Combining Trust Graphs and Keystroke Dynamics to Counter Fake Identities in Social Networks

Francesco Buccafurri, *Member, IEEE,* Gianluca Lax, *Member, IEEE,* Denis Migdal, Lorenzo Musarella and Christophe Rosenberger

Abstract—Fake identity in social networks is a phenomenon that is strongly increasing, and it is used for discovering personal information, identity theft, influencing people, spreading fake news, fraud, and so on. In this paper, we face this problem by introducing the concept of certified social profiles and by propagating this property through a collaborative approach that exploits keystroke-dynamic-recognition techniques to identify illegal access to certified profiles. We propose a decentralized approach to compute the trust level of a social profile, and we show the robustness of the proposal by analyzing the security of the trust mechanism through experimental validation.

Index Terms—Social Networks, Trust, Fake Profiles, Keystroke Dynamics

1 Introduction

In daily life, all communications are taking rapidly the direction of the digital and the virtual domain. In particular, social networks and social media platforms represent huge sources for information sharing where people can interact with each other easily and in a fast way. At the same time, online social networks are big catching areas for collecting and spreading trash news and fake information as well [19], [26]. Unfortunately, the awareness of users of the most common cyber threats is not growing as quickly as risks do. Indeed, the *online* community still has to be educated concerning cyber threats [2]. For this reason, it is important to improve the security of services, and trust represents a fundamental property that users must look for when they interact on social networks. Usually, fake news is shared and forwarded mostly by fake profiles.

Social network profiles whose claimed identity does not match the real user are certainly potential security threats on the Web. This happens in two cases. The first case is that of fake profiles, in which the attacker intentionally creates a clone of a real-life identity profile of the victim, pretending to be them in the interactions. The second case is that of compromised profiles, in which an intruder, permanently or temporarily, uses the real social profile of the victim fraudulently. In both cases, the risk of anomalous behavior with potential damage to the victim's reputation, espionage, or social engineering attacks toward people connected to the victim is very high.

- *F. Buccafurri, G. Lax, and Lorenzo Musarella are with the DIIES Dept., University Mediterranea of Reggio Calabria, Italy.*
  E-mail: bucca@unirc.it, lax@unirc.it, lorenzo.musarella@unirc.it
- *D. Migdal is with Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, LIMOS, France.*
  E-mail: denis.migdal@uca.fr
- *C. Rosenberger is with Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France.*
  E-mail: christophe.rosenberger@ensicaen.fr

The problem faced in this paper regards the fact that, through fake profiles, attackers can entice users to give up personal data, hijack them toward infected websites and, once their email addresses are known, launch spear-phishing campaigns. Several studies have been proposed in the literature to counter this problem [9], [13], [21], [30], mainly based on associating each social profile with a certain degree of trust. All the existing proposals require a strong analysis effort by the social network provider, which takes into account all the behavioral and topological information of the profiles.

Differently from these pioneer methods, we propose an approach based on a collaborative trust mechanism that may operate in a truly decentralized fashion, in which trust is combined with behavioral biometric mechanisms to counter profile compromising. The novelty of our proposal is that the computation of the trust level is decentralized: indeed, it exploits only user interactions and does not require any central authority for trust management or computation.

The underlying idea exploits the social structure of our domain. The trust model is based on a robust implementation of the *word-of-mouth* approach. Robustness is obtained by redundancy. In words, we follow the principle that if a sufficient number of people trust the identity of a social network profile, we can trust it as well. This way, we obtain a graph of trust, because we propagate trust under the basic assumption that a fake user (and then fake behavior) is transitively excluded. We base our assumption on the consideration that, when the real-life identity is known, sanctions are facilitated in case of misbehavior (e.g., victims could sue users who certified the perpetrator), thus misbehavior is prevented.

Trust is obtained through redundant trust chains in which any node plays a role similar to an intermediate certifier in a certification chain until a certified profile (i.e., a trust anchor) is reached. Our model requires the presence of some profiles certified by a Trusted Third Party. To identify possible intrusions in a legitimate profile, the trust model takes also into account the behavioral biometric traits of users that they record and verify in a peer-to-peer fashion. Importantly, no biometrics are stored by the social network provider. In other words, the word-of-mouth mechanism propagates the information that the current behavior of a given node is not compliant with that of the initial safe state, thus reducing the trust of the community towards that node. Keystroke dynamics is used as a behavioral biometric modality, which is very easy to collect on web pages (e.g., by using JavaScript code) and allows a simple and low-cost solution to verify the identity of a user [29], [38]. We can summarize the contribution of this paper as follows:

- We propose a theoretical framework based on a resilient trust graph model that enables the computation of a trust degree with respect to a social network participant in a decentralized way.
- Such framework is then implemented and several experiments are carried out with real datasets to validate the system performance and resilience, comparing such results with other proposals as well.
- The proposal exploits the behavioral biometric feature of the user, in terms of keystroke dynamics, to identify possible intrusion with respect to a legitimate profile and to measure its trustworthiness.
- We provide the security analysis, validating the robustness of our approach.

The structure of the paper is the following. In Section 2, we contextualize our proposal in the state of the art. Then, in Section 3 we provide a general overview of our approach, while in Section 4 we introduce the biometric features and the behavioral biometric modality used in our approach. Section 5 describes how our trust model works and it provides the theoretical support for the trust mechanism. In Section 6 we present a decentralized way to compute the trust level and in Section 7 we describe the system functions that a social network platform needs to implement our approach. In Section 8 we test our approach and show how security properties are fulfilled. Then, Section 9 includes details about datasets and implementation issues related to our approach. Finally, in Section 10 we draw our conclusions and discuss future work.

2 Related Work

In online social networks, the detection of fake profiles is becoming every day more and more important because there are many threats (such as scamming, trolling, phishing, Sybil attacks, and social bots) that need to be faced [33], [39].

In this scenario, it can be helpful to create a model that includes the dynamic computation of a trust degree for each user. Trust is becoming a fundamental element of a successful social network [36], and it derives from the "social capital", which is based on the density of interactions among people.

The concept of trust applied to the digital domain has been introduced through PGP (Pretty Good Privacy). The original intent of PGP was to provide a "cryptographic tool for the masses". Its main purpose was to encrypt e-mail messages using public or conventional key encryption. For this reason, PGP does not adopt the traditional hierarchical trust architecture but chooses the "web of trust" approach, in which the users sign each other's public keys. Thus, a network of public keys is progressively originated, with links formed by signatures. This way, there is no need for a central authority [1]. Although the first instance of PGP allowed determining the maximum length of the certification chain through the `CERT_DEPTH` parameter, researchers have not shown much interest in seeking new solutions for trust propagation because it can be quite difficult to use it properly in real applications [1], [20]. In particular, only in recent years, this aspect has been investigated by [7], in which authors propose a blockchain-based solution.

A survey of trust in social networks [36] introduces three categories of trust models: *(i) graph-based* models, which consider only how members are related to each other and do not consider the real interaction between them; *(ii) interaction-based* models, which consider only interaction in the community and ignore the social network structure; *(iii) hybrid* models, which try to consider both the aspects to compute the social trust.

One of the most crucial points of this research area is, surely, how trust propagates in the network. For this purpose, the small world theorem and the social network analysis techniques have been explored. In particular, the authors of [30] enhanced the *STrust* model by proposing an association-based trust propagation model that considers two types of interactions: active and passive. *STrust* model is computed as a combination of the popularity trust and the engagement trust as $STrust(u) = \alpha PopTrust(u) + (1 - \alpha) EngTrust$, which are defined in [30]. The paper [21] proposes an approach to select the most trusted path between two nodes of a social network by merging seven criteria, such as profile similarity, topological similarity, Dunbar's theorem, and other measures related to social network analysis. The authors specify that their approach allows them to consider longer trust propagation paths than others. A model based on the uniform trust propagation called SN-GDM (Social Network-based Group Decision Making) is presented in [43]. In particular, the authors propose the two concepts of *Trust Score (TS)* and *Knowledge Degree (KD)*, which are combined to define a social trust value that does not lose any trust information during its propagation. Their model requires the intervention of Trusted Third Parts (TTPs) for the validation of results.

A decentralized and privacy-preserving online social network is presented in [10]. Here, the authors propose *Safebook*, a system that provides registered users with data storage and data management functions relying on trust relationships that are part of social networks in real life.

Recently, the use of machine learning has been proposed for detecting fake profiles. In [42], a dynamic Convolutional Neural Network (CNN) is built for fake profile classification that provides better results in terms of accuracy and loss than other commonly used learning algorithms. Another machine-learning-based classifier to detect bots in online social networks is proposed in [37] to counter Sybil attacks.

Our approach is also related to the concept of information diffusion in online social networks, in the sense that the roots influence the trust values of other nodes in the network following the rules of information flow [32]. Most of these works study how information flows in online social networks and propose strategies to maximize this diffusion by identifying strategic nodes for information propagation. The aim of our paper is somehow orthogonal to these studies and may exploit these solutions to improve trust propagation through the social network.

Furthermore, our approach leverages biometric data, which is a practice not new for social network applications [8], even though there are few papers in this area. In particular, most of the papers focus on user authentication using biometric data to enhance its security. Some papers in the literature considered soft biometrics with possible applications to social networks [14]. Most of the works consider gender recognition by analyzing the type of images posted or the keystroke dynamics [15], [40]. To the best of our knowledge, no work considers keystroke dynamics as a solution to enhance trust in social networks.

Several works exploit biometrics to implement continuous authentication schemes [12], [28], [41]. In high-security environments, the typical session-level authentication can be exposed to session hijacking, in which an attacker targets a post-authenticated session. In those contexts, continuous and real-time verification of user identity may become mandatory, and a lot of research effort has been devoted to the use of biometrics as a means to achieve this objective.

However, the goal of these strategies is very far away from ours. In detail, our approach does not aim at proposing a strategy to continuously verify that an active login session is controlled by the right user. Instead, our approach exploits biometric data as feedback to our trust model to measure the trustworthiness of an online profile.

From the literature review described earlier, it appears that the objective pursued by our paper is new. Therefore, to the best of our knowledge, there is no approach that can be (analytically or experimentally) fully compared with our proposal. In Section 9.3.2, we provide a comparative evaluation between our method and that proposed in [30] from the perspective of resilience against slandering attacks.

The rough original idea at the basis of this paper has been initially presented in [4]. However, there is a significant difference between the two papers. In particular, being [4] the report of early-stage research, the theoretical framework was not complete, and the proof of theorems was not included. No distributed implementation of trust computation and certification infrastructure was provided, as well as the concept of *certificate* was not presented. Moreover, there was not any security analysis. In addition, the experimental validation of the proposal carried out in the old version is very preliminary, because it only considered one small synthetic dataset (i.e., 2,500 nodes) with very few experiments instead of five real, heterogeneous, and bigger datasets. We extend the experiments through a comparative analysis of our method with respect to the literature and by considering (experimentally) an attack to demonstrate the robustness of our approach. We improve the part of the paper including biometrics and, finally, we add the spatial and temporal complexity analysis of all the functionalities of our proposal.

3 Overview of the Approach

In this section, we provide a general overview of how our solution works.

First, we highlight that our proposal is applied to online social networks and that it works by employing trust chains built among users. In particular, each chain starts from a root node, which is a social profile certified by a Trusted Third Party (TTP). To build a root profile, a user has to register with the social network via TTP, by executing an identification process proving their real-life identity. This could be achieved by using a public digital identity system. In addition, in this phase, TTP gathers the biometric (behavioral) parameters of the user to create a model that will be exploited in future interactions with the user to adapt the trust level and verify whether the account is still under the user's control. Indeed, in the negative case, the certificate associated with that profile will be revoked.

At the initial state of our protocol, the list of certified profiles coincides with the set of roots. At this point, new profiles could be certified by root nodes via certificate generation. When a new profile reaches a given *trust level*, it will be considered certified and it will play an active role in trust propagation.

We model the social network as a directed graph $G = \langle N, E \rangle$, where $N = N_c \bigcup N_{nc}$ is the union between the set of certified profiles N_c and the set of non-certified profiles N_{nc}, and E is the set of edges representing the friendship among these peers. We use the notion of directed graphs because they handle the case of symmetric friendship (as happens in Facebook) simply by including two edges in both directions. For example, if we want to represent a Facebook friendship between i and j, then we set the edges $E_{i,j}$ and $E_{j,i}$.

Any node of the social network (both certified and non-certified) may directly recognize some of its direct contacts. The basic idea is that a node recognizes only those nodes for which past real-life interactions occurred, allowing the node to conclude, also by using external knowledge, that the claimed identity is real (this situation typically happens for a significant portion of social network contacts). When a safe interaction happens (for instance, at the first message exchange allowing the recognition of the interlocutor) the profile acts as a recognizer and builds a biometric model of the recognizing node. This way, a subsequent intrusion can be detected. Furthermore, we remark that only a node already recognized can play the role of the recognizer. The underlying rationale is that the misbehavior of a user is directly connected to their anonymity in the social network. In other words, by making the recognizing process fully accounted and traced (and related to a real-life identity), we can increase the trust in recognized identities, provided that transitively, the process leads to root nodes.

Since we cannot give an absolute value to the above principle, we have to increase the level of trust by requiring redundancy in the recognizing process, thus making more improbable the conjunct misbehavior of identified recognizers. The level of redundancy sets the level of trust. The biometric model built by any participant, allows us to detect possible profile compromising, thus including in the trust also the expectation that an initially identified profile is still under the exclusive control of the legitimate owner. It is worth noting that, in principle, the biometric model could be learned by exploiting multiple channels (social network interactions, chats, shared editing, and so on).

We remark that the proposed approach is not aimed at defining a digital identity system, since only a level of trust is obtained. In fact, when a user A recognizes a user B, they are stating that B is not claiming a false identity, not the veracity of all published information. The quantity and quality of information needed by A to reach this conclusion depends on the social context. Besides name and surname, they may regard other information, such as age, job, and friendships.

Furthermore, we highlight that this approach does not recognize directly untrusted social profiles, but it aims to provide a certain degree of positive trust that is tuned and refined considering all the features, including biometrics, which we will explain in the following so that we can consider a social profile as certified. Clearly, if a certain social profile is not certified because it has a low trust degree or has no degree computed at all, it would represent an implicit suggestion to the user that such a social profile could convey dangers.

4 Biometric Features

In this paper, we propose to use biometric features to guarantee the previous security requirements. We focus in this work on a behavioral biometric modality as it can be easily collected in a transparent way [45]. Among existing solutions, we can cite mouse dynamics [27], touchscreen interactions [44], or keystroke dynamics [3]. The last technique is a behavioral biometric modality consisting of analyzing the user's way of typing on a keyboard. This biometric information can be computed easily on the Internet using simple JavaScript code. Keystroke dynamics is a biometric modality identified more than 40 years ago. We can cite a pioneer work in 1980 with a study where seven secretaries were asked to type three different texts [16]. The results were promising but lacked a sufficient number of users involved in the database. Most research works in keystroke dynamics assume a quasi-keyboard invariance or intend to propose algorithms dealing with slight behavioral modifications related to the keyboard type. In a previous study [18], experimental results from data collected from 133 users when using 2 types of keyboard (laptop and desktop) showed similar performance considering the keyboard used for enrollment or verification steps. However, some papers showed some performance differences when different keyboards were used during enrollment and verification. We can mention the following study [34] with 2 used keyboards (laptop and desktop) even if collected data are only from 17 users. A more recent paper [24] involving data from 4 keyboards and 86 users can also be mentioned. We assume in this work, as many in the literature, that the keyboard has a low impact on feature extraction. The use of mobile devices is not considered in this paper but many methods exist to deal with this type of capture [11].

The capture process of keystroke dynamics is shown in Figure 1. It consists in computing several features when the keys are pressed and released (timestamp of the event, code of the key, ...) provided by any Operating System (OS). The feature extraction consists mainly in measuring different latency and duration time between each key. Figure 1 shows an example where the user presses two keys on the keyboard. The user presses "C" at T1, "O" at T2, and releases "C" at T3 and "O" at T4. Note that the following relation is always respected: $T3 > T1$ and $T4 > T2$ (we always release a key after pressing it), while the following condition may not always be respected: $T2 > T3$ (because, as in our example, a user may press another key before releasing the previous one). We can extract three latency values (T2-T1, T4-T3, T2-T3) which we call PP (latency between two pressures), RR (latency between two releases), RP (latency between one release and one pressure) respectively, and one type of duration (T3-T1 or T4-T2) which we call PR (duration of a key press). The described process is repeated for all the keys.

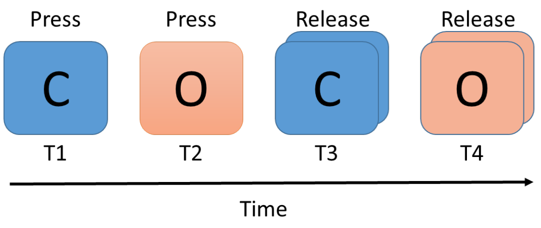

Fig. 1. Information captured in a keystroke dynamics system when pressing C and O keys [17].

