Vis-a-Vis Verification: Social Network Identity Management Through Real World Interactions

Marco Maier, Chadly Marouane
and Claudia Linnhoff-Popien
Mobile and Distributed Systems Group
Ludwig-Maximilians-University Munich, Germany
{marco.maier, chadly.marouane, linnhoff}@ifi.lmu.de

Benno Rott
VIRALITY GmbH
Munich, Germany
rott@virality.de

Stephan A. W. Verclas
T-Systems International GmbH
Darmstadt, Germany
stephan.verclas@t-systems.com

***Abstract*—Online services, particularly those aimed at a specific user base such as a company's employees, face the problem of identity management. Especially when the service constitutes some kind of social network, i.e., the validity of the users' identities matters, secure and reliable means for identity verification and authentication are required. In this paper, we propose a novel identity management concept based on a) verification through physical presence and b) authentication through ownership. Our approach being a hybrid solution between a centralized authority and decentralized trust management is settled on a sweet spot between security and convenience for the users.**

Keywords—identity management systems; authentication; social network services; mobile computing

I. INTRODUCTION

Nowadays, with about 2.8 billion people using the Internet worldwide [1] and over 1.1 billion people participating in the world's largest online social network Facebook [2], online service providers have a clear need for identity management, i.e., *administration*, *verification*, *authentication* and *authorization* of virtual identities and their real-world counterparts.

Especially when a service's users are linked to their real-world identity (i.e., the service constitutes some kind of online social network) and more so, when the service furthermore requires a high level of security, a key part of identity management is to verify that a virtual account really belongs to the real-world person it is supposed to be linked to, and to provide a secure and intuitive means of authentication. Typically in such services, a user Alice would decide for or against granting certain permissions to a virtual user Bob based on whether she wants to grant those permissions to the real-world Bob. Thus, she has to be sure that the user account really belongs to the real-world Bob (verification), and that nobody else can make requests on behalf of that account (authentication).

There are several ways of verifying a user's real-world identity, which to date either are easy to implement/use but quite easy to attack, or are reasonably secure but introduce a huge overhead in the general process of account creation. In the same way, currently used authentication procedures differ in potential for security breaches on the one, and intuitivity on the other hand.

With the now near ubiquitous usage of smartphones, we see huge potential to improve upon the currently used ways of identity verification and authentication in online services. In this work, we present an approach that is based on two key ideas

- New user accounts are verified to belong to a certain real-world identity by requiring an interaction of an existing user with the new user in the real world.
- The users employ their personal smartphone as the credential for authentication, i.e., the security token is stored on the users' smartphone.

Our approach constitutes a hybrid system. There is a central authority which is the root of the system's trust relations and is controlled by the organisation employing the system. In order to avoid the typical overhead of sophisticated identity verification, verification tasks are distributed among the system's existing users. Consequently, our system provides a high degree of trustworthiness of the user accounts while keeping the introduced overhead at a reasonable level. To the best of our knowledge, to date, no other approach has settled on that sweet spot between security and convenience/ease-of-use.

The rest of the paper is structured as follows. In Section II, we give an overview of various concepts for identity verification and authentication, together with their individual strengths and weaknesses. In Section III, we discuss related work which is or could be used similar to our approach. In Section IV, we present our system for identity verification and authentication. After that, we describe a real implementation of our concept which has been deployed for production usage (Section V). In Section VI, we describe some scenarios how our approach could be used, and in Section VII, we conclude with an outlook at future work.

II. IDENTITY MANAGEMENT

Identity management of online services comprises several sub-topics like authorization and management of user accounts. The focus of this work specifically lies on *identity verification* and *authentication*. We define identity verification as the process to check the real-world identity of a person and to connect this identity to a virtual account. Authentication then requires some kind of credential to prove that a request is made by that virtual account (i.e., on behalf of the real person).

A. Identity Verification

There are several mechanisms to verify an online identity, i.e., to link a virtual account to a real-world person. These mechanisms can be categorized into three groups, namely *verification through another online identity provider*, *verification through a second communication channel* and *verification through physical presence*.

