the literature review. *Section III* clears some of profile features that may be exploited by Cloning Attack. *Section IV* presents the major proposed definitions and concepts by which our methodology based on it. *Section V* presents the proposed theoretical framework which used for detecting cloning attacks through a proposed Social Graph Topology called a *Trusted Social Graph (TSG)* as well as, a proposed detection model for recognizing malicious social behaviors of fake profiles in Online Social Networks (ONS). Finally, *section VI* formulates the general conclusion and suggests how we can improve this framework in the future.

II. Related Work

Impersonations threats in Online Social Networks (OSN) such as creating Fake Profiles by which OSN hackers use to launch cloning attacks, are considering one of the hot topics from OSN security and privacy point of view. The intruder seeks to create fake profiles to setup fake relations with the victim's friends and to mount malicious activities such as misusing the reputations of OSN users. In Ref [9] profile cloning and identity theft attacks are introduced, and then, a framework for detecting suspicious identity is proposed. This approach is based on attribute similarity and friend network similarity measures. According to similarity measures which are computed in each step and by having predetermined threshold, it will be decided which profile is clone and which one is genuine. In [10] the authors proposed a methodology for detecting profile cloning in OSN, the study presents architectural design and implementation details of a prototype system that can be employed by users to investigate whether they have fallen victims to such an attack. The core idea behind this approach is to identify any information contained in the user's profile that uniquely identifies him. The experimental results from use of this prototype system prove its efficiency and simplicity. In [8] the authors introduced a new model aims to mitigate the threat of the fake profile attack, where an adversary tries to impersonate the victim on a specific OSN where the victim has no prior profile in place. The main idea behind this approach is to study the temporal evolution of OSN and characterize real user profiles where data can be collected and used to identify set of feature in dynamic mode of OSN. These features can be used to study the time evolution of a given test profile and detect any major deviations from expected behavior of a profile. In [11] the study showed how to use *exclusive shared knowledge* to allow users to take responsibility for identifying their own trusted friends in an OSN in a completely online way. The idea is, secret information shared only between a pair of OSN friends, then rules of " *Bond Breaker Game"* are applied which is a practical tool associated with public Key Infrastructure (PKI) for identifying friends in OSN. The results were presented through a study on Facebook, which showed that users who do *share exclusive knowledge* with their Facebook friends, the adversary are rarely able to know or guess that knowledge. M.Fire,G.Katz, and Y.Elovici in Paper [12] presented a novel methodology for detecting malicious profiles in OSN. The proposed methodology used a combination of graph theory algorithms and machine learning in order to detect these malicious profiles, the proposed algorithm had been evaluated on several social networks and was found to be effective in detecting various types of fake profiles. In [4] the authors investigated two automatic attacks: First, *Identity Theft Attack* within one OSN. The attack is simulated in an automated way for existing user profile and sending of friend requests to the friends of cloned victim. The hope of the intruder here is to gain the trust of these profiles. By establishing friendship relationships with the contacts of the victim, hence, the attacker becomes able to access the sensitive personal information of these contacts. The second attack is *Cross-Site Profile Cloning Attack*, which simulated in an automatic way by creating forged profiles in OSN in which the victim isn't belong to it but his friends are registered in it, and then connect to the victim's friends who are registered on both networks. The experimental results showed that these two simulated attacks are effective and feasible in practice. The author also introduced some suggestions for improving security and privacy of OSN in face of these two attacks. In a

similar study [6] that present misusing OSN using automated profiling attack, the authors introduced a prototype system which simulates the fact that an attacker can query several OSNs to obtain registered e-mail addresses on a large scale. By automatically crawling and correlating these profiles that associated with the collected e-mails addresses, personal information about

each user are gathered which used for automated profiling attack. The proposed prototype system is applied on the most popular OSN such as Facebook, MySpace, and Twitter. The results showed that the automated profiling attack executed in efficient way . in addition the results reflect that these social networks still have common weakness and still need for more security precautions to protect the vast amount of personal information stored on these networks. This study also introduced several mitigation strategies that can be used to limit the extent of this attack. One of the common risks of creating fake profiles is to use in launching spam messages [13]. The authors presented a design and real world evaluation of a novel social honeypot-based approach to social spam detection, the research goal is to investigate techniques and develop effective tools for automatically detecting and filtering spammers who target social systems. Another similar study for detecting spammers in Twitter-Social Network [14] introduced a spam detection prototype system to identify suspicious users on Twitter. Architectural social graph model is proposed to explain the "follower" and "friend" relationships among Twitter's users based on the spam policy of Twitter, a novel content-based and graph-based feature are proposed to simplify spam detection process. The author exploited classical classification algorithms to detect suspicious of spam accounts and developed a web crawler using Twitter API method to collect real data set from public available information on Twitter, finally, he analyzed the data set and evaluated the performance of his detection system. Also in [15], the authors discussed the functionality and the implementation of SocialNetworkingBot which is a proof of concept botnet that uses Twitter for command and control of individual bots.