Keystroke dynamics can be used either with passwords to enhance the security of user authentication or on free text. For example, the keystroke dynamics of a user could be analyzed while exchanging with another (e.g., through a social network chat). Subsequently, we consider the different timing information between two-character sequences known as *digraphs*. Digraphs are the latency times between two successive keystrokes. The biometric template associated with user z is composed of n digraphs $T_z = \{T_z^1, .., T_z^n\}$. The considered digraphs could be associated with one language. In this work (see Section 8.3), we considered 14 different values (common ones in English). If we consider a higher value of n, a few sentences will allow us to compute the keystroke dynamics feature vector. Another solution is to compute the distribution of latency values from the text typed by a user as a behavioral signature.

First, we need to generate the reference biometric template for each user by analyzing their keystroke dynamics during a period of time when we assume only the legitimate user interacts with the social network. To guarantee user privacy, we apply a biometric template protection scheme on digraph data (see Algorithm 1). The BioHashing algorithm [25] is applied to biometric templates that are represented by real-valued vectors of fixed length (so the metric used to evaluate the similarity between two biometric features is the Euclidean distance). It generates binary templates of length lower than or equal to the original length (here, the metric D_T used to evaluate the similarity between two transformed templates is the Hamming distance). A complete review of cancelable biometric systems has been reported by [31].

The reference template of user z is defined by $\widetilde{B}_z = \{E[B_z], \sigma[B_z]\}$ where $E[.]$ corresponds to the average value of biometric templates of user z and $\sigma[.]$ the associated

Algorithm 1 BioHashing

```
1: **Inputs**
2: $T = (T_1, \dots, T_n)$: biometric template,
3: $K_z$: secret seed
4: **Output** $B = (B_1, \dots, B_m)$: BioCode
5: Generation with the seed $K_z$ of $m$ pseudorandom vectors $V_1, \dots, V_m$ of length $n$,
6: Orthogonalize vectors with the Gram-Schmidt algorithm,
7: **for** $i = 1, \dots, m$ **do** compute $x_i =< T, V_i >$.
8: Compute BioCode:
       $B_i = \begin{cases} 0 & \text{if } x_i < \tau \\ 1 & \text{if } x_i \geq \tau \end{cases}$
   where $\tau$ is a given threshold, generally equal to 0.
```

standard deviation. To decide if a biometric template B_x of size n belongs to user z, we need to compare it with the reference template of user z denoted $\tilde{B_z}$ as follows [22]:

$$Score = 1 - \frac{1}{n}\sum_{i=1}^{n} e^{-\frac{|B_x - E[B_z]|}{\sigma[B_z]}} \quad (1)$$

This score gives a confidence measure about how much the user z is legitimate and will be used in the trust model proposed in the next section.

As we will see in the rest of the paper, the biometric system is used in such a way that the following security properties are guaranteed:

- *Confidentiality:* the biometric data are not disclosed to others.
- *Non-reversibility:* the biometric template is not enabled to retrieve the biometric data.
- *Discriminant:* the biometrics discriminate users.
- *Constant:* the biometrics remain constant.
- *Non-usurpation:* a user is not able to forge/imitate the biometric of another user.
- *Not-costly:* in terms of memory/time/ergonomics/money (no additional devices).

5 The Trust Graph

In this section, we give the theoretical framework underlying our trust model works.

Throughout this section, consider given a directed graph $G = \langle N, E \rangle$ representing a social network and a *redundancy* parameter t, i.e., a positive integer representing a level of trust. Let TTP be a Trusted Third Party. Let denote by N_c the set of *certified nodes*, i.e., the nodes whose identity is assured and monitored by TTP. Given a node $u \in N$, we denote by $\Gamma(u)$ the set of neighbors of u (i.e., adjacent nodes). Moreover, we denote by $R(u) \subseteq \Gamma(u)$ the set of nodes recognized by u.

Our trust model is based on the notion of recognition done by a certain user (in this formal framework just a node) towards other directly connected users. However, there is a recursive requirement. If we establish that t is the minimum number of recognitions that a user should receive to be considered trusted, we require that such recognitions, to be valid, must be done by trusted users (i.e., either users who received at least t recognition or certified users, which play the role of roots). This is formally encoded in the next definition:

Definition 1. We say that a node $u \in N$ is *t-recognized* (*in $A \subseteq N$*) if either: (i) $u \in N_c$ (i.e., is a certified node), or (ii) there exist t other t-recognized nodes in A that recognize u.

When the set A of the definition above is not specified, we intend that a node is t-recognized in N. From the above definition, it immediately follows that nodes in N_c are t-recognized for any t and in any set A.

We want now to understand how to characterize (and then compute) the set of users who, thanks to the propagation mechanism enforced by the above recursive definition, are trusted (with level t), when a set of roots N_c is fixed.

To do this, we first need to introduce the notion of *t-closed* set. Informally, a *t-closed* set is a set of set of t-recognized nodes that is closed with respect to the trust propagation mechanism. More formally:

Definition 2. A set $A \subseteq N$ of t-recognized nodes in A is said *t-closed*, if there is no $u \in N \setminus A$ that is t-recognized in A too.

From the above definition, it immediately follows that all certified nodes must belong to any t-closed set.

It is easy to see that the operator $\subseteq$ induces a partial order over the set of t-closed sets, which is a lower semi-lattice. We denote by $N^t \subseteq 2^N$ the set of non-empty t-closed subsets of N. N^t is then a lower semi-lattice according to $\subseteq$. We denote by N^t_b the bottom of N^t.

In our model, the role of N^t_b is central, because it includes exactly all nodes that are t-recognized, but, due to subset minimality, they do not form clusters whose recognizing is only mutual. In other words, N^t_b is the set of nodes for which trust paths start from certified nodes. Therefore, N^t_b represents the set of trust nodes, as more formally stated in the next definition.

Definition 3. Given a node $u \in N$ we say that u is *t-trusted (in N)* if $u \in N^t_b$. N^t_b is also said the *set of t-trusted nodes (in N)*.

It is now important to understand how to compute N^t_b. To do this, we provide an operational definition of N^t_b, based on the fixpoint of a monotone operator Λ_t, called *t-recognizing operator*. From this definition, it easily arises that the trust of nodes in N^t_b can be directly or indirectly linked to (at least) t certified nodes.

Definition 4. Given $t > 0$, let denote by $2^N_t \subseteq 2^N$ the set of subsets $A \subseteq N$ such that, for each $x \in A$, x is t-recognized in A. We define the *t-recognizing operator* $\Lambda_t : 2^N_t \rightarrow 2^N_t$ as follows: (i) $\Lambda_t(\emptyset) = N_c$ (ii) $\Lambda_t(A) = \{u \in N \mid \exists B \subseteq A \; s.\,t.\; |B| \geq t \wedge u \in \bigcap_{v \in B} R(v)\}$. It is easy to see that if A is in 2^N_t, then $\Lambda_t(A)$ is in 2^N_t too.

Now, we define the following sequence of sets: $\Lambda^0_t = \Lambda_t(\emptyset)$; $\Lambda^k_t = \Lambda_t(\Lambda^{k-1}_t)$, for any $k > 0$.

Lemma 1. *The operator Λ_t is monotone.*

Proof. To prove the claim of the Lemma, we have to show that given two subsets A and B of 2^N such that $A \subseteq B$, it holds that $\Lambda_t(A) \subseteq \Lambda_t(B)$, for any $t > 0$. We proceed by contradiction by assuming that there exists $x \in \Lambda_t(A)$ such that $x \notin \Lambda_t(B)$. This means that x is not t-recognized in B. But this is impossible because x is t-recognized in A (since $A \in 2_t^N$) and $A \subseteq B$. □

Lemma 2. *A set of nodes $A \in 2_t^N$ is t-closed if and only if $\Lambda_t(A) \subseteq A$.*

Proof. (*only-if-part*). Let x be belonging to $\Lambda_t(A)$. Then, there exist at least t nodes in A that recognize x. Therefore, if A is t-closed (for hypothesis), then $x \in A$ too.
(*if-part*). Let $x \in \Lambda_t(A)$. Then, there exist at least t nodes in A that recognize x. Since $\Lambda_t(A) \subseteq A$, $x \in A$ too. Then A is t-closed. □

We are ready to state the following result, which gives us an operational way to compute N_b^t.

Theorem 1. *The infinite application of the operator Λ_t, denoted by Λ_t^∞, is a fixpoint (i.e., $\Lambda^t(\Lambda_t^\infty) = \Lambda_t^\infty$), it is the least fixpoint, and this corresponds to the set of t-trusted nodes N_b^t.*

Proof. The proof of the first part of the theorem (i.e., the existence of the least fixpoint) directly derives from Lemma 1 and the theory on fixpoints (i.e., Tarski's theory).

Now, we have to prove that $\Lambda_t^\infty = N_b^t$. First, we prove that $N_b^t \subseteq \Lambda_t^\infty$. By Lemma 2, as N_b^t is t-closed, $\Lambda_t(N_b^t) \subseteq N_b^t$. If $\Lambda_t(N_b^t) = N_b^t$, then it is a fixpoint and then it is the least fixpoint Λ_t^∞. Thus, $N_b^t = \Lambda_t^\infty$, and the claim is proven. If $\Lambda_t(N_b^t) \subset N_b^t$, then, due to Lemma 1, $\Lambda_t(\Lambda_t(N_b^t)) \subseteq \Lambda_t(N_b^t)$. By Lemma 2, this implies that $\Lambda_t(N_b^t)$ is t-closed. But this is not possible because $\Lambda_t(N_b^t) \subset N_b^t$ and considering that N_b^t is the bottom of the lattice. This part of the proof is then completed.

Now we have to prove that $N_b^t \subseteq \Lambda_t^\infty$. Since Λ_t^∞ is a fixpoint, then $\Lambda_t(\Lambda_t^\infty) = \Lambda_t^\infty$. Therefore, by Lemma 2, Λ_t^∞ is t-closed and this implies that $N_b^t \subseteq \Lambda_t^\infty$. The proof is then concluded. □

The above notion of t-trustworthiness embeds a lossless propagation of trust in which the level of assurance of identity based on the recognition of users does not degrade if the t redundancy property holds at every step of propagation. In other words, the t-redundancy property is considered a threshold to propagate the trust. The t-redundancy parameter implicitly represents the assumption that the multiple identifications of a node u done by nodes in turn identified with the same trust level, and so on, until t certified nodes are reached, can be considered sufficient to trust the identity of u. The approach applies the concept of trust chain used in the context of digital certification to the domain of identity management in social networks, with the aim to counter the problem of fake identities. It is worth remarking that the model cannot provide absolute guarantees but only a trust level directly connected with the value t. The higher t, the higher the trust in identities.

So far, the trust model assumes that once a user has recognized another user, no revision of this information must be done. This assumption would be valid only in the absence of attacks that give the attacker access to the user profile (even temporarily). So we assume a sort of *safe state* regarding fraudulent accesses. In other words, the trust model above prevents the risk of fake profiles and fake identities but not from fraudulent access to legitimate profiles.

To counter this further case, we introduce a biometric-based reinforcement and combine it with the above trust-chain mechanism, to decrease the trust in a given subject if the biometric trait is not recognizable. Therefore, we can also manage non-safe states. The full trust in our mechanism is obtained by relying on the assumption that the disclosure of trusted real-life identities prevents the misbehavior of users in the trust mechanism itself, under the t-redundancy assumption.

But, if the operating user is not the legitimate one, then the above assumption fails. Thus, the identity of those users whose trust is based on paths involving the attacked profile should be not fully trusted. In other words, to take into account this aspect, we have to enable a gradual level of trust, from 0 to 1 (while before the trust was basically either 0 or 1), and use an ϵ-approximation approach to trust identities. The first step is to modify the notion of t-recognized. Obviously, we keep the redundancy parameter t in the new definition, but we introduce the possibility that a user is not fully identified in a given moment, due to the fact that the biometric support is giving a warning rate. We require that nodes in N_c (i.e., certified nodes) lose their state if the biometric support gives a warning rate. Thus, we can assume that certified nodes are not attacked.

Given a node u, we define the set $R^\epsilon(u)$ (where $0 \leq \epsilon \leq 1$) as the set of pairs $\langle v, b_r(v)\rangle$ such that $v \in R(u)$ (i.e., v is a node recognized by u in the safe state) and $1-\epsilon \leq b_r(v) \leq 1$ is the current biometric rate (i.e., the score computed earlier normalized from 0 to 1), provided that it is higher than a given threshold $0 < 1-\epsilon \leq 1$ under which v must be currently considered not recognized. Obviously, when $\epsilon = 0$ we fall in the safe state. We say that nodes in $R^\epsilon(u)$ are ϵ-*recognized* by u. At this point, we are ready to extend the notion of t-recognized to a non-safe state.

Definition 5. *We say that a node $u \in N$ is $\langle t, \epsilon, r\rangle$-*recognized *(in N) if either: (1) $u \in N_c$ (i.e., is a certified node), or (2) there exists a set B of $\langle t, \epsilon, r\rangle$-recognized nodes such that both (i) $u \notin B$, (ii) $|B| \geq t$, (iii) $u \in R^\epsilon(v)$, for each $v \in B$, (iv) $1-\epsilon \leq r \leq 1$, and (iv) $\frac{\sum_{v \in R} b_r(v)}{|B|} \geq r$.*

It is easy to see that a node is t-recognized, according to Definition 1, if and only if it is $\langle t, 0, 1\rangle$-recognized. The intended meaning of Definition 5 is to take into account warnings triggered by the biometric support (through the parameter ϵ), and, at the same time, to require by means of the parameter r that a possible fault of trust introduced by ϵ can be partially recovered by fortifying redundancy to reduce approximation. In words, if we can trust fewer nodes because we are not sure they are not attacked we need a larger set of witnesses to reach a safe conclusion anyway. This means that r modulates the level of assurance of trust, so that the higher r, the higher the trust in the identity of $\langle t, \epsilon, r\rangle$-recognized nodes. Actually, to talk about trust we have to avoid a mutual self-sustained cluster of $\langle t, \epsilon, r\rangle$-recognized nodes, so we have to proceed as in the safe state above by requiring the minimality condition. For brevity, we do not give all detail, but it is rather clear that definitions of N_b^t and, consequently, of t-trustworthiness of a node, can be easily extended to the non-safe case, on the basis of Definition 5. We reach thus the definition of $N_b^{\langle t,\epsilon,r\rangle}$ as the bottom of the semi-lattice of subsets of $\langle t,\epsilon,r\rangle$-closed nodes of N. Therefore, a node is $\langle t,\epsilon,r\rangle$-trusted if belongs to the set $N_b^{\langle t,\epsilon,r\rangle}$. Also the definition of *recognizing operator* can be trivially extended so obtaining the operator $\Lambda_{\langle t,\epsilon,r\rangle}$ in such a way that $N_b^{\langle t,\epsilon,r\rangle}$ coincides with the fixpoint $\Lambda^{\infty}_{\langle t,\epsilon,r\rangle}$ of such operator.

6 Decentralized Trust Computation

We present here a decentralized way to compute the trust level that does not require any central authority for trust management. Obviously, standard TTPs, such as certification authorities are required to certify root profiles, but they are not involved in the propagation trust mechanism. First, we assume the collaborative behavior of participants in the network. This assumption is reasonable because every participant can benefit from the trust system so constructed.

We propose a distributed algorithm for trust computation, in which each node sends, upon query, a proof that it is t-recognized, where t is the chosen trust level. In the scope of this paper, we choose the proof to be a certificate graph, a generalization of certificate chains, in which each node is certified t times.

We assume that each node owns its pair of asymmetric cryptographic keys and that the private key is kept secret. The public key is used also as the user's identifier.

Now, we define the (recursive) structure of a certificate. We have two kinds of certificates:

- *intermediate certificates*: containing the certified node public key and the certifier public key;
- *final certificates*: containing explicitly the certified node identity (e.g., URL) and public key, and the certifier's public key.

These certificates prove that the certifier node trusts the certified node (we highlight that only the certifier node is able to issue such certificates because they are signed). Intermediate certificates enable the certified node to certify other nodes without revealing its real identity in the certificate graph, and the final certificates enable the certified node to prove that it is t-confirmed by sending a certificate graph and its final certificates.

Specifically, a user's certificate is composed of:

1) *issuer*: this field contains the public key of the generator of the certificate;
2) *target*: this is the public key of the user to be certified;
3) *profile*: this field is obtained as $url \oplus r$, where url is the URL of the social network profile of the target, r is a randomly generated bit string, and $\oplus$ denotes the exclusive OR operator;
4) *key*, an optional field containing r (i.e., the value generated above). If this attribute is missing, then this certification is said *intermediate certificate*; otherwise, it is said *final certificate*;
5) *certifier*: this field may contain a (possibly empty) set of intermediate certificates of the users who previously have certified the issuer. If this field is empty, then the issuer should be a root node certifier and this target user is said *root node*.
6) *expiration date*: this field may contain the date until the certificate can be considered valid.
7) *value*: this field contains a value between 0 and 1; it is 0 when the certificate is revoked (for any reason), it is equal to 1 when the trust in the certificate is maximum, and a value between o and 1 in all the other cases.

Furthermore, the certificate is signed by the issuer's private key to guarantee information integrity.

In practice, when a user u wants to be t-recognized, u asks their social friends $u_1,\ldots,u_t$ for a final certificate. In this example, we assume that $u_1,\ldots,u_t$ are already t-recognized.

