Post-print version of the article https://doi.org/10.1016/j.ins.2015.08.046

Analysis-Preserving Protection of User Privacy against Information Leakage of Social-Network Likes[☆]

Francesco Buccafurri[§], Lidia Fotia, Gianluca Lax, Vishal Saraswat[◇]
DIIES, Università Mediterranea di Reggio Calabria,
Via Graziella, Località Feo di Vito, 89122 Reggio Calabria, Italy
◇ CRRao AIMSCS, Hyderabad, India
{bucca, lidia.fotia, lax}@unirc.it, vishal.saraswat@gmail.com
[§]Corresponding Author

Abstract

Recent scientific results have shown that social network *Likes*, such as the "Like Button" records of Facebook, can be used to automatically and accurately predict even highly sensitive personal attributes. Although this could be the goal of a number of non-malicious activities, to improve products, services, and targeting, it represents a dangerous invasion of privacy with possible intolerable consequences. However, completely defusing the information power of Likes appears improper. In this paper, we propose a protocol able to keep Likes unlinkable to the identity of their authors, in such a way that the user may choose every time she expresses a Like, those non-identifying (even sensitive) attributes she wants to reveal. This way, analysis anonymously relating Likes to various characteristics of people is preserved, with no risk for users' privacy. The protocol is shown to be secure and also ready to the possible future evolution of social networks towards P2P fully distributed models.

Keywords: Social Networks, Privacy, Privacy-preserving data analysis, Partially Blind Signature.

1. Introduction

Social network *Likes*, among which the most famous is the Facebook one, are a mechanism massively used by social network users to express their positive/negative association with online contents, such as photos, posts, users' status, groups, music, etc. As a matter of fact, through the above resource evaluation process, users reveal a lot of precious information, mostly

[☆]A shorter abridged version of this paper appears in the Proceedings of the Eleventh Annual Conference on Privacy, Security and Trust (PST 2013) [14].

unknowingly. Indeed, it is often unknown to users the possibility of predicting even hidden aspects of their own personality from digital records of human behavior. A recent study described in [44] involved 58,000 volunteers to demonstrate that Facebook Likes can be used to automatically and accurately predict highly sensitive personal attributes, including sexual orientation, ethnicity, religious and political views, personality traits, intelligence, happiness, use of addictive substances, parental separation, age, and gender. Thus, social network Likes present serious risks related to users' privacy, whose protection is more and more a salient issue, after the first Social Web era, where users seemed little careful about privacy problems. One of the reasons of this, besides personal ones, is that social networks themselves are typically designed in such a way that a user is stimulated to release private information, due to the value that such information has in term of business. As stated in [44], a distinction has to be done between the data that is actually recorded and the knowledge that can be predicted from such data, by using statistical or data mining techniques. But, whenever people have chosen not to reveal certain pieces of information about their lives, predicting them (for example, to propose products or services) represents a dangerous invasion of privacy. For example, the positive association with a political announcement may be welcome in many cases, but it could also lead to a potentially problematic outcome in some other contexts (e.g., when it reveals the users' political leaning). Besides potential risks, we may also cite real-life events showing the dramatic consequences of a seeming innocuous click on a Like button, as the news appeared in the Washington Post that an employee logged on to Facebook and liked the page that was for a candidate challenging his boss, causing his boss to fire him from his job [40].

As it generally happens in data mining, a lot of profitable knowledge can be discovered by analyzing digital records of human behavior in social networks without breaking users' privacy. For this reason, data should be made available in such a way that only privacy-preserving analysis is possible [52, 81, 10, 13, 14]. However, the assumption that any third party which is interested to analyze data can be considered trustworthy is in fact unrealistic, due to the strategic advantage that the utilization of all data, including identifying and sensitive ones, may give to these parties. In the particular case of social network Likes, the strongest measure that can be adopted is

to make Likes completely unlinkable to any attribute of people who express the association. This is what is proposed in [12], where Likes statements are treated as "light-weight e-voting procedures" in such a way that no information about the voter (that is, the author of the Like click) is related to the vote (that is, the Like). The above proposal obviously does not permit any kind of analysis about the population of users who express preferences, thus not only defusing privacy threats but also strategic analysis.