III. Profile Features And Cloning Attacks

The intruders seek to create fake profiles in OSN for launching cloning attack that used to misuse user's profiles through main features of user's profile. We shed light on these features as follows:

A. Personal page

Personal page is considered one of the vital assets in user profile since this page contain personal information such as (name, gender, job, education, photos, … etc). The default seating in OSN such as Facebook is that you able to access on the personal page of any user even he isn't in your friend list. Also, you can share any content without any access control, as well you can send message to any profile without any authenticated prove about the verification of identity of the sender. Although this may be seen as a flexible use in online social networks, but this may be a weakness from security and privacy point of view. Such that the fake profiles that created by OSN intruders able to setup trusted relationships with OSN user through exploiting access flexibility on users' personal page.

B. Home Page

Is the front end of user's profile in which all shared contents such as photos, texts, videos, or some commercial advertisements are appearing through it. The shared contents may be from your friends or may be from others. The OSN attacker may exploit the home page to share advertisement for a specific product or for a specific company through fake page in an attractive layout. Then he invites the victim to enroll to this page without any mechanism for recognizing the identity of this page.

C. Friend Requests List

Is a menu that includes all friend requests that requires choosing one of three options, confirming, ignoring, or delaying the request. Friend requests list is agglutinative contact which the attacker can use to send fake requests to users. As soon as accepting this request, the fake profile will added to your friend list. Hence the fake profile can exploit this permission to mount malicious activities with your friends in the friend list.

D. Messages List

Is a menu that involves all messages you sent to or received from other user in OSN. The other hand, messages list is another channel through which cloning attack can occur. The intruder may send spam message [1, 3] to specific user and this messages may contain malicious link to third party or may be a malicious link for running malicious software or deceive users to join to other malicious web sites. Since there is no an effective mechanism by which we can control in sending and receiving messages across OSNs, the users' profiles may be threaten with spam messages which are sent through fake profiles in OSN.

E. Friends and Close Friends Lists

Friend list is a menu that involves all friends whom user may contact with them through several instances of communications using synchronous or asynchronous messaging. The other side, Close Friend list is a derived menu from the friend list that involves the most important friends for you. The social service provider flags them with "yellow star" as a mark to distinguish them from other friends. In the opposite direction, OSN adversaries seek to obtain victim's friend list, then send fake friend request to them which will be accepted at the most. Such that most smart OSN attackers add some victim's friends to his close friend list to be common with the victim's close friends, this will make the fake profile more genuine and has more trust [9, 14].

IV. Basic Concepts For The Detection Framework

In this section we present some definitions and concepts by which our detection framework depends on them. We import some techniques, Principles, and methodologies that used in compiler when recognizes the language, but we correlated it to adapt with our problem

that explains how to recognize and detect Fake Profiles in Online Social Networks.

Definition 1: (Profile)

Let x be a specific user in an OSN system, we define the profile of x as two tuples (L_x, $(FP)_x$) where:-

1- L_x is an attribute label in the form user's name and specific image by which *user* x is digitally represented in specific OSN.

2- $(FP)_x$ is a unique *Friend Pattern (FP)* of *user* x that used as *a rule* by which a set of trusted profiles (i.e. trusted users) in the friend list of *user* x can be recognized. In addition, each Friend Pattern (FP) is represented mathematically as a *Regular Expression* over specific Alphabet $\sum$.

Definition 2: (Friend List)

From authentication point of view, we define the Friend List of *user x* in specific OSN as a *Regular Set* that includes some of *Rules (R).* Each *rule* is represented as two concatenated *Friend Patterns* in the form $R_i = FP_x.FP_y$. Such that FP_x is the friend pattern of *user* x. and FP_y the *Friend Pattern* of *user* y.