Now, each friend u_i of u signs and issues a certificate c_i with all fields filled-in (i.e., containing also the random r). At the end of this step, u has got t final certificates: if all of them are valid, then u becomes t-recognized as well. At this point, u can propagate the trust to another friend v, by issuing and signing a new certificate of v, in which the field *certifier* is composed of the certificates c_i^* with $1 \leq i \leq t$, where c_i^* is obtained from c_i by erasing the field *key*.

Concerning the validation of certificates, we have two cases.

An intermediate certificate is valid if:

- it is signed by the issuer's private key (i.e., the signature is valid);
- the issuer is a root node certifier (if the field certifier is empty); the field certifier contains at least t (recursively) valid certificates (otherwise);

A final certificate is valid if, in addition to the above conditions, it holds also that the value of the field profile $\oplus$ the value of the field key is equal to the URL of the social-network profile of the user being certified.

As introduced by Definition 5, our model takes into account also the possibility that the level of trust of a specific user can be gradually reduced if the biometric recognition is not full. Therefore, it may happen that an already t-recognized user's account becomes potentially compromised. In this case, the associated level of trust decreases. Furthermore, since our model is based on the concept of collaboration and propagation, when a certified user is compromised also all the paths involving the attacked profile should take into consideration this information and decrease the overall level of trust. According to Definition 5, the trust level t does not act just as an *on-off* switch, but it can assume values between $[0,1]$, so that if one node has been compromised, then the overall computation of each trust level in which that node is involved dynamically changes.

In particular, if we set $t = 10$ to be considered t-recognized and we consider an ideal world in which every peer has a $t\_level = 1$, we can state that if ten people sign a certificate directed to a given user, then it would be sufficient for this user to become t-recognized. Otherwise, in a real-world scenario, it would happen that more people would be needed to reach the same trust level because some participants could have a $t\_level \leq 1$.

Optimization.

Our method may suffer in terms of storage size because duplicate certificates are saved in the chain of certificates. For this reason, we provide an optimized version of our approach in which the certificate graph does not contain duplicate certificates.

Observe that when a node receives a (trust) certificate, the corresponding intermediate certificate is added to the set of intermediate certificates and the final certificate to the set of final certificates. The other certificates included in the received certificate graph are added to a third internal structure, consisting of a mapping in which each element key is the pair (*certified public key, certifier public key*), thus ensuring the uniqueness of each certificate. When issuing a new certificate, the node sends the status of the certificate along with the certificate itself.

The resulting certificate graph is represented by a multimap structure whose index is the certified public key. For each index, the number of certificates and the certificates are verified. Then, starting from the final certificates, the absence of loops is ensured by a depth-first search. If a visited node already belongs to the current path, then a loop is detected.

7 System Functions

In this section, we provide the basic functions of our trust model, which would be the basis for any social network platform implementing our approach.

- *Root registration*: this function is used by a node to become a *root node*.
- *Set Biometric Model*: it is invoked whenever two users establish a contact in the social network in the set-up phase. It is also invoked by the Root registration function to allow the social network provider to keep the biometric model of the root node.
- *Trust*: this function is invoked whenever a node wants to sign a certificate to another participant of the network;
- *Verify trust*: this operation is invoked whenever a node wants to verify that a given participant belongs to the set of certified nodes;
- *Revoke*: it is invoked whenever a participant wants to revoke a past certificate previously given to a certain node. This can happen for any reason.

We describe such functions and their complexity over time and space w.r.t. the number of nodes *n* and the trust level *t*.

Root Registration.

This operation involves a Trusted Third Party (TTP) and a social network profile that claims to become a root node. In particular, the TTP receives the request from a social profile and it starts an authentication process to demonstrate the association between the real-life identity and the social network profile. This process could use a public digital identity system. After being authenticated, the social profile becomes a root node, representing the first node of a new trust chain.

The complexity of this operation is constant over time and space.

Set Biometric Model.

This function is invoked every time it is necessary to model the biometric behavior of a social profile. In particular, the process of capturing keystroke dynamics consists of extracting different features, such as the time interval between pressing and releasing two characters on the keyboard, the code of the key, etc.

The complexity of this operation is constant over time and space.

Trust.

This operation can be carried out after a real-life interaction between the two users and it can be invoked only by those peers that have already been t-recognized and/or by root nodes. Such an operation is used to trust another profile by means of a signed certificate. In detail, it accepts a certificate as input and it can be invoked by every node belonging to the set of certified profiles N_c. In particular, two different certificates are signed: an *intermediate* certificate, which does not contain any identity-related information, and a *final* certificate, which contains also the certified user's identity.

The complexity of this operation is constant over time and space.

Verify trust.

It is called whenever a node wants to check the value of trust associated with a given social profile. In particular, the function accepts a social profile and a required trust level t and returns whether the social profile can be considered trusted or not.

To carry on the operation, the social profile being verified has to reveal its final certificates so that the node can check the validity of the certificate chain.

In Figure 2, we show an architecture in which node A becomes t-recognized by nodes C and B, with $t = 2$. Note that intermediate certificates do not include the identity-related information about the destination of such certificates, while final certificates contain the URI of the destination node because intermediates are used by a certified node to give trust in other peers, while finals are used in this verification process, where it is necessary to reveal the link between the certificate and the corresponding social identity. We can conclude that this phase succeeds when the sum of values included in final certificates is greater than or equal to the trust level t set by the user.

As for the spatial complexity, we note that the size of the certificate chain associated with each t-recognized social profile is $\sum_{i=1}^{d} t^i$, where d is the hop number between the social profile taken into consideration and the root nodes. Considering the example in Figure 2, d is equal to 2 because the social profile A to be verified is linked to the three root nodes D, E, F via the following paths: $A \rightarrow C \rightarrow F$, $A \rightarrow C \rightarrow D$, $A \rightarrow B \rightarrow D$, and $A \rightarrow C \rightarrow E$. Thus, the spatial complexity can be approximated to t^d, which is polynomial thanks to the small-world phenomenon and the six-degree separation law [5]. In particular, in social networks, the average degree of separation between two peers is found to be 3.43 [6]. Following the same reasoning seen for the spatial complexity, we obtain that the time complexity is t^d.

Revoke.

This task represents an important function of our solution. As is the case in most certification infrastructures, a certificate-revocation-list approach is used. The list of revoked certificates is maintained by the same TTP introduced in the root selection phase. This method is invoked every time we require that a certificate previously signed that is still valid (the scheduled expiration date has not come yet) is no longer considered trusted. Each entry of the certificate revocation list is associated with the identifier of the certificate and a timestamp. The node that invokes this function has to sign the certificate revocation to prevent malicious requests. Following the same idea of trust propagation, every time a certificate is found to be revoked, child branches of such certificate are no longer considered valid. Our model provides for two different ways: (1) an *explicit* revocation, for which any standard solution (e.g., the TLS approach) can be adopted to revoke the certificate before the expiration date, and (2) an *implicit* solution, denoted by setting the value associated with the certificate equal to zero.

Both time and spatial complexities are constant since this function does not involve either t or n.

As a final observation of this section, we highlight that our tree-like certification process might apparently represent an infeasible source of exponential growth. For example, whenever a certificate is revoked, all branches that are generated from that certificate are cut off and considered revoked as well. However, the real-life structure of online social networks (i.e., the small diameter and path lengths due to the small-world principle) makes this risk not effective.

8 Security Analysis

In this section, we discuss the security requirements presented earlier and explain how we provide these security properties and how our model prevents reputation attacks. We start by defining the threat model adopted in our approach. We focus our attention on the robustness of the trust mechanism, by assuming that the biometric system is secure.

The adversary is any user of the social network. We allow collusion among users, by assuming that the maximum number of colluding adversaries is less than the fixed trust level t.

Specifically, the adversary can perform the following attacks against the trust mechanism:

- *Sybil attack:* The adversary tries to generate multiple fake identities to jeopardize the system.
- *Slandering:* The adversary tries to fraudulently reduce the trust of a victim user.
- *Self Promotion:* The adversary tries to fraudulently increase their trust.
- *Whitewashing:* The adversary tries to fraudulently erase the reduction of trust.
- *Social engineering:* The adversary tries to compromise the trust mechanism by using social engineering attacks against other users.

Collusion and Sybil attacks are strongly mitigated because, as highlighted in the theoretical model, every participant must be certified to propagate trust and every certification chain must lead back to a root node. *Slandering attacks* are prevented, since it is impossible to defame someone because our model provides, by construction, only trust certificates and there is no way to vilify another participant. Furthermore, the solution we propose makes it impossible for a peer to *self-promote* because there is the need for the collaboration of the community to reach a certain trust level. The resilience of our approach against this type of attack is studied in Section 9.3.2.

Whitewashing attacks are mitigated because if a participant creates a new account to start with a new reputation, they will lose their certificates and it is necessary to execute the whole process of our algorithm to reach a certain trust level.

Our approach is robust against *social engineering attacks* because of the reinforcement given by adding biometric features to the model, which enable continuous authentication for every participant. Indeed, even if an attacker gains full access to the victim's social profile, our approach will discover the attack thanks to the continuous authentication provided by the biometric features.

Apart from the resistance to the previous attacks, we highlight that our solution achieves the following security properties.

Integrity is ensured because the attributes of certificates are signed by the issuer's private key. For the same reason, *Accountability* is provided as well. *Availability* is reached because certificates are computed once and then they are stored so that the computation of the t-recognized nodes is possible also in the case in which nodes are for any reason unavailable.

Anonymity is achieved in pseudonymous form because our model uses intermediate certificates to propagate trust. In these certificates, the real identity of the actors involved in the process is not revealed. The list of revoked certificates is stored and maintained by the TTP which is in charge also to carrying on the root registration step.

9 Experiments

In this section, we validate our proposal. First, we present the dataset used and discuss how biometrics is managed. Then, we analyze the results of the experiments we carried out.

9.1 Datasets

For the validation of the proposal, we used five real-life datasets downloaded from the repository `http://networkrepository.com/` [35]. Each dataset provides a list of edges between two nodes, which represent that the two nodes are friends.

In the Facebook section of this repository, several datasets are provided, each sampling a certain portion of the network, with different characteristics and properties. Among them, we selected five different data sources so that we can cover heterogeneous and different contexts.

In Table 1, we report the characteristics of our datasets, where $|N|$ is the number of nodes, $|E|$ is the number of edges, d_{max} is the maximum degree of a node in the given dataset, and d_{avg} is the average degree of the nodes. By analyzing such values, we can say that $D1$ and $D2$ are the most connected datasets because they have high degrees and they have many edges. $D3$ and $D4$ have the highest number of nodes but fewer edges and lower degrees, meaning that they are less connected and more scattered. Instead, $D5$ is the trade-off.

As for the biometric part, we used a biometric benchmark database composed of keystroke dynamics [23]. It is composed of a biometric template from 110 individuals

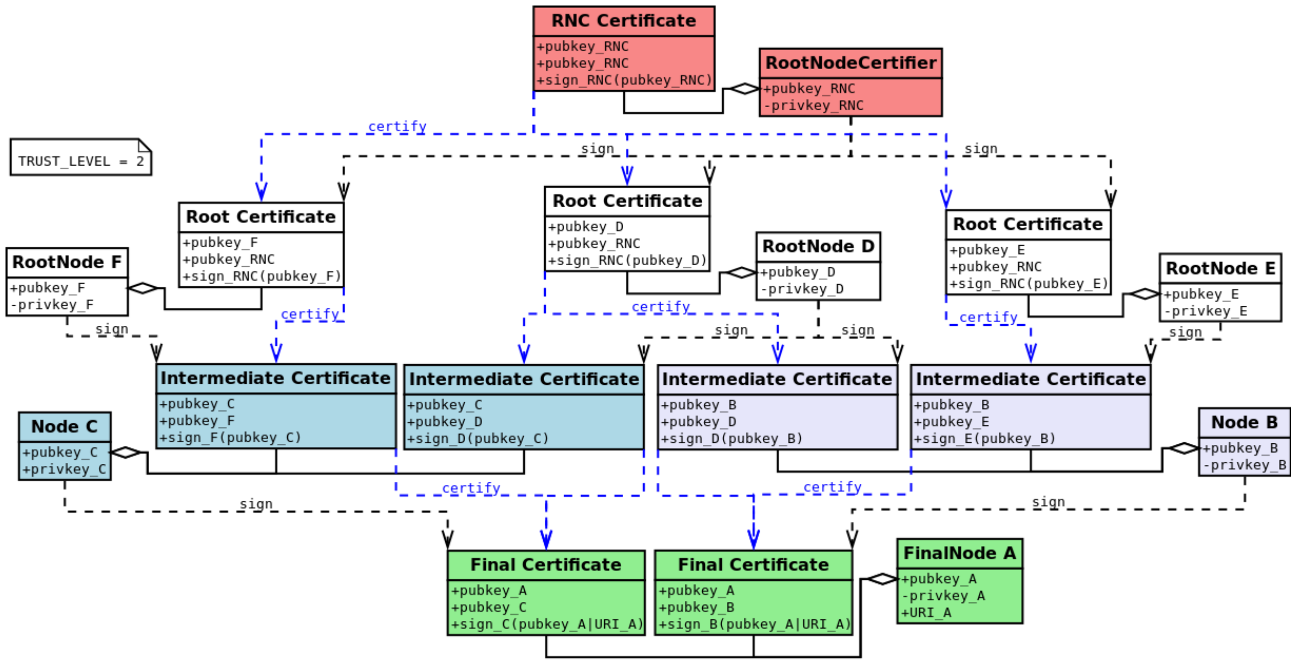

Fig. 2. Example of certificates with trust level = 2

TABLE 1
Dataset characteristics.

	$\|N\|$	$\|E\|$	d_{max}	d_{avg}
$D1$	24k	1M	3k	96
$D2$	36k	2M	5k	87
$D3$	64k	1M	2K	39
$D4$	63k	817k	1k	25
$D5$	42k	1M	4k	65

typing on two desktop keyboards (a French keyboard for users in France and a Norwegian keyboard for users in Norway) *i.e.* AZERTY and QWERTY (this is not a classical QWERTY keyboard, however, we do not use specific Norwegian keys), respectively. An entry is composed of the typing of 5 passwords. Several studies tested that no significant difference between a laptop and a USB keyboard exists [17]. For this dataset, we considered 14 digraphs: latency of 'ca', 'ic', 'ed', 'he', 'pe', 'te', 'ch', 'li', 'ri', 'll', 'on', 'er', 'es' and 'st'. The size of the biometric template is 14. As an entry contains several times each digraph, the template is computed as the mean latency of each 14 digraphs. The reference template is computed by merging several entries (here 14 entries).

9.2 Managing biometrics

For managing biometrics features, we used the dataset described in the previous section and generated legitimate scores by comparing the reference template with templates from the same user. In our attack model, we simulated impostors by replacing the reference template of the victim user with templates coming from other users.

Since the social network graphs we have derived from edge lists contain way more users than the keystroke dynamics dataset, we cannot associate a unique keystroke dynamic with each social network user. User biometrics behavior is thus simulated by randomly picking, for each social network user, a user from the keystroke dynamics datasets. Attacks are simulated by using different keystroke dynamics from the keystroke dynamic dataset. For the BioHashing verification, a different random secret is used for each account and each of its certifiers, and the secret is assumed known by the attacker if the account has been attacked. It is worth noting that the modality can be easily collected by any website, in particular phishing websites. To limit such attacks, Keystroke Dynamics Anonymisation Schemes could be used.

The performances of such systems could be improved using other modalities (s.a. the mouse), that need to enable continuous authentication to not be vulnerable to some attacks, (e.g., lunchtime attacks). The usage of soft biometrics could improve authentication performances and also verify the consistency with the claimed profile (gender, age, ...) and template update techniques could also be implemented to improve the overall system performances as the users' biometrics behavior changes over time.

Biometrics can be used to automatically revoke certifications. However, such revocation could be manually performed by the certifier in response to a flag raised by the biometrics algorithm. This enables the certifier to perform complementary verification before choosing to revoke the certification.

9.3 Experimental results

To validate our proposal, we implemented a simulator in Python 3.11 and ran the experiments on a PC with 16GB of RAM and an $i7$ Intel processor running $Ubuntu$ as the operating system.

Datasets are formatted as edge lists, which is a particular data structure used to represent a graph as a list of its edges. For example, if between nodes a and b there exists a connection (an edge), then in the edge list we will find a row containing the pair (a, b).

Thus, we read the edge lists in such a way that we can deduce all the information about nodes and their relationships. After this operation, we proceed by selecting root nodes. We could have implemented many different algorithms for this selection (for instance, we could have used different centrality measures to select the best nodes). Anyway, we decided to select randomly roots among all the nodes since it would be fairer and more representative of a real situation. In fact, if we had chosen the best nodes by following some measure of centrality, the results would have been somewhat biased and far from a realistic scenario. The only constraint we require in the root nodes selection step is that a node, to be chosen, should have a number of relationships at least equal to `t_level` indicated in the simulation.

Since we used `t_levels` that are in the order of magnitude of ten, we can conclude that this condition is acceptable because most real social profiles easily reach ten friendships. Otherwise, there would have been chosen some root nodes unable to propagate the trust in the network, acting as a bottleneck.