1) Verification through another online identity provider: The idea of this mechanism is to rely on a third party to verify a new user account. The typical and most widely used example is to require an existing email address when a new account shall be created. To confirm the email address, the online service sends a message to the registrant containing a confirmation link. By clicking the link, the new user can ensure that he is the real owner of the email address. In this case, one relies upon the third party to have checked the identity of the potential user. Thus, it depends on the third party whether the real-world identity is verified, and even if so, typically the real-world identity is not handed over to other parties, leaving the online service with the email address only.

An email address of course is only a very weak personal detail for a real identity. Another approach is to rely on real identity providers. For example, online services like Facebook.com, plus.google.com, or LinkedIn.com manage user profiles which are verified to some degree. These services can be used either through proprietary interfaces (e.g., *Facebook Login* [3]), or by employing standardised mechanisms like OpenID [4].

Verification through a third party often is the most convenient method of identity verification, both for the end user and the online service provider. The main drawback is the dependence on the trustworthiness of the third party.

2) Verification through a second communication channel: Another approach is the integration of a second communication channel into the verification procedure, typically using an endpoint which requires or inherently is linked to a more sophisticated identity verification like a mobile phone number or a postal address.

When using a mobile phone number, the online service e.g., can send a randomly generated unique token as a text message to the phone. The user then has to enter that token into a form at the online service, which ensures the provider that the user really is the owner of that specific phone number.

A similar procedure can be performed by sending the token in a letter to the user's postal address. Though this alternative takes several days to complete, the online service can obtain a verification of the user's name and residency.

Again, one relies on a third party to verify the identity of a new user. However, e.g., mobile phone providers are required by law to verify the identity of their customers in most countries, leading to a higher trustworthiness of those third parties compared to the previous approach (II-A1).

3) Verification through physical presence: The most sophisticated variant of identity verification is verification through physical presence, i.e., the user whose identity has to be checked is in direct proximity of authorized personnel of the online service provider or a trusted third party which acts on behalf of the provider.

Depending on whether the verifying person already knows the to-be-verified user or not, the new user might have to provide official identity documents like passports or ID cards to prove its identity.

Physical verification by the online service provider itself can be regarded as the most secure option. However, it is often unfeasible to establish a dedicated verification entity at the provider and to manually check the identity of maybe thousands of users. Therefore, services like *Postident* [5] by German logistics company *Deutsche Post* exist which provide personal identity verification for third parties. In this case, a new user could verify its online account in one of the many stores of the logistics company.

Summing up the alternatives, verification through another online identity provider can be regarded as the most convenient but also most insecure variant. Verification through a second channel like the mobile phone network or old-school snail mail is more reliable due to law-enforced requirements or the sheer characteristics of the channel (e.g., name and postal address is correct when the letter arrives). However, it is also less convenient and more costly for the participants. Finally, verification through physical proximity provides the most secure procedure at the cost of increased effort for both the online service provider and the end user.

B. Authentication

Within the scope of online services, authentication can be defined as the act of confirming the origin of a request, i.e., from which user or account the request was sent. One can distinguish between three categories (*factors*) of authentication, namely authentication by *something you know* (*knowledge*), by *something you are* (*inherence*), and by *something you have* (*ownership*).

1) Something you know: This authentication factor involves some kind of secret only the respective user knows. Typical examples are passwords or pass phrases, personal identification numbers (PIN), or challenge response procedures (i.e., asking a question only the user can answer). This way of authentication usually can be implemented without much overhead at the provider, but is prone to security breaches resulting from users employing secret credentials too easy to guess or infer from other knowledge. Furthermore, this method can be attacked through phishing [6].

2) Something you are: This means of authentication is based on the behavioral and/or biological characteristics of an individual. Typical methods are to recognize fingerprint, face, voice or retinal pattern. Using inherent characteristics of a human being is convenient for the user because she does not have to remember a secret, but often is complex to implement, error prone and furthermore, the user might be unwilling to share such personal details with a provider.

3) Something you have: In this case, authentication is based on the possession of a key, smart card, security token and the like. In the scope of online services, using this method has the advantage that longer and much more complex security tokens can be used, compared to an ordinary password a user has to know by heart. Implementation usually is straight-forward at the provider, and this method furthermore is very intuitive for the users since it resembles the real-world usage of ordinary keys. However, users might be unwilling to carry additional hardware such as smart cards with them.