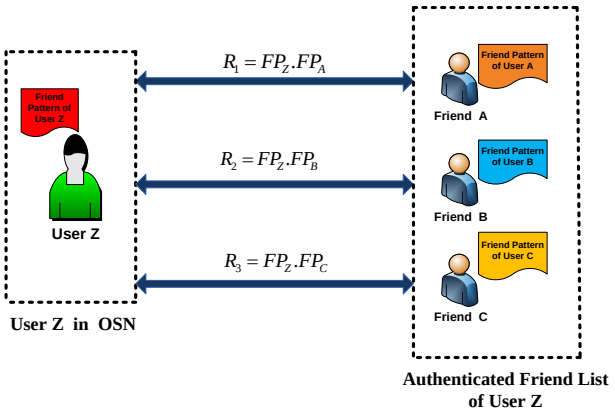

Fig. 1. a novel view for representing a user's Profile and the associated Friend List in OSN

Fig.1. Show an authenticated way by which a specific user in a specific OSN can set up social relations between him and his friend list. For example, the *Rule/protocol* which authenticate the social relation between *User Z* and his *Friend A* is; $R_1 = FP_Z.FP_A$ .Such FP_Z *and* FP_A is the friend patterns of *user Z* and *friend A* respectively. In addition, each Friend Pattern is represented mathematically as a *Regular Expression* as we cleared in Definition 1.

Definition 3: (Finite Automaton FA)

Is a mathematical model used as a recognizer for regular set of strings that derived from a specific *Regular Expression*. A Finite Automaton consists of five tuples:

$$FA = (Q, \sum, \delta, q_0, F)$$

Such that

Q is a non empty finite set of states, $\forall\ q \in Q$, q is called state of Q.

$\sum$ is an input alphabet such that the derived strings of specific Friend Pattern is from $\sum$.

δ is a transition function that maps from state to another state in Q over input symbol of the alphabet $\sum$, such that, $\delta : Q \times \sum \rightarrow Q$.

q_0 is the initial state of FA.

F is a finite set of final state of FA, $F \subseteq Q$, $\forall\ q \in F$, q is called the final state.

As we see in Fig. 2. that presents an example of *Finite Automata (FA)* that can recognize all strings that derived from the Regular Expression: a^*bac^*, such as (*aabacc, aaabaccc, abacccc, aaaaabac,* etc). Really, *the Finite Automata (FA)* doesn't be a novel concept, however, we defined it here since we use it as an effective tool for recognizing *Friend patterns* which represented as a Regular Expressions to accept or reject profiles' social actions in Online Social Networks.

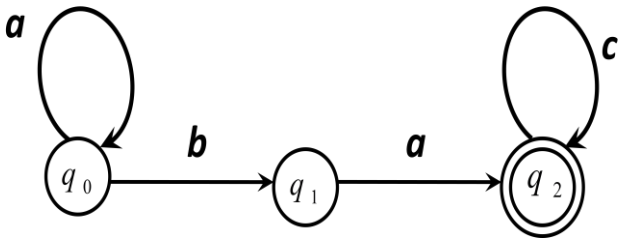

Fig. 2. Finite Automata for recognizing all strings derived from the Friend Pattern: a^*bac^*.

According to *Definition 3* we have:

1- $Q = (q_0, q_1, q_2)$.

2- $\sum = \{a, b, c\}$.

3- δ can be defined as:

$$\delta_1(q_0, a) = q_0$$
$$\delta_2(q_0, b) = q_1$$
$$\delta_3(q_1, a) = q_2$$
$$\delta_4(q_2, c) = q_2$$

4- The initial state is q_0

5- The set of final state is $F = \{q_2\}$

Definition 4: (Genuine Profile)

A specific profile of *user* y in OSN is said to be a *Genuine Profile* to *user* x ,*if and only if,* there exist a *Deterministic Finite Automat (DFA)* that associated with the profile of *user x* and able to recognize the authentication Rule/Protocol between *user x* and *user y*.

Definition 5: (Fake Profile)

A specific profile of *user* y in OSN is said to be a ***Fake Profile*** to *user* x *if and only if,* the *Deterministic Finite Automat (DFA)* that associated with the profile of *user x* able not to recognize the authentication Rule/Protocol between *user x* and *user y*.

For visualizing our framework about detecting fake profiles in Online Social Network (OSN) , we need to define special type of graphs called *DeBruijn Graph* [16] on which our proposed social graph topology is based on it.

Definition 6: (DeBruijn Graph)

In graph theory, a graph of order ***k*** denoted by ***G (k)*** is said to be *DeBruijn Graph* if it was directed graph with 2^k vertices, each labeled with a unique *k-bit string*. *Vertex* ***a*** is joined to *vertex* ***b*** by an arc if a bit string of ***b*** is obtainable from bit string of ***a*** by either *cyclic shift* or *DeBruijn shift*. Additionally, each arc of ***G(k)*** is designated as *cyclic shift arc* or *debruijn arc*. According to the shift operation, each arc is labeled by the first bit of the vertex at which it originates, followed by the label of the vertex at which it terminate.