We remark that experiments do not involve the computation of biometric features, as experiments are devoted to validating the trust-based approach. As it is clear from the definition of the model, the biometric component may just change the number of equivalent profiles needed to trust a given profile. Therefore, it can be viewed as an orthogonal component. On the other hand, the effectiveness of biometric-based mechanisms considered in this paper has been shown in the literature.

9.3.1 System performance

In the first experiment, we want to study the performance of our approach when no attack is performed and when participants are always collaborative in trusting a friend node. A relevant value we measure is the *coverage* that could be reached by running our algorithm in the network. Since we use five datasets with different properties, we expect that some differences will be mirrored in the results. Specifically, the more the network is dense and connected, the higher we expect the coverage will be.

We carry out our experiments by varying the following parameters:

- t_level;
- $number\_of\_roots$.

In particular, we use $t\_level = \{3, 6, 8, 10\}$ and $number\_of\_roots = \{0.1\%, 1\%, 2\%, 5\%, 10\%\}$ (these percentages are calculated with respect to the total number of nodes of the given dataset). We run every combination of these two parameters fifty times and we average the results to avoid outlier cases.

In Tables 2, 3, 4, 5, and 6, we report the coverage results of these simulations for each dataset.

As expected, the more connected the datasets, the higher the coverage rates. It is interesting to underline that, in three out of five, the coverage reached is quite relevant with only 0.5% of roots chosen ($D1, D2$, and $D5$) and only in those two datasets less connected ($D3$ and $D4$) this little number of roots is not able to reach a significant number of nodes. By analyzing those two datasets, we see that the average degree of a node is quite low (39 and 25, resp.). This means that every node has, on average, only thirty-nine and twenty-five friends.

We remark that these results have been obtained by running only one cycle of the trust operation (i.e., every node asks to be certificated only once). In a real context, instead, a participant can ask many times a friend to be recognized, so, in this sense, we can conclude that our experiment underestimates the real potential coverage.

TABLE 2
Coverage rate in $D1$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	95,9	90	83,9	72,3
roots = 1%	96,5	91,7	86,7	80,5
roots = 2%	96,9	92,6	89	85,1
roots = 5%	97,2	93,5	90,8	88,1
roots = 10%	97,5	94,5	92,1	89,9

TABLE 3
Coverage rate in $D2$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	95,5	87,5	80,2	71,6
roots = 1%	96	90	84,7	77,3
roots = 2%	96,2	91,29	87,2	82,6
roots = 5%	96,5	93,5	89,19	85,5
roots = 10%	97	94,5	90,6	87,7

TABLE 4
Coverage rate in $D3$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	69,9	44,4	6,6	4,6
roots = 1%	70,5	47	35,2	23,7
roots = 2%	71	49,4	38,9	29,6
roots = 5%	72	52	43,4	36,7
roots = 10%	73,5	55,1	47	41

TABLE 5
Coverage rate in $D4$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	70,1	44,7	8,9	5,7
roots = 1%	71,2	47,1	34,4	20,2
roots = 2%	72	49,3	38,2	29,9
roots = 5%	73	52,4	43,5	36,4
roots = 10%	74,5	55,7	47,8	41,8

TABLE 6
Coverage rate in $D5$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	93,1	82,8	75,1	64,4
roots = 1%	93,8	84,3	77,4	68,9
roots = 2%	94,3	86	79,7	72,7
roots = 5%	94,9	87,9	83,2	77,8
roots = 10%	95,4	89,5	85,6	81,4

9.3.2 Resilience of the system

In this section, we study the resilience of our proposal to the *slandering attack* (described in Section 8). Under this attack, a certain amount of root nodes become malicious and do not play anymore a collaborative and positive role in the propagation of the trust. To better evaluate the results of this experiment, we compare the performance of our proposal with that of [30] (presented in Section 2, where the parameter $\alpha = 1$ because the used datasets allow us to identify positive and negative interactions. Specifically, under attack, malicious nodes propagate negative interactions. Under attack, we randomly picked 30% of the roots as malicious nodes. As evaluation metrics, we use *coverage loss* defined as $1 - \frac{\hat{T}_a}{\hat{T}}$, where $\hat{T}_a$ is the average trust of nodes under attack and $\hat{T}$ is the average trust of nodes when no attack is carried out.

We varied the same values as the first experiment (t_level and $number\_of\_roots$) and, in Table 7, we report the percentage of coverage loss when varying the percentage of root nodes $(0.5, 1, 2, 5, 10)$ and, for the sake of presentation, we averaged the results. In Table 8, we report the percentage of coverage loss measured for the *Strust* model [30] with the same experiment setup.

From the analysis of these experiments, we see that a *slandering attack carried out by the* 30% *of root nodes slightly reduces the performance of our proposal (on average, about 7%). Comparing these results with that of* [30], we can conclude that our approach contrasts *slandering attacks*.

TABLE 7
Trust loss measured for our proposal (in percentage)

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
D1	0,64	0,96	9,09	12,08
D2	0,71	1,37	1,69	13,88
D3	1,92	3,07	12,56	21,30
D4	2,04	4,32	10,48	16,40
D5	0,80	1,10	2,21	22,30

TABLE 8
Trust loss measured for *Strust* [30] (in percentage).

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
D1	29,41	26,52	32,40	33,64
D2	31,22	35,36	36,80	37,72
D3	9,98	18,36	18,67	22,74
D4	13,04	13,98	22,73	27,53
D5	35,49	39,84	40,01	41,92

10 Conclusion and perspectives

Trusting identities in social networks is an important challenge in the literature. The approach we designed for this purpose exploits a collaboration among social network users and a keystroke-dynamics-based technique. In our proposal, each user shares with the other users the information about the neighborhoods. This information is propagated among the network in a trusted way. We provide the theoretical characterization of the proposed approach and its validation by experiments carried out on real-life data. The experiment results show that trusted information reaches most of the nodes (depending on some system parameters). The novelty of our proposal is to distribute, in a decentralized fashion, over the social network itself, the trust mechanism. Moreover, we embed in the trust mechanism behavioral biometric features, to better support the maintenance of trust values and then isolate possible attacked profiles. We remark that the paper addresses the biometric component of the model in a concrete and definite way, by identifying which type of biometric data is suitable to our purpose, and by defining how the biometric component affects the trust values. Therefore, we argued the effectiveness of the trust mechanism is the most meaningful aspect to assess. Regarding this aspect, it is worth noting that the exchanging of messages from a social network user to another profile, in principle, is a reason for characterizing that profile as trusted provided that at least one of the two profiles is real. This makes recursive the problem and that needs further research. As a future work, we plan to implement the model in a *homemade* online social network and experiment it in a real-life setting, possibly in a research project reaching large communities and involving industrial partners. This will allow us to collect real biometric data for proof of concept validation.

References

[1] Abdul-Rahman A (1997) The pgp trust model. In: *EDI-Forum: the Journal of Electronic Commerce*, volume 10. pp. 27–31.
[2] Ahmad, N., Laplante, P. A., DeFranco, J. F., and Kassab, M. (2021). A cybersecurity educated community. *IEEE Transactions on Emerging Topics in Computing*. 10(3), 1456-1463.
[3] Bhana B and Flowerday S (2020) Passphrase and keystroke dynamics authentication: Usable security. *Computers & Security* 96: 101925.
[4] Buccafurri F, Lax G and Migdal D and Nicolazzo S and Nocera A and Rosenberger C (2017) Contrasting false identities in social networks by trust chains and biometric reinforcement. In: *2017 International Conference on Cyberworlds (CW)* IEEE, pp. 17–24.
[5] Watts, Duncan J (1999) Networks, dynamics, and the small-world phenomenon. *American Journal of sociology* 105(2): 493–527.
[6] Bakhshandeh, R and Samadi, M and Azimifar, Z and Schaeffer, J (2011) Degrees of separation in social networks. In: *Proceedings of the International Symposium on Combinatorial Search* 2(1): 18–23.
[7] Buccafurri F, Musarella L and Nardone R (2019) Enabling propagation in web of trust by ethereum. In: *Proceedings of the 23rd International Database Applications & Engineering Symposium*. pp. 1–6.
[8] Choi K, Toh KA and Byun H (2012) Incremental face recognition for large-scale social network services. *Pattern Recognition* 45(8): 2868–2883.
[9] Conti M, Poovendran R and Secchiero M (2012) Fakebook: Detecting fake profiles in on-line social networks. In: *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*. IEEE, pp. 1071–1078.
[10] Cutillo LA, Molva R and Strufe T (2009) Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine* 47(12).

[11] Dafer M and El-Abed M (2017) Evaluation of keystroke dynamics authentication systems: Analysis of physical and touch screen keyboards. In: *Developing Next-Generation Countermeasures for Homeland Security Threat Prevention*. IGI Global, pp. 306–329.

[12] Deutschmann I, Nordström P and Nilsson L (2013) Continuous authentication using behavioral biometrics. *IT Professional* 15(4): 12–15.

[13] Yasyn Elyusufi, Zakaria Elyusufi and M'hamed Ait Kbir (2019) Social networks fake profiles detection using machine learning algorithms. In: *3rd International Conference on Smart City Applications, 2019.* Springer, pp. 30–40.

[14] Fairhurst M and Da Costa-Abreu M (2011) Using keystroke dynamics for gender identification in social network environment .

[15] Gadiya M and Jain S (2016) Gender prediction using images posted on online social networks .

[16] Gaines R, Lisowski W, Press S and Shapiro N (1980) Authentication by keystroke timing: some preliminary results. Technical report, Rand Corporation.

[17] Giot R, El-Abed M, Hemery B and Rosenberger C (2011) Unconstrained keystroke dynamics authentication with shared secret. *Computers & Security* 30(6): 427–445.

[18] Giot R, El-Abed M and Rosenberger C (2009) Greyc keystroke: a benchmark for keystroke dynamics biometric systems. In: *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*. IEEE, pp. 1–6.

[19] Guo B, Ding Y, Sun Y, Ma S, Li k annd Yu Zhiwen (2021) The mass, fake news, and cognition security *Frontiers of Computer Science* 15(3): 1–13.

[20] Haenni R and Jonczy J (2007) A New Approach to PGP's Web of Trust. In: *EEMA'07, European e-Identity Conference*.

[21] Hamzelou N, Ashtiani M and Sadeghi R (2021) A propagation trust model in social networks based on the a* algorithm and multi-criteria decision making. *Computing* 103(5): 827–867.

[22] Hocquet S, Ramel JY and Cardot H (2007) User classification for keystroke dynamics authentication. In: *The Sixth International Conference on Biometrics (ICB2007)*. pp. 531–539.

[23] Idrus SZS, Cherrier E, Rosenberger C and Bours P (2013) Soft biometrics database: A benchmark for keystroke dynamics biometric systems. In: *Biometrics Special Interest Group (BIOSIG), 2013 international conference of the*. IEEE, pp. 1–8.

[24] Wahab, A. A., Hou, D., Banavar, M., Schuckers, S., Eaton, K., Baldwin, J., Wright, R. (2022, April). Shared multi-keyboard and bilingual datasets to support keystroke dynamics research. Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy (pp. 236-241).

[25] Jin ATB, Ling DNC and Goh A (2004) Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition* 37(11): 2245–2255.

[26] Khan A, Brohman K and Addas S (2021) The anatomy of 'fake news': Studying false messages as digital objects. *Journal of Information Technology* : 02683962211037693.

[27] Matthiesen JJ and Brefeld U (2020) Assessing user behavior by mouse movements. In: *International Conference on Human-Computer Interaction*. Springer, pp. 68–75.

[28] Mondal S and Bours P (2013) Continuous authentication using behavioural biometrics. In: *Collaborative European Research Conference (CERC'13)*. pp. 130–140.

[29] Mondal S and Bours P (2017) Person identification by keystroke dynamics using pairwise user coupling. *IEEE Transactions on Information Forensics and Security* 12(6): 1319–1329.

[30] Nepal S, Bista SK and Paris C (2015) Behavior-based propagation of trust in social networks with restricted and anonymous participation. *Computational Intelligence* 31(4): 642–668.

[31] Patel V, Ratha NK and Chellappa R (2015) Cancelable biometrics: A review. *IEEE Signal Processing Magazine* : 54–65.

[32] Peng S, Yang A, Cao L, Yu S and Xie D (2017) Social influence modeling using information theory in mobile social networks. *Information Sciences* 379: 146–159.

[33] Ramalingam D and Chinnaiah V (2018) Fake profile detection techniques in large-scale online social networks: A comprehensive review. *Computers & Electrical Engineering* 65: 165–177.

[34] Tsimperidis, I., Katos, V. (2013, September). Keystroke forensics: are you typing on a desktop or a laptop?. Proceedings of the 6th Balkan Conference in Informatics (pp. 89-94).

[35] Rossi RA and Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. In: *AAAI*. http://networkrepository.com.

[36] Sherchan W, Nepal S and Paris C (2013) A survey of trust in social networks. *ACM Computing Surveys (CSUR)* 45(4): 47.

[37] Shetty NP, Muniyal B, Anand A and Kumar S (2022) An enhanced sybil guard to detect bots in online social networks. *Journal of Cyber Security and Mobility* : 105–126.

[38] Song X, Zhao P, Wang M and Yan C (2016) A continuous identity verification method based on free-text keystroke dynamics. In: *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE, pp. 000206–000210.

[39] Tiwari V (2017) Analysis and detection of fake profile over social network. In: *Computing, Communication and Automation (ICCCA), 2017 International Conference on*. IEEE, pp. 175–179.

[40] Tsimperidis G, Katos V and Rostami S (2017) Age detection through keystroke dynamics from user authentication failures. *International Journal of Digital Crime and Forensics (IJDCF)* 9(1): 1–16.

[41] Upadhyaya SJ (2017) Continuous authentication using behavioral biometrics. In: *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*. ACM, pp. 29–29.

[42] Wanda P and Jie HJ (2020) Deepprofile: Finding fake profile in online social network using dynamic cnn. *Journal of Information Security and Applications* 52: 102465.

[43] Wu J, Xiong R and Chiclana F (2016) Uninorm trust propagation and aggregation methods for group decision making in social network with four tuple information. *Knowledge-Based Systems* 96: 29–39.

[44] Xu X, Yu J, Chen Y, Hua Q, Zhu Y, Chen YC and Li M (2020) Touchpass: towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations. In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. pp. 1–13.

[45] Yang Y, Guo B, Wang Z, Li M, Yu Z and Zhou X (2019) Behavesense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics. *Ad Hoc Networks* 84: 9–18.

Francesco Buccafurri is a full professor of computer science at the University Mediterranea of Reggio Calabria, Italy. In 1995 he took the Ph.D. degree in CS at the University of Calabria. In 1996 he was visiting researcher at Vienna University of Technology. His research interests include cybersecurity, privacy, social networks, e-government, and P2P systems. He has published more than 160 papers in top-level journals and conference proceedings. He serves as a referee for international journals and he is a member of several conference PCs. He is Associate Editor of Information Sciences (Elsevier) and IEEE Transactions on Industrial Informatics and played the role of PC chair and PC member in many international conferences. He is member of the IEEE computer society.

Gianluca Lax is an Associate Professor of Computer Science at the University Mediterranea of Reggio Calabria, Italy. In 2005, he received his Ph.D. in computer science from the University of Calabria. In 2018, he got the habilitation as full professor of computer science. His main research interests include information security and social network analysis. He is an author of more than 140 papers published in leading international journals and conference proceedings. He serves as a referee for many international journals and is in the program committee of many conferences. He is also included in the editorial board of several international journals and participates in many funded projects.

Denis Migdal is an associate professor at the University of Clermont Auvergne (UCA) since 2021. He obtained his Ph.D. degree in Computer Science in 2019 at the University of Caen Normandy (UNICAEN). From 2016 to 2020 he was part of the Research Group in Computer Science, Image and Instrumentation of Caen (GREYC) laboratory. He is now part of the Laboratory of Informatics, Modeling and Optimization of the Systems (LIMOS) since 2021. His research focuses on Biometrics and Keystroke Dynamics.

Lorenzo Musarella is an Assistant Professor of Computer Science at the University Mediterranea of Reggio Calabria, Italy. In 2021 he received his Ph.D. in Information Engineering from the University Mediterranea of Reggio Calabria. His research interest includes cybersecurity, trust, privacy, and network analysis. He is author of 15 papers published in top-level international journals and conference proceedings.

Christophe Rosenberger obtained his Ph.D. in Information Technology from the University of Rennes 1 in 1999. His PhD thesis work was undertaken at ENSSAT in Lannion between 1996 and 1999 in the field of hyperspectral image segmentation. He joined the ENSI de Bourges school of engineering in Bourges (known as INSA Centre Val de Loire) as assistant professor in 2000. In 2007, he joined the ENSICAEN school of engineering in Caen as full professor. He belongs to the SAFE (Security, Architecture, Forensics, biomEtrics) research group in the GREYC research lab. His current work focuses on the domain of computer security, in particular research activities in biometrics (keystroke dynamics, soft biometrics, evaluation of biometric systems, fingerprint quality assessment...).