Comparing the three methods, authentication based on ownership is the best compromise between security on the one hand, and intuitivity for the users on the other hand. However, using a dedicated hardware component might not be feasible. The latter can be prevented when using a user's smartphone to store the token [7].

C. Problem statement

Today, most online services rely on a verification procedure based on third party identity providers, typically only requiring a valid email address, and employ username-password-credentials for authentication (i.e., something you know). As we have explained, verification through physical presence and authentication via something you have would be a very promising combination regarding security and intuitivity and would therefore be a superior solution to those mechanisms currently most widely used. However, existing ideas result in increased inconvenience for the end-user and more complexity at the provider.

In this work we present a solution that uses that exact combination of identity verification by physical presence and authentication by something you have, which at the same time keeps the typical overhead at a feasible and usable level.

III. Related work

As seen in the previous section, there is a multitude of ways and combinations online services can perform identity verification and authentication. In this section, we focus on systems that resemble our approach with regard to the employed concepts.

Public Key Infrastructures (PKI) are the most widely used method conceptually comparable to our approach. Digital certificates are issued and verified by a Certificate Authority (CA), which can then be used to authenticate oneself. Dependent on the CA and the type of certificate, obtaining this credential requires the verification of one's real-world identity [8]. PKIs are used in conjunction with Secure Socket Layer (SSL) to ensure secure communication, which in general results in increased complexity leading to vulnerabilities, e.g., with regard to validation of SSL certificates within non-browser environments [9]. However, the main disadvantage is that PKIs in its current form are mostly aimed at organisations and corporations, and distribution of certificates to individual users often is not possible to employ with only a reasonable overhead. Since PKIs allow for hierarchical relationships between the CAs among themselves (i.e., one CA may vouch for another), the resulting structure can be regarded as a tree, which is similar to our approach.

An alternative to the rather centralized trust model of a PKI, which relies exclusively on CAs, is the Web of Trust concept. The latter is a decentralized approach to certificate signing, requiring the users to ensure their respective identities among themselves, often based on personal encounters [10]. PGP and GnuPG are well known implementations of this concept, which allow people to exchange messages securely with mutual authentication [11].

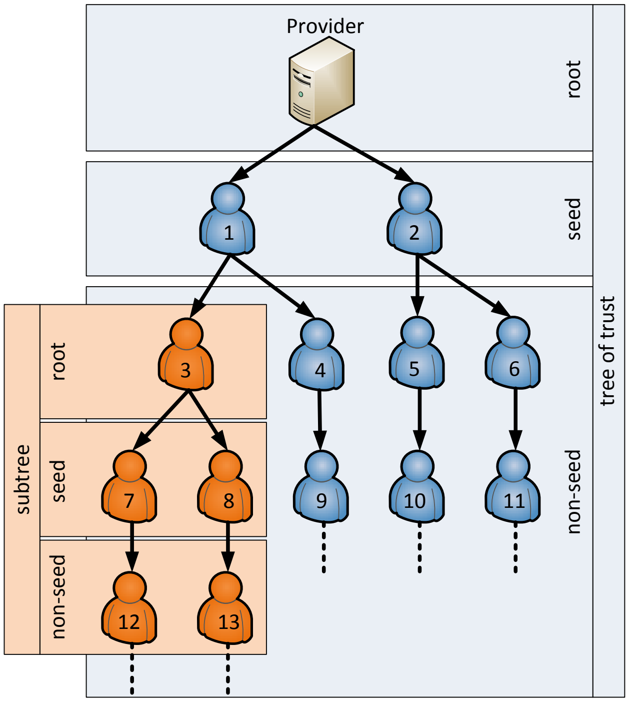

Fig. 1. Participants forming a tree of trust, consisting of three levels root, seed and non-seed. Each subtree also is a tree of trust in itself.