Fig.3.depicts a Debruijn Graph of order 3 (i.e. $G(3) = 2^3$=8 vertices) and the bit-strings associated with these vertices are: 000, 100, 001, 010, 101, 110, 011, and 111.

In DeBruijn graph example a k- bitsting $b = b_1 b_2 b_3 .....b_k$ is said to be obtainable from a k-bitstring $a = a_1 a_2 a_3 .....a_k$ (i.e. *there is a directed relation from vertex a to vertex b*) by a left shift operation if $b_i = a_{i+1}$, for $i = 1,2,3,.....k-1$.

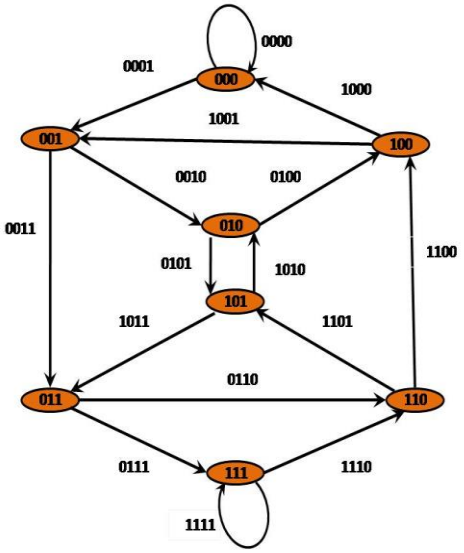

Fig. 3. An example of DeBruijn graph of order 3: G (3) =8 vertices.

V. The Proposed Detection Framework

We suggest that we can use the DeBruijn graph to derive a novel social graph topology called "*Trusted Social Graph (TSG)* ". The proposed graph topology can visualize social relations in an authenticated manner. Hence, detecting fake profiles and recognizing trusted friends in OSN became a possible issue through the proposed approach. Our new social graph *(TSG)* has common properties and features with DeBruijn graph, but it also has proposed properties that differ from DeBruijn graph model. The new properties are proposed to adapt with our detection methodology. We can formulate the general definition of our proposed *Trusted Social Graph model (TSG)* as in the next definition;

Definition 7 (Trusted Social Graph (TSG))

Trusted Social (TSG) is a directed graph; *TSG (V, E)*, such that each vertex $v \in V$ is defined as three tuples,

$$v = (FP_{(v)}, L_{(v)}, DFA_{(v)})$$

Where,

$FP_{(v)}$: is a Friend Pattern of *vertex* v that is represented mathematically as a unique *Regular Expression* over a specific *Alphabet* $\sum$.

$L_{(v)}$: is an attribute label in the form user's name and specific image of *vertex v* by which the *user* associated with the *vertex v* can commonly represented in OSN.

$DFA_{(v)}$: is *a Deterministic Finite Automaton* which associated with the *vertex* v to recognize the trusted and authenticated relations. Also, $DFA_{(v)}$ used for detecting fake social relations through parsing the predefined *Rule/Protocol* between *vertex v* and other vertices in the *Trusted Social Graph (TSG).*

Also, each edge $e \in E$ from v_1 *to* v_2 is identified with a derived string s_2 of the Friend Pattern of *vertex* v_2 (i.e. $FP_{(v_2)}$) concatenated with a derived string s_1 of the Friend Pattern of *vertex* v_1 (i.e. $FP_{(v_1)}$) in the form " $s_2 . s_1$ " which parsed by the *Deterministic Finite Automata* that associated with *vertex* v_2 (i.e. $DFA_{(v_2)}$

A. Trusted Social Graph (TSG) Topology

As we mentioned above that our TSG model is based on DeBruijn graph with proposed properties and features, before we explain these additional features, we present first the common features between TSG and DeBruijn graph as follows:

1 TSG and DeBruijn graph are directed graph

2 Each arc directed from v_1 *to* v_2 can be identified as a function in v_2 *and* v_1

3 Each vertex $v_i or v_j \in V$ is labeled with a unique code that used to derive a specific string s_i used in identifying the relation between v_i *and* v_j such that v_i *and* $v_j \in V$.