Combining Trust Graphs and Keystroke Dynamics to Counter Fake Identities in Social Networks

Francesco Buccafurri, *Member, IEEE,* Gianluca Lax, *Member, IEEE,* Denis Migdal, Lorenzo Musarella and Christophe Rosenberger

Abstract—Fake identity in social networks is a phenomenon that is strongly increasing, and it is used for discovering personal information, identity theft, influencing people, spreading fake news, fraud, and so on. In this paper, we face this problem by introducing the concept of certified social profiles and by propagating this property through a collaborative approach that exploits keystroke-dynamic-recognition techniques to identify illegal access to certified profiles. We propose a decentralized approach to compute the trust level of a social profile, and we show the robustness of the proposal by analyzing the security of the trust mechanism through experimental validation.

Index Terms—Social Networks, Trust, Fake Profiles, Keystroke Dynamics

1 Introduction

In daily life, all communications are taking rapidly the direction of the digital and the virtual domain. In particular, social networks and social media platforms represent huge sources for information sharing where people can interact with each other easily and in a fast way. At the same time, online social networks are big catching areas for collecting and spreading trash news and fake information as well [19], [26]. Unfortunately, the awareness of users of the most common cyber threats is not growing as quickly as risks do. Indeed, the *online* community still has to be educated concerning cyber threats [2]. For this reason, it is important to improve the security of services, and trust represents a fundamental property that users must look for when they interact on social networks. Usually, fake news is shared and forwarded mostly by fake profiles.

Social network profiles whose claimed identity does not match the real user are certainly potential security threats on the Web. This happens in two cases. The first case is that of fake profiles, in which the attacker intentionally creates a clone of a real-life identity profile of the victim, pretending to be them in the interactions. The second case is that of compromised profiles, in which an intruder, permanently or temporarily, uses the real social profile of the victim fraudulently. In both cases, the risk of anomalous behavior with potential damage to the victim's reputation, espionage, or social engineering attacks toward people connected to the victim is very high.

- F. Buccafurri, G. Lax, and Lorenzo Musarella are with the DIIES Dept., University Mediterranea of Reggio Calabria, Italy.
E-mail: bucca@unirc.it, lax@unirc.it, lorenzo.musarella@unirc.it
- D. Migdal is with Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, LIMOS, France.
E-mail: denis.migdal@uca.fr
- C. Rosenberger is with Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France.
E-mail: christophe.rosenberger@ensicaen.fr

The problem faced in this paper regards the fact that, through fake profiles, attackers can entice users to give up personal data, hijack them toward infected websites and, once their email addresses are known, launch spear-phishing campaigns. Several studies have been proposed in the literature to counter this problem [9], [13], [21], [30], mainly based on associating each social profile with a certain degree of trust. All the existing proposals require a strong analysis effort by the social network provider, which takes into account all the behavioral and topological information of the profiles.

Differently from these pioneer methods, we propose an approach based on a collaborative trust mechanism that may operate in a truly decentralized fashion, in which trust is combined with behavioral biometric mechanisms to counter profile compromising. The novelty of our proposal is that the computation of the trust level is decentralized: indeed, it exploits only user interactions and does not require any central authority for trust management or computation.

The underlying idea exploits the social structure of our domain. The trust model is based on a robust implementation of the *word-of-mouth* approach. Robustness is obtained by redundancy. In words, we follow the principle that if a sufficient number of people trust the identity of a social network profile, we can trust it as well. This way, we obtain a graph of trust, because we propagate trust under the basic assumption that a fake user (and then fake behavior) is transitively excluded. We base our assumption on the consideration that, when the real-life identity is known, sanctions are facilitated in case of misbehavior (e.g., victims could sue users who certified the perpetrator), thus misbehavior is prevented.

Trust is obtained through redundant trust chains in which any node plays a role similar to an intermediate certifier in a certification chain until a certified profile (i.e., a trust anchor) is reached. Our model requires the presence of some profiles certified by a Trusted Third Party. To identify possible intrusions in a legitimate profile, the trust model takes also into account the behavioral biometric traits of users that they record and verify in a peer-to-peer fashion. Importantly, no biometrics are stored by the social network provider. In other words, the word-of-mouth mechanism propagates the information that the current behavior of a given node is not compliant with that of the initial safe state, thus reducing the trust of the community towards that node. Keystroke dynamics is used as a behavioral biometric modality, which is very easy to collect on web pages (e.g., by using JavaScript code) and allows a simple and low-cost solution to verify the identity of a user [29], [38]. We can summarize the contribution of this paper as follows:

- We propose a theoretical framework based on a resilient trust graph model that enables the computation of a trust degree with respect to a social network participant in a decentralized way.
- Such framework is then implemented and several experiments are carried out with real datasets to validate the system performance and resilience, comparing such results with other proposals as well.
- The proposal exploits the behavioral biometric feature of the user, in terms of keystroke dynamics, to identify possible intrusion with respect to a legitimate profile and to measure its trustworthiness.
- We provide the security analysis, validating the robustness of our approach.

The structure of the paper is the following. In Section 2, we contextualize our proposal in the state of the art. Then, in Section 3 we provide a general overview of our approach, while in Section 4 we introduce the biometric features and the behavioral biometric modality used in our approach. Section 5 describes how our trust model works and it provides the theoretical support for the trust mechanism. In Section 6 we present a decentralized way to compute the trust level and in Section 7 we describe the system functions that a social network platform needs to implement our approach. In Section 8 we test our approach and show how security properties are fulfilled. Then, Section 9 includes details about datasets and implementation issues related to our approach. Finally, in Section 10 we draw our conclusions and discuss future work.

2 Related Work

In online social networks, the detection of fake profiles is becoming every day more and more important because there are many threats (such as scamming, trolling, phishing, Sybil attacks, and social bots) that need to be faced [33], [39].

In this scenario, it can be helpful to create a model that includes the dynamic computation of a trust degree for each user. Trust is becoming a fundamental element of a successful social network [36], and it derives from the "social capital", which is based on the density of interactions among people.

The concept of trust applied to the digital domain has been introduced through PGP (Pretty Good Privacy). The original intent of PGP was to provide a "cryptographic tool for the masses". Its main purpose was to encrypt e-mail messages using public or conventional key encryption. For this reason, PGP does not adopt the traditional hierarchical trust architecture but chooses the "web of trust" approach, in which the users sign each other's public keys. Thus, a network of public keys is progressively originated, with links formed by signatures. This way, there is no need for a central authority [1]. Although the first instance of PGP allowed determining the maximum length of the certification chain through the `CERT_DEPTH` parameter, researchers have not shown much interest in seeking new solutions for trust propagation because it can be quite difficult to use it properly in real applications [1], [20]. In particular, only in recent years, this aspect has been investigated by [7], in which authors propose a blockchain-based solution.

A survey of trust in social networks [36] introduces three categories of trust models: *(i) graph-based* models, which consider only how members are related to each other and do not consider the real interaction between them; *(ii) interaction-based* models, which consider only interaction in the community and ignore the social network structure; *(iii) hybrid* models, which try to consider both the aspects to compute the social trust.

One of the most crucial points of this research area is, surely, how trust propagates in the network. For this purpose, the small world theorem and the social network analysis techniques have been explored. In particular, the authors of [30] enhanced the *STrust* model by proposing an association-based trust propagation model that considers two types of interactions: active and passive. *STrust* model is computed as a combination of the popularity trust and the engagement trust as $STrust(u) = \alpha PopTrust(u) + (1 - \alpha)EngTrust$, which are defined in [30]. The paper [21] proposes an approach to select the most trusted path between two nodes of a social network by merging seven criteria, such as profile similarity, topological similarity, Dunbar's theorem, and other measures related to social network analysis. The authors specify that their approach allows them to consider longer trust propagation paths than others. A model based on the uniform trust propagation called SN-GDM (Social Network-based Group Decision Making) is presented in [43]. In particular, the authors propose the two concepts of *Trust Score (TS)* and *Knowledge Degree (KD)*, which are combined to define a social trust value that does not lose any trust information during its propagation. Their model requires the intervention of Trusted Third Parts (TTPs) for the validation of results.

A decentralized and privacy-preserving online social network is presented in [10]. Here, the authors propose *Safebook*, a system that provides registered users with data storage and data management functions relying on trust relationships that are part of social networks in real life.

Recently, the use of machine learning has been proposed for detecting fake profiles. In [42], a dynamic Convolutional Neural Network (CNN) is built for fake profile classification that provides better results in terms of accuracy and loss than other commonly used learning algorithms. Another machine-learning-based classifier to detect bots in online social networks is proposed in [37] to counter Sybil attacks.

Our approach is also related to the concept of information diffusion in online social networks, in the sense that the roots influence the trust values of other nodes in the network following the rules of information flow [32]. Most of these works study how information flows in online social networks and propose strategies to maximize this diffusion by identifying strategic nodes for information propagation. The aim of our paper is somehow orthogonal to these studies and may exploit these solutions to improve trust propagation through the social network.

Furthermore, our approach leverages biometric data, which is a practice not new for social network applications [8], even though there are few papers in this area. In particular, most of the papers focus on user authentication using biometric data to enhance its security. Some papers in the literature considered soft biometrics with possible applications to social networks [14]. Most of the works consider gender recognition by analyzing the type of images posted or the keystroke dynamics [15], [40]. To the best of our knowledge, no work considers keystroke dynamics as a solution to enhance trust in social networks.

Several works exploit biometrics to implement continuous authentication schemes [12], [28], [41]. In high-security environments, the typical session-level authentication can be exposed to session hijacking, in which an attacker targets a post-authenticated session. In those contexts, continuous and real-time verification of user identity may become mandatory, and a lot of research effort has been devoted to the use of biometrics as a means to achieve this objective.

However, the goal of these strategies is very far away from ours. In detail, our approach does not aim at proposing a strategy to continuously verify that an active login session is controlled by the right user. Instead, our approach exploits biometric data as feedback to our trust model to measure the trustworthiness of an online profile.

From the literature review described earlier, it appears that the objective pursued by our paper is new. Therefore, to the best of our knowledge, there is no approach that can be (analytically or experimentally) fully compared with our proposal. In Section 9.3.2, we provide a comparative evaluation between our method and that proposed in [30] from the perspective of resilience against slandering attacks.

The rough original idea at the basis of this paper has been initially presented in [4]. However, there is a significant difference between the two papers. In particular, being [4] the report of early-stage research, the theoretical framework was not complete, and the proof of theorems was not included. No distributed implementation of trust computation and certification infrastructure was provided, as well as the concept of *certificate* was not presented. Moreover, there was not any security analysis. In addition, the experimental validation of the proposal carried out in the old version is very preliminary, because it only considered one small synthetic dataset (i.e., 2,500 nodes) with very few experiments instead of five real, heterogeneous, and bigger datasets. We extend the experiments through a comparative analysis of our method with respect to the literature and by considering (experimentally) an attack to demonstrate the robustness of our approach. We improve the part of the paper including biometrics and, finally, we add the spatial and temporal complexity analysis of all the functionalities of our proposal.

3 Overview of the Approach

In this section, we provide a general overview of how our solution works.

First, we highlight that our proposal is applied to online social networks and that it works by employing trust chains built among users. In particular, each chain starts from a root node, which is a social profile certified by a Trusted Third Party (TTP). To build a root profile, a user has to register with the social network via TTP, by executing an identification process proving their real-life identity. This could be achieved by using a public digital identity system. In addition, in this phase, TTP gathers the biometric (behavioral) parameters of the user to create a model that will be exploited in future interactions with the user to adapt the trust level and verify whether the account is still under the user's control. Indeed, in the negative case, the certificate associated with that profile will be revoked.

At the initial state of our protocol, the list of certified profiles coincides with the set of roots. At this point, new profiles could be certified by root nodes via certificate generation. When a new profile reaches a given *trust level*, it will be considered certified and it will play an active role in trust propagation.

We model the social network as a directed graph $G = \langle N, E\rangle$, where $N = N_c \bigcup N_{nc}$ is the union between the set of certified profiles N_c and the set of non-certified profiles N_{nc}, and E is the set of edges representing the friendship among these peers. We use the notion of directed graphs because they handle the case of symmetric friendship (as happens in Facebook) simply by including two edges in both directions. For example, if we want to represent a Facebook friendship between i and j, then we set the edges $E_{i,j}$ and $E_{j,i}$.

Any node of the social network (both certified and non-certified) may directly recognize some of its direct contacts. The basic idea is that a node recognizes only those nodes for which past real-life interactions occurred, allowing the node to conclude, also by using external knowledge, that the claimed identity is real (this situation typically happens for a significant portion of social network contacts). When a safe interaction happens (for instance, at the first message exchange allowing the recognition of the interlocutor) the profile acts as a recognizer and builds a biometric model of the recognizing node. This way, a subsequent intrusion can be detected. Furthermore, we remark that only a node already recognized can play the role of the recognizer. The underlying rationale is that the misbehavior of a user is directly connected to their anonymity in the social network. In other words, by making the recognizing process fully accounted and traced (and related to a real-life identity), we can increase the trust in recognized identities, provided that transitively, the process leads to root nodes.

Since we cannot give an absolute value to the above principle, we have to increase the level of trust by requiring redundancy in the recognizing process, thus making more improbable the conjunct misbehavior of identified recognizers. The level of redundancy sets the level of trust. The biometric model built by any participant, allows us to detect possible profile compromising, thus including in the trust also the expectation that an initially identified profile is still under the exclusive control of the legitimate owner. It is worth noting that, in principle, the biometric model could be learned by exploiting multiple channels (social network interactions, chats, shared editing, and so on).

We remark that the proposed approach is not aimed at defining a digital identity system, since only a level of trust is obtained. In fact, when a user A recognizes a user B, they are stating that B is not claiming a false identity, not the veracity of all published information. The quantity and quality of information needed by A to reach this conclusion depends on the social context. Besides name and surname, they may regard other information, such as age, job, and friendships.

Furthermore, we highlight that this approach does not recognize directly untrusted social profiles, but it aims to provide a certain degree of positive trust that is tuned and refined considering all the features, including biometrics, which we will explain in the following so that we can consider a social profile as certified. Clearly, if a certain social profile is not certified because it has a low trust degree or has no degree computed at all, it would represent an implicit suggestion to the user that such a social profile could convey dangers.

4 Biometric Features

In this paper, we propose to use biometric features to guarantee the previous security requirements. We focus in this work on a behavioral biometric modality as it can be easily collected in a transparent way [45]. Among existing solutions, we can cite mouse dynamics [27], touchscreen interactions [44], or keystroke dynamics [3]. The last technique is a behavioral biometric modality consisting of analyzing the user's way of typing on a keyboard. This biometric information can be computed easily on the Internet using simple JavaScript code. Keystroke dynamics is a biometric modality identified more than 40 years ago. We can cite a pioneer work in 1980 with a study where seven secretaries were asked to type three different texts [16]. The results were promising but lacked a sufficient number of users involved in the database. Most research works in keystroke dynamics assume a quasi-keyboard invariance or intend to propose algorithms dealing with slight behavioral modifications related to the keyboard type. In a previous study [18], experimental results from data collected from 133 users when using 2 types of keyboard (laptop and desktop) showed similar performance considering the keyboard used for enrollment or verification steps. However, some papers showed some performance differences when different keyboards were used during enrollment and verification. We can mention the following study [34] with 2 used keyboards (laptop and desktop) even if collected data are only from 17 users. A more recent paper [24] involving data from 4 keyboards and 86 users can also be mentioned. We assume in this work, as many in the literature, that the keyboard has a low impact on feature extraction. The use of mobile devices is not considered in this paper but many methods exist to deal with this type of capture [11].

The capture process of keystroke dynamics is shown in Figure 1. It consists in computing several features when the keys are pressed and released (timestamp of the event, code of the key, ...) provided by any Operating System (OS). The feature extraction consists mainly in measuring different latency and duration time between each key. Figure 1 shows an example where the user presses two keys on the keyboard. The user presses "C" at T1, "O" at T2, and releases "C" at T3 and "O" at T4. Note that the following relation is always respected: $T3 > T1$ and $T4 > T2$ (we always release a key after pressing it), while the following condition may not always be respected: $T2 > T3$ (because, as in our example, a user may press another key before releasing the previous one). We can extract three latency values (T2-T1, T4-T3, T2-T3) which we call PP (latency between two pressures), RR (latency between two releases), RP (latency between one release and one pressure) respectively, and one type of duration (T3-T1 or T4-T2) which we call PR (duration of a key press). The described process is repeated for all the keys.

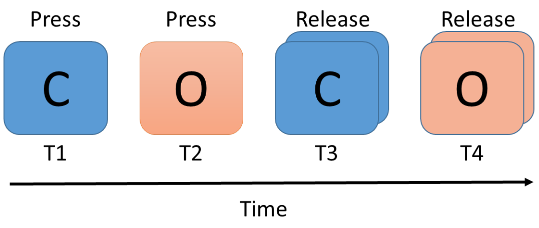

Fig. 1. Information captured in a keystroke dynamics system when pressing C and O keys [17].