A core concept of the Vis-a-Vis system is the so-called *tree of trust* (see Section IV-D). There are similarly named concepts in other areas which should not be confused with our approach. Presti [12] defines a "tree structure of trust" within the scope of Trusted Computing. In this case, the tree's nodes represent the components of the whole Trusted Computing platform, i.e., from the hardware modules up to the applications. Verbauwhede and Schaumont [13] take a similar approach by partitioning different abstraction levels of electronic embedded systems (e.g., the software level or the circuit level) into secure and non-secure parts. They call the resulting structure a "tree of trust", too. Although both approaches regard trees as a suitable structure for representing trust relationships, they are aimed at different scopes than our system.

IV. Vis-a-Vis

In the following, we describe the Vis-a-Vis concept for identity verification and authentication.

A. Authentication

In order to authenticate the users in the Vis-a-Vis system, a notion of the "something you have" principle is used. The idea is based on the omnipresence of mobile devices such as smartphones or tablets, and the assumption that such devices (or specific accounts on them in case of multi user systems) belong to one and only one user. The device is like a key in the physical world. Authenticating the device therefore suffices to authenticate the respective user.

Technically, authentication is performed by issuing a secret, unique token to each device in the system, which then is included in all requests of the device to the backend (i.e., the provider). To prevent leaking the token, communication between mobile devices and the backend has to be encrypted (e.g., by using SSL). To authenticate the backend itself, traditional means such as SSL certificates can be used.

B. Participants

Vis-a-Vis is a *hybrid system* with some core components being central elements and most of the other participants self-organizing in a decentralized manner. As such, it is not intended as a single web-wide system but to be deployed individually at organizations. A schematic overview is depicted in Figure 1.

The *Vis-a-Vis provider* is the central entity representing the respective organization. It is fully trusted by default since it manages the whole system. At the moment, there is no interaction beyond provider boundaries and thus, there is no need for further, mutual verification of different Vis-a-Vis providers among themselves.

Providers are responsible to activate *seed users*. These users are verified directly by the provider, by any means regarded secure enough for the given scenario, e.g., by authorized personell such as system administrators verifying a user's identity in person (on-location) or by sending activation information via snail mail. Seed users are fully trusted by the provider.

In order to distribute the verification overhead among the participating entities, seed users can further activate *non-seed users*. The identity of non-seed users is verified by seed users through physical proximity, i.e., seed users may decide to hand over the activation token (from mobile device to mobile device) based on existing knowledge (seed user already knows the new user) or based on official documents (seed user checks e.g., ID card or passport).

Non-seed users are also allowed to activate new users - in the same manner as seed users - resulting in further non-seed users. As a consequence, non-seed users differ in their distance from the root node (*distance from root*, see Section IV-E), a measure which can be used to quantify the trustworthiness of a user.

C. Protocol

Adding new users to the system is performed in several steps (see Figure 2). First, an online identity (i.e., an account) has to be created for the new user at the provider (step 1). This step can be triggered by the user itself, by the provider (which is reasonable when the future users are known upfront, such as within a company) or by an existing user. It is important to note that in this step, only the account is created (i.e., prepared). It is neither yet activated nor linked to the user's device, i.e., it is not usable, yet.

In order to activate the account, the user needs a one-time key which is generated by the provider. This one-time key can only be given to the new user by the provider itself or by an existing user - the latter case being the more interesting (step 2). Thus, an existing user wanting to activate a new user requests the new user's one-time key from the provider (step 3 and 4) and then forwards it to the new user (step 5). The forwarding has to be done in a way requiring physical proximity (i.e., "vis-a-vis"), e.g., transfer via Near Field Communication (NFC) or optical codes like QR codes.