The other side, the novelty features in our proposed TSG model are:

1- Each vertex $v_i \in V$ is described as three tuples $v_i = (FP_{(v)}, L_{(v)}, DFA_{(v)})$ as described in *definition 7*

2- Each vertex $v_i \in V$ has n in-degree and m out-degree such that $n \neq m$, but in DeBruijn graph it has 2 in-degree and 2 out-degree.

3- The Friend Pattern $FP_{(v_i)}$ that associated with specific vertex $v_i \in V$ is used to derive set of strings, which used in represent instances of social communications between vertexes v_i and other *vertex* v_j in the Trusted Social Graph.

4- The in-arcs directed to specific vertex $v_i \in V$ represent instances of social communications from other vertices in TSG model.
5- TSG is a dynamic graph such that it visualizes changeable instances of social communications between OSN users.
6- Any arc from v_i *to* v_j is labeled with a string in the form $S = s_j.s_i$ such that s_j is derived from the Friend Pattern of v_j and s_i is derived from the Friend Pattern of v_i and S represent the *Rule/Protocol* of the Social Communication instance.
7- Each $v_i \in V$ is associated with a Deterministic Finite Automaton $DFA_{(v_i)}$ to parse the in-relations to v_i which labeled with the string $S = s_j.s_i$ such that s_j is derived from the Friend Pattern of v_j and s_i is derived from the Friend Pattern of v_i
8- TSG has no any self loop in any vertex $v_i \in V$.

Our proposed TSG topology is described in Fig. 4**.** that visualizes some instances of social communications between eight users(e.g.. A, B,C,D,E,F,G,H) in a specific Online Social Network. Each node is represented as three tuples, $v_i = (FP_{(v)}, L_{(v)}, DFA_{(v)})$ as we previously mentioned. The Friend Pattern in each node is represented mathematically as a unique Regular Expression over specific Alphabet as presented in Table 1.

Table.1. Friend Patterns of eight users in TSG

# Profile	Alphabet $\sum$	Friend Pattern (FP)
Profile of *user A*	$\sum = (a,b,c)$	ab^*c
Profile of *user B*	$\sum = (0,1)$	010^*
Profile of *user C*	$\sum = (\#,2)$	$\#2^*\#$
Profile of *user D*	$\sum = (y,x)$	y^*xy^*
Profile of *user E*	$\sum = (m,n,o)$	mn^*o
Profile of *user F*	$\sum = (1,0,r)$	1^*0r
Profile of *user G*	$\sum = (k,l,m)$	klm^*
Profile of *user* H	$\sum = (p,q,z)$	pq^*z

For Forming authenticated and trusted social communications between a specific user and his genuine friends of his friend list, this problem requires three phases to mitigate it, as described in Fig.5

1. *Forming the Friend Pattern (FP):* in this phase, a specific Regular Expression is used to set up and form the usable Friend Pattern (FP) of each user in an Online Social Network.
2. *Deriving strings of social communications:* is the second phase, in which the regular set that involves some of strings is derived from the coordinate Friend Pattern of a specific user. The regular set of strings is used in identifying and distinguishing the current social relation between two users in OSN.
3. *Forming the Protocol of Social Events:* is the last phase, in which the Rule/Protocol that used to authenticate social relations between a specific user and his friend list is constructed. The Rule/Protocol between a specific user and his friends can be formed as two concatenated Friend Patterns as: $FP_{User\,A}.FP_{friend_I}$