Keystroke dynamics can be used either with passwords to enhance the security of user authentication or on free text. For example, the keystroke dynamics of a user could be analyzed while exchanging with another (e.g., through a social network chat). Subsequently, we consider the different timing information between two-character sequences known as *digraphs*. Digraphs are the latency times between two successive keystrokes. The biometric template associated with user z is composed of n digraphs $T_z = \{T_z^1, .., T_z^n\}$. The considered digraphs could be associated with one language. In this work (see Section 8.3), we considered 14 different values (common ones in English). If we consider a higher value of n, a few sentences will allow us to compute the keystroke dynamics feature vector. Another solution is to compute the distribution of latency values from the text typed by a user as a behavioral signature.

First, we need to generate the reference biometric template for each user by analyzing their keystroke dynamics during a period of time when we assume only the legitimate user interacts with the social network. To guarantee user privacy, we apply a biometric template protection scheme on digraph data (see Algorithm 1). The BioHashing algorithm [25] is applied to biometric templates that are represented by real-valued vectors of fixed length (so the metric used to evaluate the similarity between two biometric features is the Euclidean distance). It generates binary templates of length lower than or equal to the original length (here, the metric D_T used to evaluate the similarity between two transformed templates is the Hamming distance). A complete review of cancelable biometric systems has been reported by [31].

The reference template of user z is defined by $\widetilde{B}_z = \{E[B_z], \sigma[B_z]\}$ where $E[.]$ corresponds to the average value of biometric templates of user z and $\sigma[.]$ the associated

Algorithm 1 BioHashing

```
1: **Inputs**
2: $T = (T_1, \dots, T_n)$: biometric template,
3: $K_z$: secret seed
4: **Output** $B = (B_1, \dots, B_m)$: BioCode
5: Generation with the seed $K_z$ of $m$ pseudorandom vectors $V_1, \dots, V_m$ of length $n$,
6: Orthogonalize vectors with the Gram-Schmidt algorithm,
7: **for** $i = 1, \dots, m$ **do** compute $x_i =< T, V_i >$.
8: Compute BioCode:
       $B_i = \begin{cases} 0 & \text{if} \quad x_i < \tau \\ 1 & \text{if} \quad x_i \geq \tau \end{cases}$
   where $\tau$ is a given threshold, generally equal to 0.
```

standard deviation. To decide if a biometric template B_x of size n belongs to user z, we need to compare it with the reference template of user z denoted $\tilde{B_z}$ as follows [22]:

$$Score = 1 - \frac{1}{n}\sum_{i=1}^{n} e^{-\frac{|B_x - E[B_z]|}{\sigma[B_z]}} \tag{1}$$

This score gives a confidence measure about how much the user z is legitimate and will be used in the trust model proposed in the next section.

As we will see in the rest of the paper, the biometric system is used in such a way that the following security properties are guaranteed:

- *Confidentiality:* the biometric data are not disclosed to others.
- *Non-reversibility:* the biometric template is not enabled to retrieve the biometric data.
- *Discriminant:* the biometrics discriminate users.
- *Constant:* the biometrics remain constant.
- *Non-usurpation:* a user is not able to forge/imitate the biometric of another user.
- *Not-costly:* in terms of memory/time/ergonomics/money (no additional devices).

5 The Trust Graph

In this section, we give the theoretical framework underlying our trust model works.

Throughout this section, consider given a directed graph $G = \langle N, E \rangle$ representing a social network and a *redundancy* parameter t, i.e., a positive integer representing a level of trust. Let TTP be a Trusted Third Party. Let denote by N_c the set of *certified nodes*, i.e., the nodes whose identity is assured and monitored by TTP. Given a node $u \in N$, we denote by $\Gamma(u)$ the set of neighbors of u (i.e., adjacent nodes). Moreover, we denote by $R(u) \subseteq \Gamma(u)$ the set of nodes recognized by u.

Our trust model is based on the notion of recognition done by a certain user (in this formal framework just a node) towards other directly connected users. However, there is a recursive requirement. If we establish that t is the minimum number of recognitions that a user should receive to be considered trusted, we require that such recognitions, to be valid, must be done by trusted users (i.e., either users who received at least t recognition or certified users, which play the role of roots). This is formally encoded in the next definition:

Definition 1. We say that a node $u \in N$ is *t-recognized* (*in $A \subseteq N$*) if either: (i) $u \in N_c$ (i.e., is a certified node), or (ii) there exist t other t-recognized nodes in A that recognize u.

When the set A of the definition above is not specified, we intend that a node is t-recognized in N. From the above definition, it immediately follows that nodes in N_c are t-recognized for any t and in any set A.

We want now to understand how to characterize (and then compute) the set of users who, thanks to the propagation mechanism enforced by the above recursive definition, are trusted (with level t), when a set of roots N_c is fixed.

To do this, we first need to introduce the notion of *t-closed* set. Informally, a *t-closed* set is a set of set of t-recognized nodes that is closed with respect to the trust propagation mechanism. More formally:

Definition 2. A set $A \subseteq N$ of t-recognized nodes in A is said *t-closed*, if there is no $u \in N \setminus A$ that is t-recognized in A too.

From the above definition, it immediately follows that all certified nodes must belong to any t-closed set.

It is easy to see that the operator $\subseteq$ induces a partial order over the set of t-closed sets, which is a lower semi-lattice. We denote by $N^t \subseteq 2^N$ the set of non-empty t-closed subsets of N. N^t is then a lower semi-lattice according to $\subseteq$. We denote by N_b^t the bottom of N^t.

In our model, the role of N_b^t is central, because it includes exactly all nodes that are t-recognized, but, due to subset minimality, they do not form clusters whose recognizing is only mutual. In other words, N_b^t is the set of nodes for which trust paths start from certified nodes. Therefore, N_b^t represents the set of trust nodes, as more formally stated in the next definition.

Definition 3. Given a node $u \in N$ we say that u is *t-trusted (in N)* if $u \in N_b^t$. N_b^t is also said the *set of t-trusted nodes (in N)*.

It is now important to understand how to compute N_b^t. To do this, we provide an operational definition of N_b^t, based on the fixpoint of a monotone operator Λ_t, called *t-recognizing operator*. From this definition, it easily arises that the trust of nodes in N_b^t can be directly or indirectly linked to (at least) t certified nodes.

Definition 4. Given $t > 0$, let denote by $2_t^N \subseteq 2^N$ the set of subsets $A \subseteq N$ such that, for each $x \in A$, x is t-recognized in A. We define the *t-recognizing operator* $\Lambda_t : 2_t^N \rightarrow 2_t^N$ as follows: (i) $\Lambda_t(\emptyset) = N_c$ (ii) $\Lambda_t(A) = \{u \in N \mid \exists B \subseteq A \ s.\ t.\ |B| \geq t \wedge u \in \bigcap_{v \in B} R(v)\}$. It is easy to see that if A is in 2_t^N, then $\Lambda_t(A)$ is in 2_t^N too.

Now, we define the following sequence of sets: $\Lambda_t^0 = \Lambda_t(\emptyset)$; $\Lambda_t^k = \Lambda_t(\Lambda_t^{k-1})$, for any $k > 0$.

Lemma 1. *The operator Λ_t is monotone.*

Proof. To prove the claim of the Lemma, we have to show that given two subsets A and B of 2^N such that $A \subseteq B$, it holds that $\Lambda_t(A) \subseteq \Lambda_t(B)$, for any $t > 0$. We proceed by contradiction by assuming that there exists $x \in \Lambda_t(A)$ such that $x \notin \Lambda_t(B)$. This means that x is not t-recognized in B. But this is impossible because x is t-recognized in A (since $A \in 2_t^N$) and $A \subseteq B$. □

Lemma 2. *A set of nodes $A \in 2_t^N$ is t-closed if and only if $\Lambda_t(A) \subseteq A$.*

Proof. (*only-if-part*). Let x be belonging to $\Lambda_t(A)$. Then, there exist at least t nodes in A that recognize x. Therefore, if A is t-closed (for hypothesis), then $x \in A$ too.
(*if-part*). Let $x \in \Lambda_t(A)$. Then, there exist at least t nodes in A that recognize x. Since $\Lambda_t(A) \subseteq A$, $x \in A$ too. Then A is t-closed. □

We are ready to state the following result, which gives us an operational way to compute N_b^t.

Theorem 1. *The infinite application of the operator Λ_t, denoted by Λ_t^∞, is a fixpoint (i.e., $\Lambda^t(\Lambda_t^\infty) = \Lambda_t^\infty$), it is the least fixpoint, and this corresponds to the set of t-trusted nodes N_b^t.*

Proof. The proof of the first part of the theorem (i.e., the existence of the least fixpoint) directly derives from Lemma 1 and the theory on fixpoints (i.e., Tarski's theory).

Now, we have to prove that $\Lambda_t^\infty = N_b^t$. First, we prove that $N_b^t \subseteq \Lambda_t^\infty$. By Lemma 2, as N_b^t is t-closed, $\Lambda_t(N_b^t) \subseteq N_b^t$. If $\Lambda_t(N_b^t) = N_b^t$, then it is a fixpoint and then it is the least fixpoint Λ_t^∞. Thus, $N_b^t = \Lambda_t^\infty$, and the claim is proven. If $\Lambda_t(N_b^t) \subset N_b^t$, then, due to Lemma 1, $\Lambda_t(\Lambda_t(N_b^t)) \subseteq \Lambda_t(N_b^t)$. By Lemma 2, this implies that $\Lambda_t(N_b^t)$ is t-closed. But this is not possible because $\Lambda_t(N_b^t) \subset N_b^t$ and considering that N_b^t is the bottom of the lattice. This part of the proof is then completed.

Now we have to prove that $N_b^t \subseteq \Lambda_t^\infty$. Since Λ_t^∞ is a fixpoint, then $\Lambda_t(\Lambda_t^\infty) = \Lambda_t^\infty$. Therefore, by Lemma 2, Λ_t^∞ is t-closed and this implies that $N_b^t \subseteq \Lambda_t^\infty$. The proof is then concluded. □

The above notion of t-trustworthiness embeds a lossless propagation of trust in which the level of assurance of identity based on the recognition of users does not degrade if the t redundancy property holds at every step of propagation. In other words, the t-redundancy property is considered a threshold to propagate the trust. The t-redundancy parameter implicitly represents the assumption that the multiple identifications of a node u done by nodes in turn identified with the same trust level, and so on, until t certified nodes are reached, can be considered sufficient to trust the identity of u. The approach applies the concept of trust chain used in the context of digital certification to the domain of identity management in social networks, with the aim to counter the problem of fake identities. It is worth remarking that the model cannot provide absolute guarantees but only a trust level directly connected with the value t. The higher t, the higher the trust in identities.

So far, the trust model assumes that once a user has recognized another user, no revision of this information must be done. This assumption would be valid only in the absence of attacks that give the attacker access to the user profile (even temporarily). So we assume a sort of *safe state* regarding fraudulent accesses. In other words, the trust model above prevents the risk of fake profiles and fake identities but not from fraudulent access to legitimate profiles.

To counter this further case, we introduce a biometric-based reinforcement and combine it with the above trust-chain mechanism, to decrease the trust in a given subject if the biometric trait is not recognizable. Therefore, we can also manage non-safe states. The full trust in our mechanism is obtained by relying on the assumption that the disclosure of trusted real-life identities prevents the misbehavior of users in the trust mechanism itself, under the t-redundancy assumption.

But, if the operating user is not the legitimate one, then the above assumption fails. Thus, the identity of those users whose trust is based on paths involving the attacked profile should be not fully trusted. In other words, to take into account this aspect, we have to enable a gradual level of trust, from 0 to 1 (while before the trust was basically either 0 or 1), and use an ϵ-approximation approach to trust identities. The first step is to modify the notion of t-recognized. Obviously, we keep the redundancy parameter t in the new definition, but we introduce the possibility that a user is not fully identified in a given moment, due to the fact that the biometric support is giving a warning rate. We require that nodes in N_c (i.e., certified nodes) lose their state if the biometric support gives a warning rate. Thus, we can assume that certified nodes are not attacked.

Given a node u, we define the set $R^\epsilon(u)$ (where $0 \leq \epsilon \leq 1$) as the set of pairs $\langle v, b_r(v) \rangle$ such that $v \in R(u)$ (i.e., v is a node recognized by u in the safe state) and $1-\epsilon \leq b_r(v) \leq 1$ is the current biometric rate (i.e., the score computed earlier normalized from 0 to 1), provided that it is higher than a given threshold $0 < 1-\epsilon \leq 1$ under which v must be currently considered not recognized. Obviously, when $\epsilon = 0$ we fall in the safe state. We say that nodes in $R^\epsilon(u)$ are *ϵ-recognized* by u. At this point, we are ready to extend the notion of t-recognized to a non-safe state.

Definition 5. *We say that a node $u \in N$ is $\langle t, \epsilon, r \rangle$-*recognized *(in N) if either: (1) $u \in N_c$ (i.e., is a certified node), or (2) there exists a set B of $\langle t, \epsilon, r \rangle$-recognized nodes such that both (i) $u \notin B$, (ii) $|B| \geq t$, (iii) $u \in R^\epsilon(v)$, for each $v \in B$, (iv) $1-\epsilon \leq r \leq 1$, and (iv) $\frac{\sum_{v \in R} b_r(v)}{|B|} \geq r$.*

It is easy to see that a node is t-recognized, according to Definition 1, if and only if it is $\langle t, 0, 1 \rangle$-recognized. The intended meaning of Definition 5 is to take into account warnings triggered by the biometric support (through the parameter ϵ), and, at the same time, to require by means of the parameter r that a possible fault of trust introduced by ϵ can be partially recovered by fortifying redundancy to reduce approximation. In words, if we can trust fewer nodes because we are not sure they are not attacked we need a larger set of witnesses to reach a safe conclusion anyway. This means that r modulates the level of assurance of trust, so that the higher r, the higher the trust in the identity of $\langle t, \epsilon, r \rangle$-recognized nodes. Actually, to talk about trust we have to avoid a mutual self-sustained cluster of $\langle t, \epsilon, r \rangle$-recognized nodes, so we have to proceed as in the safe state above by requiring the minimality condition. For brevity, we do not give all detail, but it is rather clear that definitions of N_b^t and, consequently, of t-trustworthiness of a node, can be easily extended to the non-safe case, on the basis of Definition 5. We reach thus the definition of $N_b^{\langle t,\epsilon,r\rangle}$ as the bottom of the semi-lattice of subsets of $\langle t,\epsilon,r\rangle$-closed nodes of N. Therefore, a node is $\langle t,\epsilon,r\rangle$-trusted if belongs to the set $N_b^{\langle t,\epsilon,r\rangle}$. Also the definition of *recognizing operator* can be trivially extended so obtaining the operator $\Lambda_{\langle t,\epsilon,r\rangle}$ in such a way that $N_b^{\langle t,\epsilon,r\rangle}$ coincides with the fixpoint $\Lambda^{\infty}_{\langle t,\epsilon,r\rangle}$ of such operator.

6 Decentralized Trust Computation

We present here a decentralized way to compute the trust level that does not require any central authority for trust management. Obviously, standard TTPs, such as certification authorities are required to certify root profiles, but they are not involved in the propagation trust mechanism. First, we assume the collaborative behavior of participants in the network. This assumption is reasonable because every participant can benefit from the trust system so constructed.

We propose a distributed algorithm for trust computation, in which each node sends, upon query, a proof that it is t-recognized, where t is the chosen trust level. In the scope of this paper, we choose the proof to be a certificate graph, a generalization of certificate chains, in which each node is certified t times.

We assume that each node owns its pair of asymmetric cryptographic keys and that the private key is kept secret. The public key is used also as the user's identifier.

Now, we define the (recursive) structure of a certificate. We have two kinds of certificates:

- *intermediate certificates*: containing the certified node public key and the certifier public key;
- *final certificates*: containing explicitly the certified node identity (e.g., URL) and public key, and the certifier's public key.

These certificates prove that the certifier node trusts the certified node (we highlight that only the certifier node is able to issue such certificates because they are signed). Intermediate certificates enable the certified node to certify other nodes without revealing its real identity in the certificate graph, and the final certificates enable the certified node to prove that it is t-confirmed by sending a certificate graph and its final certificates.

Specifically, a user's certificate is composed of:

1) *issuer*: this field contains the public key of the generator of the certificate;
2) *target*: this is the public key of the user to be certified;
3) *profile*: this field is obtained as $url \oplus r$, where url is the URL of the social network profile of the target, r is a randomly generated bit string, and $\oplus$ denotes the exclusive OR operator;
4) *key*, an optional field containing r (i.e., the value generated above). If this attribute is missing, then this certification is said *intermediate certificate*; otherwise, it is said *final certificate*;
5) *certifier*: this field may contain a (possibly empty) set of intermediate certificates of the users who previously have certified the issuer. If this field is empty, then the issuer should be a root node certifier and this target user is said *root node*.
6) *expiration date*: this field may contain the date until the certificate can be considered valid.
7) *value*: this field contains a value between 0 and 1; it is 0 when the certificate is revoked (for any reason), it is equal to 1 when the trust in the certificate is maximum, and a value between o and 1 in all the other cases.

Furthermore, the certificate is signed by the issuer's private key to guarantee information integrity.

In practice, when a user u wants to be t-recognized, u asks their social friends $u_1,\ldots,u_t$ for a final certificate. In this example, we assume that $u_1,\ldots,u_t$ are already t-recognized.