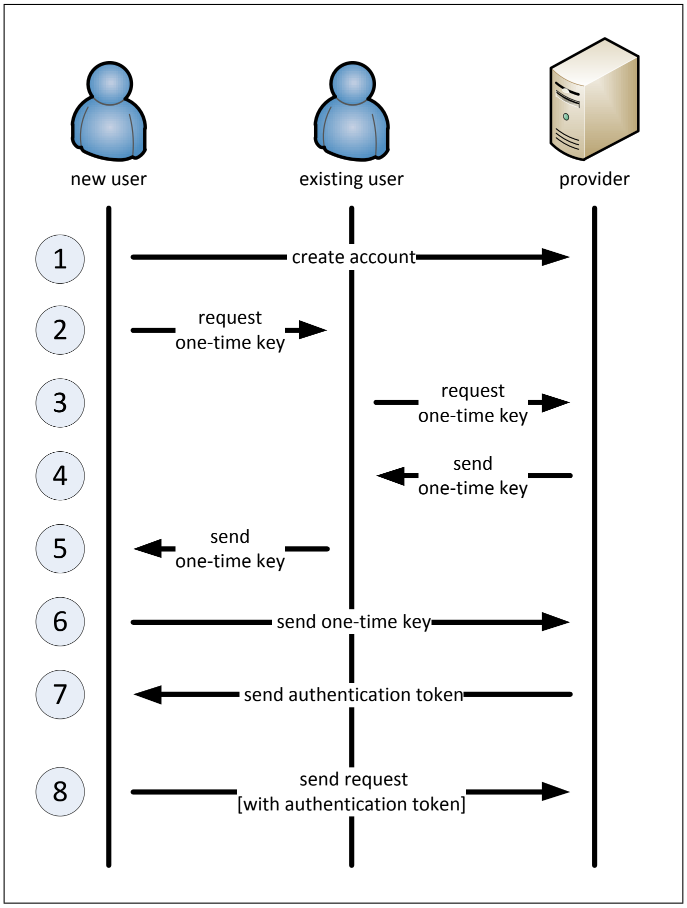

Fig. 2. The Vis-a-Vis protocol.

After receiving the one-time key, the new user sends the key directly to the provider (step 6). The provider now checks whether it is the correct key for the respective user and, when confirmed, sends an authentication token back to the new user (step 7). The user includes this token in all subsequent requests to the backend to confirm its authenticity (step 8).

D. Tree of trust

Performing the above protocol using the described participants results in a tree-like structure. Since this structure describes the evolved trust relations between the users, we can formally define a *tree of trust*

$$T = (V, E) \tag{1}$$

with nodes V and edges E as a rooted tree with *root node* $r \in V$ (the Vis-a-Vis provider), an arbitrary number of *seed nodes* (seed users)

$$S = \{s : s \in V \wedge (r, s) \in E\} \tag{2}$$

and an arbitrary number of *non-seed nodes* (non-seed users) $\bar{S} = V \setminus S$. Each *rooted subtree*

$$T' = (V', E') \tag{3}$$

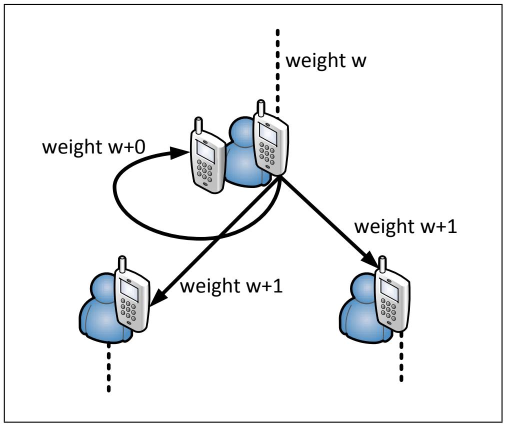

Fig. 3. Part of a weighted tree of trust, showing activation of new users (with decreasing trustworthiness) as well as self-activation with edge weight 0 (i.e., no loss of trustworthiness).

with $E' \subseteq E$ and $V' = \{v' : v' \in V \wedge (\exists v'' \in V' : (v'', v') \in E' \vee (v', v'') \in E')\}$ is also a tree of trust, i.e., each node can be regarded as the root of its own tree of trust containing users which have been activated by itself or its descendants.

Trees of trust are an analogy to the idea of the web-of-trust. The difference is that trees of trust represent a hierarchy of users allowing for a more intuitive assignment of capabilities with regard to some metric (see Section IV-E) whereas in a meshed graph the structure of trust relationships is harder to grasp.