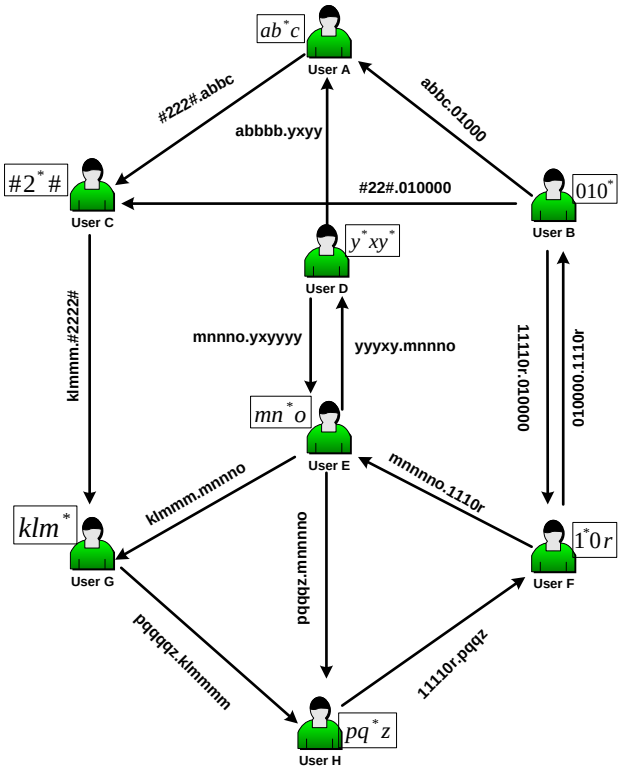

Fig. 4. The Proposed Trusted Social Graph (TSG) Topology

B. *Friend Pattern Recognizing Process in TSG*

In our dynamic *Trusted Social Graph (TSG)* which visualizes the instances of social events between OSN Profiles, each profile (i.e. *a Vertex* in TSG) is associated with *a Deterministic Finite Automaton (DFA)* as mentioned in *Definition 7*. The potential and major role of DFA that associated with a specific user's profile, is to recognize the incoming instances of social behaviors to decide whether the source of a specific social event (such as *friend requests, sending messages, sharing photos or chatting, etc*) is from a genuine and authenticated profile or from a fake and unauthenticated profile. As we saw in TSG topology (back to Fig.4.) in which each *social relation* from $profile_i$ *to* $profile_j$ is labeled with a string $S = s_j.s_i$ such that s_j is a string derived from the Friend Pattern of $user_j$, and s_i is a string derived from the Friend Pattern of $user_i$. The DFA that is defined in the next definition is used to recognize the incoming

social events to a specific user's profile through parsing the string that associated with a specific social event. Our proposed DFA that associated with each profile can decide to accept or reject a specific social action according to the predefined authenticated friend list of this profile (back to Fig.5.). Cloning Attack is considered one the active attacks in Online Social Networks since there is no an authenticated mechanism between a specific user and his friend list ensure specifying the identity of the source of social behavior. This vulnerability led to that the attacker can duplicate his presence with multiple profiles with the same layout. In response to this problem, we think that our proposed approach mitigates this problem in an effective manner.

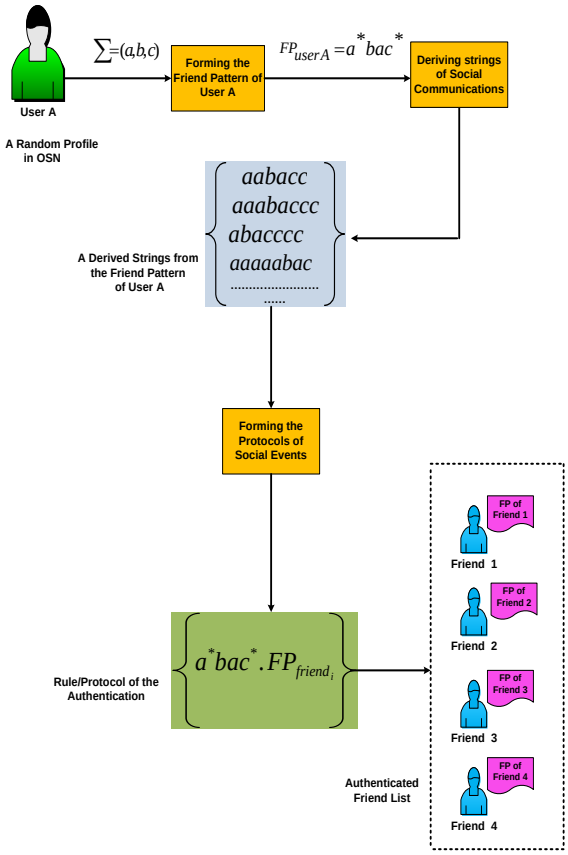

Fig. 5. a proposed methodology to authenticate user's friend list

Definition 8 (Detection Model of TSG-(DMT))

Is a Deterministic Finite Automaton. Such that, the DFA of a specific $profile_x$ in TSG is consisting of five Tuples:

$$DMT = (\mathbf{Q}, \sum, \delta, q_0, \mathbf{F})$$

Such that,

Q: is a non empty finite set of states, $\forall\ q \in Q$, q is called a state of Q

$\sum$: is an input alphabet, which used to define input symbols of *Friend Patterns* that used in authenticating social relations between OSN profiles. $\sum$ can be defined as;

$$\sum = \bullet \cup \sum_1 \cup \sum_2 \cup \sum_3 \ldots\ldots \cup \sum_n$$

Such that " ● " is the DOT concatenation operation and $\sum_i$ is the alphabet set of symbols by which the authenticated Friend Pattern of $friend_i$ is defined in the friend list of $profile_x$

δ: is a transition function that maps from state to state in **Q** over input symbol of the alphabet ∑ such that,

δ: Q×∑ →**Q** .

q_0**:** is the initial state of D FA.