Now, each friend u_i of u signs and issues a certificate c_i with all fields filled-in (i.e., containing also the random r). At the end of this step, u has got t final certificates: if all of them are valid, then u becomes t-recognized as well. At this point, u can propagate the trust to another friend v, by issuing and signing a new certificate of v, in which the field *certifier* is composed of the certificates c_i^* with $1 \leq i \leq t$, where c_i^* is obtained from c_i by erasing the field *key*.

Concerning the validation of certificates, we have two cases.

An intermediate certificate is valid if:

- it is signed by the issuer's private key (i.e., the signature is valid);
- the issuer is a root node certifier (if the field certifier is empty); the field certifier contains at least t (recursively) valid certificates (otherwise);

A final certificate is valid if, in addition to the above conditions, it holds also that the value of the field profile $\oplus$ the value of the field key is equal to the URL of the social-network profile of the user being certified.

As introduced by Definition 5, our model takes into account also the possibility that the level of trust of a specific user can be gradually reduced if the biometric recognition is not full. Therefore, it may happen that an already t-recognized user's account becomes potentially compromised. In this case, the associated level of trust decreases. Furthermore, since our model is based on the concept of collaboration and propagation, when a certified user is compromised also all the paths involving the attacked profile should take into consideration this information and decrease the overall level of trust. According to Definition 5, the trust level t does not act just as an *on-off* switch, but it can assume values between $[0,1]$, so that if one node has been compromised, then the overall computation of each trust level in which that node is involved dynamically changes.

In particular, if we set $t = 10$ to be considered t-recognized and we consider an ideal world in which every peer has a $t\_level = 1$, we can state that if ten people sign a certificate directed to a given user, then it would be sufficient for this user to become t-recognized. Otherwise, in a real-world scenario, it would happen that more people would be needed to reach the same trust level because some participants could have a $t\_level \leq 1$.

Optimization.

Our method may suffer in terms of storage size because duplicate certificates are saved in the chain of certificates. For this reason, we provide an optimized version of our approach in which the certificate graph does not contain duplicate certificates.

Observe that when a node receives a (trust) certificate, the corresponding intermediate certificate is added to the set of intermediate certificates and the final certificate to the set of final certificates. The other certificates included in the received certificate graph are added to a third internal structure, consisting of a mapping in which each element key is the pair (*certified public key, certifier public key*), thus ensuring the uniqueness of each certificate. When issuing a new certificate, the node sends the status of the certificate along with the certificate itself.

The resulting certificate graph is represented by a multi-map structure whose index is the certified public key. For each index, the number of certificates and the certificates are verified. Then, starting from the final certificates, the absence of loops is ensured by a depth-first search. If a visited node already belongs to the current path, then a loop is detected.

7 System Functions

In this section, we provide the basic functions of our trust model, which would be the basis for any social network platform implementing our approach.

- *Root registration*: this function is used by a node to become a *root node*.
- *Set Biometric Model*: it is invoked whenever two users establish a contact in the social network in the set-up phase. It is also invoked by the Root registration function to allow the social network provider to keep the biometric model of the root node.
- *Trust*: this function is invoked whenever a node wants to sign a certificate to another participant of the network;
- *Verify trust*: this operation is invoked whenever a node wants to verify that a given participant belongs to the set of certified nodes;
- *Revoke*: it is invoked whenever a participant wants to revoke a past certificate previously given to a certain node. This can happen for any reason.

We describe such functions and their complexity over time and space w.r.t. the number of nodes *n* and the trust level t.

Root Registration.

This operation involves a Trusted Third Party (TTP) and a social network profile that claims to become a root node. In particular, the TTP receives the request from a social profile and it starts an authentication process to demonstrate the association between the real-life identity and the social network profile. This process could use a public digital identity system. After being authenticated, the social profile becomes a root node, representing the first node of a new trust chain.

The complexity of this operation is constant over time and space.

Set Biometric Model.

This function is invoked every time it is necessary to model the biometric behavior of a social profile. In particular, the process of capturing keystroke dynamics consists of extracting different features, such as the time interval between pressing and releasing two characters on the keyboard, the code of the key, etc.

The complexity of this operation is constant over time and space.

Trust.

This operation can be carried out after a real-life interaction between the two users and it can be invoked only by those peers that have already been t-recognized and/or by root nodes. Such an operation is used to trust another profile by means of a signed certificate. In detail, it accepts a certificate as input and it can be invoked by every node belonging to the set of certified profiles N_c. In particular, two different certificates are signed: an *intermediate* certificate, which does not contain any identity-related information, and a *final* certificate, which contains also the certified user's identity.

The complexity of this operation is constant over time and space.

Verify trust.

It is called whenever a node wants to check the value of trust associated with a given social profile. In particular, the function accepts a social profile and a required trust level t and returns whether the social profile can be considered trusted or not.

To carry on the operation, the social profile being verified has to reveal its final certificates so that the node can check the validity of the certificate chain.

In Figure 2, we show an architecture in which node A becomes t-recognized by nodes C and B, with $t = 2$. Note that intermediate certificates do not include the identity-related information about the destination of such certificates, while final certificates contain the URI of the destination node because intermediates are used by a certified node to give trust in other peers, while finals are used in this verification process, where it is necessary to reveal the link between the certificate and the corresponding social identity. We can conclude that this phase succeeds when the sum of values included in final certificates is greater than or equal to the trust level t set by the user.

As for the spatial complexity, we note that the size of the certificate chain associated with each t-recognized social profile is $\sum_{i=1}^{d} t^i$, where d is the hop number between the social profile taken into consideration and the root nodes. Considering the example in Figure 2, d is equal to 2 because the social profile A to be verified is linked to the three root nodes D, E, F via the following paths: $A \to C \to F$, $A \to C \to D$, $A \to B \to D$, and $A \to C \to E$. Thus, the spatial complexity can be approximated to t^d, which is polynomial thanks to the small-world phenomenon and the six-degree separation law [5]. In particular, in social networks, the average degree of separation between two peers is found to be 3.43 [6]. Following the same reasoning seen for the spatial complexity, we obtain that the time complexity is t^d.

Revoke.

This task represents an important function of our solution. As is the case in most certification infrastructures, a certificate-revocation-list approach is used. The list of revoked certificates is maintained by the same TTP introduced in the root selection phase. This method is invoked every time we require that a certificate previously signed that is still valid (the scheduled expiration date has not come yet) is no longer considered trusted. Each entry of the certificate revocation list is associated with the identifier of the certificate and a timestamp. The node that invokes this function has to sign the certificate revocation to prevent malicious requests. Following the same idea of trust propagation, every time a certificate is found to be revoked, child branches of such certificate are no longer considered valid. Our model provides for two different ways: (1) an *explicit* revocation, for which any standard solution (e.g., the TLS approach) can be adopted to revoke the certificate before the expiration date, and (2) an *implicit* solution, denoted by setting the value associated with the certificate equal to zero.

Both time and spatial complexities are constant since this function does not involve either t or n.

As a final observation of this section, we highlight that our tree-like certification process might apparently represent an infeasible source of exponential growth. For example, whenever a certificate is revoked, all branches that are generated from that certificate are cut off and considered revoked as well. However, the real-life structure of online social networks (i.e., the small diameter and path lengths due to the small-world principle) makes this risk not effective.

8 Security Analysis

In this section, we discuss the security requirements presented earlier and explain how we provide these security properties and how our model prevents reputation attacks. We start by defining the threat model adopted in our approach. We focus our attention on the robustness of the trust mechanism, by assuming that the biometric system is secure.

The adversary is any user of the social network. We allow collusion among users, by assuming that the maximum number of colluding adversaries is less than the fixed trust level t.

Specifically, the adversary can perform the following attacks against the trust mechanism:

- *Sybil attack:* The adversary tries to generate multiple fake identities to jeopardize the system.
- *Slandering:* The adversary tries to fraudulently reduce the trust of a victim user.
- *Self Promotion:* The adversary tries to fraudulently increase their trust.
- *Whitewashing:* The adversary tries to fraudulently erase the reduction of trust.
- *Social engineering:* The adversary tries to compromise the trust mechanism by using social engineering attacks against other users.

Collusion and Sybil attacks are strongly mitigated because, as highlighted in the theoretical model, every participant must be certified to propagate trust and every certification chain must lead back to a root node. *Slandering attacks* are prevented, since it is impossible to defame someone because our model provides, by construction, only trust certificates and there is no way to vilify another participant. Furthermore, the solution we propose makes it impossible for a peer to *self-promote* because there is the need for the collaboration of the community to reach a certain trust level. The resilience of our approach against this type of attack is studied in Section 9.3.2.

Whitewashing attacks are mitigated because if a participant creates a new account to start with a new reputation, they will lose their certificates and it is necessary to execute the whole process of our algorithm to reach a certain trust level.

Our approach is robust against *social engineering attacks* because of the reinforcement given by adding biometric features to the model, which enable continuous authentication for every participant. Indeed, even if an attacker gains full access to the victim's social profile, our approach will discover the attack thanks to the continuous authentication provided by the biometric features.

Apart from the resistance to the previous attacks, we highlight that our solution achieves the following security properties.

Integrity is ensured because the attributes of certificates are signed by the issuer's private key. For the same reason, *Accountability* is provided as well. *Availability* is reached because certificates are computed once and then they are stored so that the computation of the t-recognized nodes is possible also in the case in which nodes are for any reason unavailable.

Anonymity is achieved in pseudonymous form because our model uses intermediate certificates to propagate trust. In these certificates, the real identity of the actors involved in the process is not revealed. The list of revoked certificates is stored and maintained by the TTP which is in charge also to carrying on the root registration step.

9 Experiments

In this section, we validate our proposal. First, we present the dataset used and discuss how biometrics is managed. Then, we analyze the results of the experiments we carried out.

9.1 Datasets

For the validation of the proposal, we used five real-life datasets downloaded from the repository `http://networkrepository.com/` [35]. Each dataset provides a list of edges between two nodes, which represent that the two nodes are friends.

In the Facebook section of this repository, several datasets are provided, each sampling a certain portion of the network, with different characteristics and properties. Among them, we selected five different data sources so that we can cover heterogeneous and different contexts.

In Table 1, we report the characteristics of our datasets, where $|N|$ is the number of nodes, $|E|$ is the number of edges, d_{max} is the maximum degree of a node in the given dataset, and d_{avg} is the average degree of the nodes. By analyzing such values, we can say that $D1$ and $D2$ are the most connected datasets because they have high degrees and they have many edges. $D3$ and $D4$ have the highest number of nodes but fewer edges and lower degrees, meaning that they are less connected and more scattered. Instead, $D5$ is the trade-off.

As for the biometric part, we used a biometric benchmark database composed of keystroke dynamics [23]. It is composed of a biometric template from 110 individuals

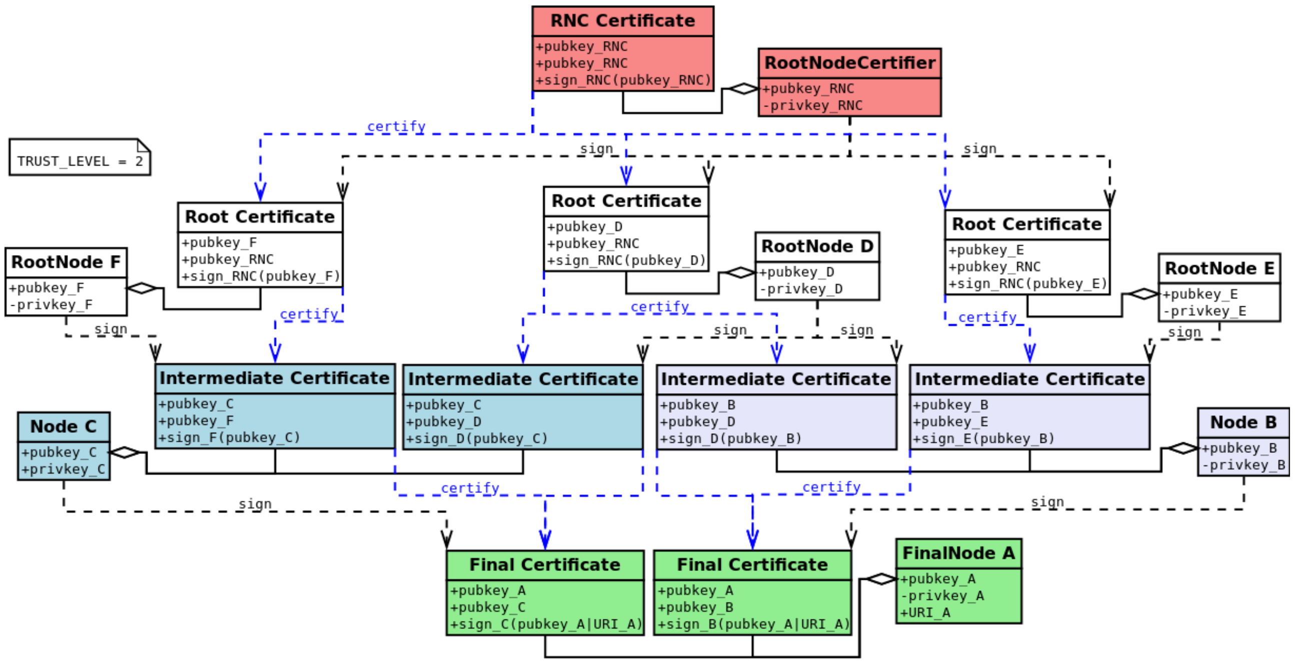

Fig. 2. Example of certificates with trust level = 2

TABLE 1
Dataset characteristics.

	$\|N\|$	$\|E\|$	d_{max}	d_{avg}
$D1$	24k	1M	3k	96
$D2$	36k	2M	5k	87
$D3$	64k	1M	2K	39
$D4$	63k	817k	1k	25
$D5$	42k	1M	4k	65

typing on two desktop keyboards (a French keyboard for users in France and a Norwegian keyboard for users in Norway) *i.e.* AZERTY and QWERTY (this is not a classical QWERTY keyboard, however, we do not use specific Norwegian keys), respectively. An entry is composed of the typing of 5 passwords. Several studies tested that no significant difference between a laptop and a USB keyboard exists [17]. For this dataset, we considered 14 digraphs: latency of ‘ca’, ‘ic’, ‘ed’, ‘he’, ‘pe’, ‘te’, ‘ch’, ‘li’, ‘ri’, ‘ll’, ‘on’, ‘er’, ‘es’ and ‘st’. The size of the biometric template is 14. As an entry contains several times each digraph, the template is computed as the mean latency of each 14 digraphs. The reference template is computed by merging several entries (here 14 entries).

9.2 Managing biometrics

For managing biometrics features, we used the dataset described in the previous section and generated legitimate scores by comparing the reference template with templates from the same user. In our attack model, we simulated impostors by replacing the reference template of the victim user with templates coming from other users.

Since the social network graphs we have derived from edge lists contain way more users than the keystroke dynamics dataset, we cannot associate a unique keystroke dynamic with each social network user. User biometrics behavior is thus simulated by randomly picking, for each social network user, a user from the keystroke dynamics datasets. Attacks are simulated by using different keystroke dynamics from the keystroke dynamic dataset. For the BioHashing verification, a different random secret is used for each account and each of its certifiers, and the secret is assumed known by the attacker if the account has been attacked. It is worth noting that the modality can be easily collected by any website, in particular phishing websites. To limit such attacks, Keystroke Dynamics Anonymisation Schemes could be used.

The performances of such systems could be improved using other modalities (s.a. the mouse), that need to enable continuous authentication to not be vulnerable to some attacks, (e.g., lunchtime attacks). The usage of soft biometrics could improve authentication performances and also verify the consistency with the claimed profile (gender, age, ...) and template update techniques could also be implemented to improve the overall system performances as the users’ biometrics behavior changes over time.

Biometrics can be used to automatically revoke certifications. However, such revocation could be manually performed by the certifier in response to a flag raised by the biometrics algorithm. This enables the certifier to perform complementary verification before choosing to revoke the certification.

9.3 Experimental results

To validate our proposal, we implemented a simulator in Python 3.11 and ran the experiments on a PC with 16GB of RAM and an $i7$ Intel processor running $Ubuntu$ as the operating system.

Datasets are formatted as edge lists, which is a particular data structure used to represent a graph as a list of its edges. For example, if between nodes a and b there exists a connection (an edge), then in the edge list we will find a row containing the pair (a, b).

Thus, we read the edge lists in such a way that we can deduce all the information about nodes and their relationships. After this operation, we proceed by selecting root nodes. We could have implemented many different algorithms for this selection (for instance, we could have used different centrality measures to select the best nodes). Anyway, we decided to select randomly roots among all the nodes since it would be fairer and more representative of a real situation. In fact, if we had chosen the best nodes by following some measure of centrality, the results would have been somewhat biased and far from a realistic scenario. The only constraint we require in the root nodes selection step is that a node, to be chosen, should have a number of relationships at least equal to `t_level` indicated in the simulation.

Since we used `t_levels` that are in the order of magnitude of ten, we can conclude that this condition is acceptable because most real social profiles easily reach ten friendships. Otherwise, there would have been chosen some root nodes unable to propagate the trust in the network, acting as a bottleneck.