E. Distance from root D_r

There is a single path $P(x, y)$ between each two nodes x and y in the tree, defined as

$$P(x, y) = (v_1, \ldots, v_n) \tag{4}$$

with $v_i \in V, v_1 = x, v_n = y, (v_i, v_{i+1}) \in E$. Based on that we define a measure *distance from root D_r* as

$$D_r(v) = |P(r, v)| \tag{5}$$

A user's distance from its tree's root is a measure for the user's trustworthiness. This measure can be considered when assigning rights or capabilities, e.g., one might limit the length of an *activation chain*, i.e., the path from the tree's root to the user, to a constant C, i.e., $D_r(v) < C : \forall v \in V$.

F. Weighted Tree of Trust

Often it might be desirable to establish a more flexible scheme to assign a trust value to the nodes, considering not only the length of the path from them to the root node but also impact factors like the trustworthiness of the used activation channel.

Furthermore, in some scenarios it is useful to not regard the users as the tree's nodes but their individual devices. Users often possess several mobile devices and it is advisable to issue individual authentication tokens to each device: In case a token is compromised, one can revoke the token without affecting the user's other devices.

Activating a new device by oneself would reduce the trust value of the new device when using the distance from root measure. This can be the desired behaviour, but more often the same person should have the same capabilities on each of its devices.

This problem is solved by introducing weights on the tree's edges (see Figure 3), i.e., each edge (x, y) is assigned a weight w_{xy} correlating to the trustworthiness of the edge itself. Thus, one can define a new trust measure $Trust$ as

$$Trust(v) = \sum \{w_{v_i v_{i+1}} : (v_i, v_{i+1}) \in P(r, v)\} \tag{6}$$

When setting the weight of all edges to 1, $Trust(v) = D_r(v)$.

Using the $Trust$ measure one can allow activation of one's own devices without loss of (calculated) trustworthiness by setting the edge weight to 0. On the other hand, one can also assign edge weights > 1 to mark "more insecure" activations.

V. IMPLEMENTATION

We have implemented and deployed the proposed concept in a real production environment at an educational institution. In this section, we will briefly describe the technical implementation of the various components.

The technical part of the Vis-a-Vis provider has been realised as a backend service, which is programmatically accessed through a REST interface. It furthermore provides a web interface which is intended for account creation. We employ a weighted tree of trust (see Section IV-F), i.e., regarding the users' devices as the tree's nodes and allowing for self-activation of more than one device. Devices are running a custom application, which stores the authentication token and furthermore is used to access protected content provided by the institution.

When a new user wants to create an account, she does so using a dedicated account creation web interface of the Vis-a-Vis provider. Thereby, the user has to provide some personal credentials like name and date of birth, as well as its affiliation to certain groups or departments of the institution. When submitting the registration request, a QR code containing a unique account ID is shown. The user has to scan that code with her smartphone running our custom application, which results in an association of the user's device to the newly created account. It has to be noted that at that point in time, only the association is created, the account itself is not activated, i.e., the user cannot access any protected content, yet.

After that, the user has two choices. She either proceeds to print out a document, containing her account credentials including the associated account ID, which she then has to sign and to provide to authorized personnel at the institution. The latter now check the provided credentials, verify the identity of the new user and then can activate the associated account. The user now can access the protected content and has become a seed user, as she was verified by the Vis-a-Vis provider itself. The seemingly cumbersome usage of printed documents is introduced because at the given institution, it is legally required that the to-be-created seed users sign a consent form. Thus, the Vis-a-Vis system is integrated into the existing workflow.

The alternative way of activating an account is via an existing user. The system is configured to allow existing users to activate new users which belong to the same group. In our mobile application, existing users can browse through and select users which they can activate. They can request the needed one-time key from the provider, which then is encoded in a QR code. The new user can scan this code, resulting in the described protocol being carried out (see Section IV-C). Consequently, the new user has become a non-seed user.

VI. Applications

The Vis-a-Vis concept is predestined to be used at any organisation with a hierarchical structure such as companies, educational institutions, clubs or small project teams. In the following, we describe two use cases, in which our system perfectly fits the inherent structure of the scenario.

A. Use in Companies

A company usually is organised in a hierarchical way, composed of departments and teams, where permissions often should be assigned in accordance to that structure. This perfectly fits the basic building blocks of the Vis-a-Vis system, where senior employees might activate other employees. The hybrid approach of the Vis-a-Vis system ensures that some kind of central authority is present and thus, that seed users can be trusted. Each principal of the respective hierarchy level acts as the responsible seed user of his subordinates. As an example, the CEO of a company would act as a main seed user and unlock its subordinate head of department. In the following, department heads can activate their subordinate team leaders, and so on.