F: is a finite set of final state of DFA, F⊆**Q,** $\forall\ q \in \mathbf{F}$, q is called the final state.

The Friend Pattern recognition process is depicted in Fig. 5. that explains how $profile_x$ can recognize the authenticated and trusted social events that come from other users in OSN. The recognition process requires first to represent each profile in OSN as a Regular Expression (i.e. Friend Pattern) as we showed previously. Then it requires to create a *Deterministic Finite Automaton* (i.e. $D(FP_{User\ x})$).to parse the social relations between $profile_x$ and his friend list (i.e. $D(FP_i)$). The social behavior between any two profiles in OSN as described in the proposed TSG Topology is represented as two concatenated strings in the form string $S = s_j . s_i$ such that s_j is a string derived from the Friend Pattern of $user_j$ (i.e. the Receiver), and s_i is a string derived from the Friend Pattern of $user_i$ (The Sender). if the current instance of social event is accepted by the DFA of $profile_x$, this mean the sender profile is genuine and authenticated profile in the friend list of the receiver profile (i.e. $profile_x$), otherwise, the DFA will apply the transition *OTHER* which mean that the current instance of social behavior is from a fake profile or from unauthenticated one. Hence these profile will rejected

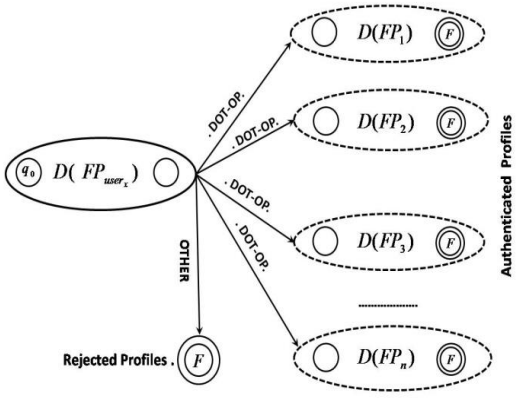

Fig. 6. A proposed Detection Model of TSG (DMT) model for recognizing Fake Profiles

VI. Conclusion and Future Work

One of the main malicious activities of OSN intruders today is their abilities for creating fake profiles which used as an effective way to perform cloning attacks across OSN platforms. From security and privacy point of

view, detecting these fake profiles represent a major and potential problem. In this paper we proposed a theoretical framework for detecting these fake profiles in a novel view. The proposed framework depends on two dimensions: the first dimension is the proposed social graph topology which called *Trusted Social Graph (TSG)*. We used TSG topology to visualize social relations between profiles across OSN platform in more controlled way. The second dimension is the detection methodology (i.e.DMT) that uses *a Deterministic Finite Automaton (DFA)* and *Regular Expression* techniques to parse and recognize the instances of social communications and social events that performed within OSN platform. The proposed detection model decides whether these social communications or social behavior from authenticated and trusted profiles or from fake and unauthenticated profiles. In the future work we will develop an algorithm to implement our detection framework and we will apply it on selected profiles as an experimental datasets from a specific Online Social Network.

REFERENCES

[1] M.Bosma, E.Meij, and W.Weerkamp, *" A Framework for Unsupervised Spam Detection in Social Networking Sites"* ECIR 2012: 34th European Conference on Information Retrival, PP.364-375.

[2] Y.Altshler, Y.Elovici, A,B.Cremers, N.Aharony,and A.Pentland, *"Security and Privacy in Social Networks"* Springer NewYork, 2013.

[3] G.Stringhini, C.Kruegel, and G.Vigna, *"Detecting Spammers on Social Networks"* in proceedings of the 26th Annual Computer Security Applications Conference, ACM 2010,PP. 1-9.

[4] L.Bilge, T.Strufe,D.Balzarotti, and E.Kirda, " *All Your Contacts Belong to us: Automated Identity Theft Attacks on Social Networks",* in proceedings of the 18th international conference on world wide web. ACM,2009, PP.551-560.

[5] B.Furth," Handbook of Social Network Technologies and Applications", Springer,NewYork,2010.

[6] M. Balduzzi,C. Platzer, T. Holz, E.Kirda, D. Balzorotti, and C. Kruegel. *"Abusing Social Networks for Automated User Profiling"* Research Report RR-10-233, EURECOM, 2012, http://www.iseclab.org/papers/socialabuse-TR.pdf.