We remark that experiments do not involve the computation of biometric features, as experiments are devoted to validating the trust-based approach. As it is clear from the definition of the model, the biometric component may just change the number of equivalent profiles needed to trust a given profile. Therefore, it can be viewed as an orthogonal component. On the other hand, the effectiveness of biometric-based mechanisms considered in this paper has been shown in the literature.

9.3.1 System performance

In the first experiment, we want to study the performance of our approach when no attack is performed and when participants are always collaborative in trusting a friend node. A relevant value we measure is the *coverage* that could be reached by running our algorithm in the network. Since we use five datasets with different properties, we expect that some differences will be mirrored in the results. Specifically, the more the network is dense and connected, the higher we expect the coverage will be.

We carry out our experiments by varying the following parameters:

- t_level;
- $number\_of\_roots$.

In particular, we use $t\_level = \{3, 6, 8, 10\}$ and $number\_of\_roots = \{0.1\%, 1\%, 2\%, 5\%, 10\%\}$ (these percentages are calculated with respect to the total number of nodes of the given dataset). We run every combination of these two parameters fifty times and we average the results to avoid outlier cases.

In Tables 2, 3, 4, 5, and 6, we report the coverage results of these simulations for each dataset.

As expected, the more connected the datasets, the higher the coverage rates. It is interesting to underline that, in three out of five, the coverage reached is quite relevant with only 0.5% of roots chosen ($D1, D2$, and $D5$) and only in those two datasets less connected ($D3$ and $D4$) this little number of roots is not able to reach a significant number of nodes. By analyzing those two datasets, we see that the average degree of a node is quite low (39 and 25, resp.). This means that every node has, on average, only thirty-nine and twenty-five friends.

We remark that these results have been obtained by running only one cycle of the trust operation (i.e., every node asks to be certificated only once). In a real context, instead, a participant can ask many times a friend to be recognized, so, in this sense, we can conclude that our experiment underestimates the real potential coverage.

TABLE 2
Coverage rate in $D1$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	95,9	90	83,9	72,3
roots = 1%	96,5	91,7	86,7	80,5
roots = 2%	96,9	92,6	89	85,1
roots = 5%	97,2	93,5	90,8	88,1
roots = 10%	97,5	94,5	92,1	89,9

TABLE 3
Coverage rate in $D2$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	95,5	87,5	80,2	71,6
roots = 1%	96	90	84,7	77,3
roots = 2%	96,2	91,29	87,2	82,6
roots = 5%	96,5	93,5	89,19	85,5
roots = 10%	97	94,5	90,6	87,7

TABLE 4
Coverage rate in $D3$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	69,9	44,4	6,6	4,6
roots = 1%	70,5	47	35,2	23,7
roots = 2%	71	49,4	38,9	29,6
roots = 5%	72	52	43,4	36,7
roots = 10%	73,5	55,1	47	41

TABLE 5
Coverage rate in $D4$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	70,1	44,7	8,9	5,7
roots = 1%	71,2	47,1	34,4	20,2
roots = 2%	72	49,3	38,2	29,9
roots = 5%	73	52,4	43,5	36,4
roots = 10%	74,5	55,7	47,8	41,8

TABLE 6
Coverage rate in $D5$ vs. t_level and number of roots.

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
roots = 0.5%	93,1	82,8	75,1	64,4
roots = 1%	93,8	84,3	77,4	68,9
roots = 2%	94,3	86	79,7	72,7
roots = 5%	94,9	87,9	83,2	77,8
roots = 10%	95,4	89,5	85,6	81,4

9.3.2 *Resilience of the system*

In this section, we study the resilience of our proposal to the *slandering attack* (described in Section 8). Under this attack, a certain amount of root nodes become malicious and do not play anymore a collaborative and positive role in the propagation of the trust. To better evaluate the results of this experiment, we compare the performance of our proposal with that of [30] (presented in Section 2, where the parameter $\alpha = 1$ because the used datasets allow us to identify positive and negative interactions. Specifically, under attack, malicious nodes propagate negative interactions. Under attack, we randomly picked 30% of the roots as malicious nodes. As evaluation metrics, we use *coverage loss* defined as $1 - \frac{\hat{T}_a}{\hat{T}}$, where $\hat{T}_a$ is the average trust of nodes under attack and $\hat{T}$ is the average trust of nodes when no attack is carried out.

We varied the same values as the first experiment (t_level and $number\_of\_roots$) and, in Table 7, we report the percentage of coverage loss when varying the percentage of root nodes $(0.5, 1, 2, 5, 10)$ and, for the sake of presentation, we averaged the results. In Table 8, we report the percentage of coverage loss measured for the *Strust* model [30] with the same experiment setup.

From the analysis of these experiments, we see that a *slandering attack carried out by the* 30% *of root nodes slightly reduces the performance of our proposal (on average, about 7%). Comparing these results with that of* [30], we can conclude that our approach contrasts *slandering attacks*.

TABLE 7
Trust loss measured for our proposal (in percentage)

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
D1	0,64	0,96	9,09	12,08
D2	0,71	1,37	1,69	13,88
D3	1,92	3,07	12,56	21,30
D4	2,04	4,32	10,48	16,40
D5	0,80	1,10	2,21	22,30

TABLE 8
Trust loss measured for *Strust* [30] (in percentage).

	$t = 3$	$t = 6$	$t = 8$	$t = 10$
D1	29,41	26,52	32,40	33,64
D2	31,22	35,36	36,80	37,72
D3	9,98	18,36	18,67	22,74
D4	13,04	13,98	22,73	27,53
D5	35,49	39,84	40,01	41,92

10 Conclusion and perspectives

Trusting identities in social networks is an important challenge in the literature. The approach we designed for this purpose exploits a collaboration among social network users and a keystroke-dynamics-based technique. In our proposal, each user shares with the other users the information about the neighborhoods. This information is propagated among the network in a trusted way. We provide the theoretical characterization of the proposed approach and its validation by experiments carried out on real-life data. The experiment results show that trusted information reaches most of the nodes (depending on some system parameters). The novelty of our proposal is to distribute, in a decentralized fashion, over the social network itself, the trust mechanism. Moreover, we embed in the trust mechanism behavioral biometric features, to better support the maintenance of trust values and then isolate possible attacked profiles. We remark that the paper addresses the biometric component of the model in a concrete and definite way, by identifying which type of biometric data is suitable to our purpose, and by defining how the biometric component affects the trust values. Therefore, we argued the effectiveness of the trust mechanism is the most meaningful aspect to assess. Regarding this aspect, it is worth noting that the exchanging of messages from a social network user to another profile, in principle, is a reason for characterizing that profile as trusted provided that at least one of the two profiles is real. This makes recursive the problem and that needs further research. As a future work, we plan to implement the model in a *homemade* online social network and experiment it in a real-life setting, possibly in a research project reaching large communities and involving industrial partners. This will allow us to collect real biometric data for proof of concept validation.

References

[1] Abdul-Rahman A (1997) The pgp trust model. In: *EDI-Forum: the Journal of Electronic Commerce*, volume 10. pp. 27–31.
[2] Ahmad, N., Laplante, P. A., DeFranco, J. F., and Kassab, M. (2021). A cybersecurity educated community. *IEEE Transactions on Emerging Topics in Computing*. 10(3), 1456-1463.
[3] Bhana B and Flowerday S (2020) Passphrase and keystroke dynamics authentication: Usable security. *Computers & Security* 96: 101925.
[4] Buccafurri F, Lax G and Migdal D and Nicolazzo S and Nocera A and Rosenberger C (2017) Contrasting false identities in social networks by trust chains and biometric reinforcement. In: *2017 International Conference on Cyberworlds (CW)* IEEE, pp. 17–24.
[5] Watts, Duncan J (1999) Networks, dynamics, and the small-world phenomenon. *American Journal of sociology* 105(2): 493–527.
[6] Bakhshandeh, R and Samadi, M and Azimifar, Z and Schaeffer, J (2011) Degrees of separation in social networks. In: *Proceedings of the International Symposium on Combinatorial Search* 2(1): 18–23.
[7] Buccafurri F, Musarella L and Nardone R (2019) Enabling propagation in web of trust by ethereum. In: *Proceedings of the 23rd International Database Applications & Engineering Symposium*. pp. 1–6.
[8] Choi K, Toh KA and Byun H (2012) Incremental face recognition for large-scale social network services. *Pattern Recognition* 45(8): 2868–2883.
[9] Conti M, Poovendran R and Secchiero M (2012) Fakebook: Detecting fake profiles in on-line social networks. In: *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*. IEEE, pp. 1071–1078.
[10] Cutillo LA, Molva R and Strufe T (2009) Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine* 47(12).

[11] Dafer M and El-Abed M (2017) Evaluation of keystroke dynamics authentication systems: Analysis of physical and touch screen keyboards. In: *Developing Next-Generation Countermeasures for Homeland Security Threat Prevention*. IGI Global, pp. 306–329.
[12] Deutschmann I, Nordström P and Nilsson L (2013) Continuous authentication using behavioral biometrics. *IT Professional* 15(4): 12–15.
[13] Yasyn Elyusufi, Zakaria Elyusufi and M'hamed Ait Kbir (2019) Social networks fake profiles detection using machine learning algorithms. In: *3rd International Conference on Smart City Applications, 2019.* Springer, pp. 30–40.
[14] Fairhurst M and Da Costa-Abreu M (2011) Using keystroke dynamics for gender identification in social network environment .
[15] Gadiya M and Jain S (2016) Gender prediction using images posted on online social networks .
[16] Gaines R, Lisowski W, Press S and Shapiro N (1980) Authentication by keystroke timing: some preliminary results. Technical report, Rand Corporation.
[17] Giot R, El-Abed M, Hemery B and Rosenberger C (2011) Unconstrained keystroke dynamics authentication with shared secret. *Computers & Security* 30(6): 427–445.
[18] Giot R, El-Abed M and Rosenberger C (2009) Greyc keystroke: a benchmark for keystroke dynamics biometric systems. In: *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*. IEEE, pp. 1–6.
[19] Guo B, Ding Y, Sun Y, Ma S, Li k annd Yu Zhiwen (2021) The mass, fake news, and cognition security *Frontiers of Computer Science* 15(3): 1–13.
[20] Haenni R and Jonczy J (2007) A New Approach to PGP's Web of Trust. In: *EEMA'07, European e-Identity Conference*.
[21] Hamzelou N, Ashtiani M and Sadeghi R (2021) A propagation trust model in social networks based on the a* algorithm and multi-criteria decision making. *Computing* 103(5): 827–867.
[22] Hocquet S, Ramel JY and Cardot H (2007) User classification for keystroke dynamics authentication. In: *The Sixth International Conference on Biometrics (ICB2007)*. pp. 531–539.
[23] Idrus SZS, Cherrier E, Rosenberger C and Bours P (2013) Soft biometrics database: A benchmark for keystroke dynamics biometric systems. In: *Biometrics Special Interest Group (BIOSIG), 2013 international conference of the*. IEEE, pp. 1–8.
[24] Wahab, A. A., Hou, D., Banavar, M., Schuckers, S., Eaton, K., Baldwin, J., Wright, R. (2022, April). Shared multi-keyboard and bilingual datasets to support keystroke dynamics research. Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy (pp. 236-241).
[25] Jin ATB, Ling DNC and Goh A (2004) Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition* 37(11): 2245–2255.
[26] Khan A, Brohman K and Addas S (2021) The anatomy of 'fake news': Studying false messages as digital objects. *Journal of Information Technology* : 02683962211037693.
[27] Matthiesen JJ and Brefeld U (2020) Assessing user behavior by mouse movements. In: *International Conference on Human-Computer Interaction*. Springer, pp. 68–75.
[28] Mondal S and Bours P (2013) Continuous authentication using behavioural biometrics. In: *Collaborative European Research Conference (CERC'13)*. pp. 130–140.
[29] Mondal S and Bours P (2017) Person identification by keystroke dynamics using pairwise user coupling. *IEEE Transactions on Information Forensics and Security* 12(6): 1319–1329.
[30] Nepal S, Bista SK and Paris C (2015) Behavior-based propagation of trust in social networks with restricted and anonymous participation. *Computational Intelligence* 31(4): 642–668.
[31] Patel V, Ratha NK and Chellappa R (2015) Cancelable biometrics: A review. *IEEE Signal Processing Magazine* : 54–65.
[32] Peng S, Yang A, Cao L, Yu S and Xie D (2017) Social influence modeling using information theory in mobile social networks. *Information Sciences* 379: 146–159.
[33] Ramalingam D and Chinnaiah V (2018) Fake profile detection techniques in large-scale online social networks: A comprehensive review. *Computers & Electrical Engineering* 65: 165–177.
[34] Tsimperidis, I., Katos, V. (2013, September). Keystroke forensics: are you typing on a desktop or a laptop?. Proceedings of the 6th Balkan Conference in Informatics (pp. 89-94).
[35] Rossi RA and Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. In: *AAAI*. http://networkrepository.com.
[36] Sherchan W, Nepal S and Paris C (2013) A survey of trust in social networks. *ACM Computing Surveys (CSUR)* 45(4): 47.
[37] Shetty NP, Muniyal B, Anand A and Kumar S (2022) An enhanced sybil guard to detect bots in online social networks. *Journal of Cyber Security and Mobility* : 105–126.
[38] Song X, Zhao P, Wang M and Yan C (2016) A continuous identity verification method based on free-text keystroke dynamics. In: *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE, pp. 000206–000210.
[39] Tiwari V (2017) Analysis and detection of fake profile over social network. In: *Computing, Communication and Automation (ICCCA), 2017 International Conference on*. IEEE, pp. 175–179.
[40] Tsimperidis G, Katos V and Rostami S (2017) Age detection through keystroke dynamics from user authentication failures. *International Journal of Digital Crime and Forensics (IJDCF)* 9(1): 1–16.
[41] Upadhyaya SJ (2017) Continuous authentication using behavioral biometrics. In: *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*. ACM, pp. 29–29.
[42] Wanda P and Jie HJ (2020) Deepprofile: Finding fake profile in online social network using dynamic cnn. *Journal of Information Security and Applications* 52: 102465.
[43] Wu J, Xiong R and Chiclana F (2016) Uninorm trust propagation and aggregation methods for group decision making in social network with four tuple information. *Knowledge-Based Systems* 96: 29–39.
[44] Xu X, Yu J, Chen Y, Hua Q, Zhu Y, Chen YC and Li M (2020) Touchpass: towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations. In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. pp. 1–13.
[45] Yang Y, Guo B, Wang Z, Li M, Yu Z and Zhou X (2019) Behavesense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics. *Ad Hoc Networks* 84: 9–18.

Francesco Buccafurri is a full professor of computer science at the University Mediterranea of Reggio Calabria, Italy. In 1995 he took the Ph.D. degree in CS at the University of Calabria. In 1996 he was visiting researcher at Vienna University of Technology. His research interests include cybersecurity, privacy, social networks, e-government, and P2P systems. He has published more than 160 papers in top-level journals and conference proceedings. He serves as a referee for international journals and he is a member of several conference PCs. He is Associate Editor of Information Sciences (Elsevier) and IEEE Transactions on Industrial Informatics and played the role of PC chair and PC member in many international conferences. He is member of the IEEE computer society.

Gianluca Lax is an Associate Professor of Computer Science at the University Mediterranea of Reggio Calabria, Italy. In 2005, he received his Ph.D. in computer science from the University of Calabria. In 2018, he got the habilitation as full professor of computer science. His main research interests include information security and social network analysis. He is an author of more than 140 papers published in leading international journals and conference proceedings. He serves as a referee for many international journals and is in the program committee of many conferences. He is also included in the editorial board of several international journals and participates in many funded projects.

Denis Migdal is an associate professor at the University of Clermont Auvergne (UCA) since 2021. He obtained his Ph.D. degree in Computer Science in 2019 at the University of Caen Normandy (UNICAEN). From 2016 to 2020 he was part of the Research Group in Computer Science, Image and Instrumentation of Caen (GREYC) laboratory. He is now part of the Laboratory of Informatics, Modeling and Optimization of the Systems (LIMOS) since 2021. His research focuses on Biometrics and Keystroke Dynamics.

Lorenzo Musarella is an Assistant Professor of Computer Science at the University Mediterranea of Reggio Calabria, Italy. In 2021 he received his Ph.D. in Information Engineering from the University Mediterranea of Reggio Calabria. His research interest includes cybersecurity, trust, privacy, and network analysis. He is author of 15 papers published in top-level international journals and conference proceedings.

Christophe Rosenberger obtained his Ph.D. in Information Technology from the University of Rennes 1 in 1999. His PhD thesis work was undertaken at ENSSAT in Lannion between 1996 and 1999 in the field of hyperspectral image segmentation. He joined the ENSI de Bourges school of engineering in Bourges (known as INSA Centre Val de Loire) as assistant professor in 2000. In 2007, he joined the ENSICAEN school of engineering in Caen as full professor. He belongs to the SAFE (Security, Architecture, Forensics, biomEtrics) research group in the GREYC research lab. His current work focuses on the domain of computer security, in particular research activities in biometrics (keystroke dynamics, soft biometrics, evaluation of biometric systems, fingerprint quality assessment...).