The resulting tree of trust can be used to assign permissions and capabilities, not only based on the user's role but also on her distance to the last directly verified user (which can be measured by the distance from root metric).

B. Use in schools

Another interesting use case is constituted by educational institutions, e.g., schools. This in fact is the scenario in which we have already deployed the system. Within a school, several roles exist, such as teachers, students and parents. These roles are subject to a predetermined hierarchy with different permissions. Furthermore, it is of highest relevance that user identities are verified, i.e., parents and teachers can be sure that they are corresponding with each other.

In this case, initially only the director of a school might have access to the system. As a director representing the highest authority within the school, he has the ability to unlock teachers as seed users. These in turn have the privilege to unlock students which belong to their assigned classes. Students can then activate their parents and give them the permission to access the school network, too.

A key benefit in this use case are the decreasing administrative costs because of the convenient but secure delegation of activation responsibilities.

In case a written agreement from the parents is required by law, the Vis-a-Vis concept is also employable, with parents being authorized directly by the school management (and therefore becoming seed users). Parents then are able to activate further family members by themselves.

VII. Conclusion and future work

In this paper, we presented a novel approach to combine the concept of identity verification through physical presence with the authentication factor ownership, i.e., authentication by something you have. We defined a structure called tree of trust, on which a distance from root metric can be calculated. The latter is a measure for a node's trustworthiness, i.e., it can be used as a parameter for permission assignment. By extending the concept to weighted trees of trust, one can also allow for self-activation of further devices as well as activation by more insecure means, resulting in a lower trustworthiness value. The system perfectly fits scenarios which inherently exhibit some kind of hierarchy and require a central authority, but in which identity verification tasks should be distributed among the system's users.

In future work, it might be interesting to investigate the integration of proximity proofs, i.e., to check whether the transmission of the one-time key really has taken place vis-a-vis, i.e., in direct physical proximity. This would further increase the system's security and the reliability on the trustworthiness of activated accounts.

References

[1] International Telecommunications Union, "Key ict data for the world," http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx, 2013, last accessed date: 2013-07-18.

[2] "Facebook key facts," http://newsroom.fb.com/Key-Facts, 2013, last accessed date: 2013-09-18.

[3] "Facebook login," https://developers.facebook.com/docs/facebook-login/, 2013, last accessed date: 2013-09-18.

[4] "Openid," http://openid.net/, last accessed date: 2013-09-18.

[5] "Postident," http://www.deutschepost.de/dpag?lang=de_EN &xmlFile=1016309, last accessed date: 2013-09-18.

[6] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Client-side defense against web-based identity theft," in *Proceedings of the 11th Annual Network and Distributed System Security Symposium*, 2004.

[7] T. V. N. Rao and K. Vedavathi, "Authentication using mobile phone as a security token," *International Journal of Computer Science & Engineering Technology*, vol. 1, no. 9, pp. 569–574, October 2011.

[8] Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design (4th Edition) (International Computer Science)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[9] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov, "The most dangerous code in the world: validating ssl certificates in non-browser software," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 38–49.

[10] J. Golbeck, B. Parsia, and J. Hendler, "Trust networks on the semantic web," in *Cooperative Information Agents VII*, ser. Lecture Notes in Computer Science, M. Klusch, A. Omicini, S. Ossowski, and H. Laamanen, Eds. Springer Berlin Heidelberg, 2003, vol. 2782, pp. 238–249.

[11] P. R. Zimmermann, *The official PGP user's guide*. Cambridge, MA, USA: MIT Press, 1995.

[12] S. L. Presti, "A tree of trust rooted in extended trusted computing," in *Proceedings of the Second Conference on Advances in Computer Security and Forensics Programme*, 2007, pp. 13–20.

[13] I. Verbauwhede and P. Schaumont, "Design methods for security and trust," in *Design, Automation & Test in Europe Conference & Exhibition*, 2007, pp. 1–6.