[7] D.Irani, M.Balduzzi, D.Balzorotti, E.Kirda, and C.Pu, " *Reverse Social Engineering Attacks in Online Social Networks"* detection of intrusions and malware and vulnerability assessment, PP.55-74, 2011.

[8] M.Conti, R.Poovendran, and, M.Secchiero, *" Fake Book: Detecting Fake Profiles in Online Social Networks"* Advanced in Social Network Analysis and Mining (ASONAM), 2012,IEEE/ACM International Conference on Pages, 1071-1078.

[9] M.R.Khoyyambashi, F.S.Rizi, *"An Approach for Detecting Profile Cloning in Online Social Networks",* e-commerce in developing countries: with focus on e-security (ECD), 2013 7th international conference, and pp.1-12.

[10] G.Kontaxi**,** I.Polakis,S.Ioannidis, and E.Markatos, *" Detecting Social Network Profile Cloning"* in previous computing and communications workshops (PERCOM workshops), 2011, IEEE International Conference on IEEE, 2011, PP.295-300.

[11] R.Baden, N.Spring,and B.Bhattachorjee, *"Identifying Close Friends on the Internet,"* in Proc. Of workshop on hot Topics in networks (HotNets-VIII), 2009.R.Baden, N.Spring,and B.Bhattachorjee, *"Identifying Close Friends on the Internet,"* in Proc. Of workshop on hot Topics in networks (HotNets-VIII), 2009.

[12] M.Fire, G.Katz, and Y.Elovici, *" Strangers Intrusion Detection_Detecting Spammers and Fake Profiles in Social Networks Based on Topology Anomalies"*2012,http:/www.academia.edu/1518357/strangers_Intrusion_Detection_Spammers_and_Fake_Profiles_in_Social_Networks_Based_on_Topology_Anomalies.

[13] K.Lee, J.Caverlee, and S.Webb, "*Uncovering Social Spammers:Social HoneyPots+ Machine Learning"* in Proceeding of the 33rd International ACM SIGIR Conference on Research and Development Information Retrieval, ACM, 2010,PP. 435-442.

[14] L.Jin, H.Takabi, and J.B.D.Joshi, *"Towards Active Detection of Iddentity Clone Attacks on Online Social Networks"* Proceedings of the first ACM Conference on Data and Application Security and Privacy,NewYork USA,2011,PP.27-38.

[15] A.Singh, A,H.Toderici, k.Ross, and M.Stamp, "Social *Networking for Botnet Command and Control"*,MECS, I.J.Computer Network and Information Security,2013, 6,pp 11-17.

[16] J.Gross, and J.Yellen, *"Hand Book of Graph Theory"* CRC Press LLC 2004,PP.253-256.

Authors' Profiles

Ali M. Meligy. A professor of Computer Science, Dept. of Mathematics, Faculty of Science, Menoufyia University, Egypt. He worked as a Professor of Computer Science, faculty of Information Technology, Middle East University for Graduate Studies, Amman, Jordan from September 2006 to August 2009. In 2002, he was a visiting Research Professor, Institute of Computer Sconce, Humbldt University, Berlin, Germany. In 2009, he was a visiting Research Professor, LRZ Computer Center, Munich University, Germany. His research interests involve: Software Engineering, Parallel and Distributed Systems, Computer Networks, Information Security, Modeling Using Petri Nets, and Social Networks.

Hani M. Ibrahim. Lecturer of Computer Science, Dept. of Mathematics, Faculty of Science, Menoufyia University, Egypt. In 2008, He obtained his PHD degree in Computer Science, Dept. of Mathematics, Faculty of Science, Menoufyia University, Egypt. In 2004, he obtained his Master degree in Computer Science, Dept. of Mathematics, Faculty of Science, Menoufyia University, Egypt. His research interests include: Image Processing, Pattern Recognition, Biometric, Neural Networks, Artificial Intelligence, and Social Networks.

Mohamed F. Torky. Born in Egypt in August 1986. He is a PHD Candidate in Computer Science from Faculty of Science, Menoufyia University, Egypt. He obtained his Master degree in Computer Science, Dept. of Mathematics, Menoufyia University, Egypt, 2013. He works as an Assistant lecturer in Dept. of Information Technology, High Institute of Computers, Ezbet El Nakhl, Cairo, Egypt. His research interests include: Computers and Information Security, Cryptography and Social Networks Security and Privacy.