In this paper, we go one step beyond. Our proposal is still to keep Likes unlinkable to the social network profiles of their authors (and, in general to their identity), but to allow users to associate some certified attribute values with their Likes, by choosing every time they state a Like, those (even sensitive) attributes they want to reveal. From this point of view, our paper regards the topic of privacy in a weaker sense w.r.t. the common meaning given by the specific scientific community to this term. Indeed, we protect privacy of users by means of unlinkability to identifying attributes, not by uncertainty-based anonymization (as k-anonymity [69], l-diversity [54], or t-closeness [47]). Thus, even though from a merely technical perspective our solution is closer to security than privacy (in a strict sense), we refer to privacy too (in a loose sense) as, eventually, personal data of users are protected.

However, anonymous analysis relating a Like to various characteristics of the people who expressed such a preference (e.g., age, job, region, country, hobbies, etc.) is preserved with no risk for users' privacy, because there is no way to relate such information to a particular user. Observe that the above requirements evoke what is provided by selective disclosure and bit commitment approaches [8, 70], but a direct application of such approaches to our case is not resolutive since the secret used by a user to enable the disclosure of the chosen attribute would allow third parties to trace the user, thus breaking anonymity. The problem is thus not trivial.

Our solution relies on a cryptographic protocol whose security is mainly based on the infeasibility of discrete logarithms and the robustness of partially blinded signatures. Moreover, we generalize the Facebook concept of Like by assuming that it is not only a positive association with an online content but also a score assigned by the user. Observe that, besides the specific not trivial requirement of linkability of Likes to only user-chosen attributes, our solution preserves the basic properties of an e-voting system as

done in [12]. Indeed, whenever we implement secretness (that is, anonymity of users who express preferences) we have at the same time to avoid that users may misbehave by duplicating improperly their preference. Moreover, all the remaining basic properties of e-voting systems [19, 64], namely individual verifiability, uncloneability, robustness and scalability should be guaranteed.

Our solution relies on a DHT-based P2P social network (assumed given), because we do not assume trustworthiness of the social network issuer. Consider that a recent yet consolidated scientific literature exists envisioning the new paradigm of social network shifting from client-server to P2P infrastructures, coupled with encryption so that users keep control of their information [85, 17, 18]. Anyway, the adoption of our model of Likes does not require a (probably unrealistic) revolution of the current social networks, as it could be implemented by distributing just the functions related to the Like expression and not the contents, possibly relying on (self-managed) cloud computing solutions.

The paper is organized as follows. In the next section, we discuss more in depth the motivations supporting our research. In Section 3, we contextualize our work in the literature giving also some important supports to our proposal. Next, in Section 4, we briefly recall some notions useful to the reader to understand the technical aspects of the paper. In Section 5, we introduce the notations used throughout the paper. The proposed protocol is described in Section 6. In Section 7, we illustrate a possible implementation of our protocol. In Section 8, the analysis of the security of this protocol is presented, by showing that all the desired features are guaranteed also against possible attacks. In Section 9, a performance analysis of the solution is provided. Finally, in Section 10, we draw our conclusions.

2. Motivations

In this section, we provide more detailed motivations that, as a side-effect, should also offer a possible business model underlying our proposal. First, we recall that our proposal comes from the need of finding a solution of the trade-off between the protection of user privacy against involuntary leakage of private information and the opportunity of exploiting digital records produced by users to make strategic analysis. We observe that both the above needs are easily recognizable as realistic, especially if we try to as-

sume a perspective view. Indeed, what today might appear little appealing from a business point of view, tomorrow could become attractive. In our specific case, the core question is the weight that privacy will have in future business models. Likely, in a world where the digital pervasiveness will be dramatically increased together with awareness about threats (concerning both security and privacy), we may expect that people will perceive as critical the risks related to an uncontrolled exposure of private life over the Internet. As a consequence, the attention of users towards the information leakage related to the use of social networks will increase. In this plausible evolution scenario, a social network offering advanced tools to protect privacy could acquire advantages w.r.t. competitors. As an extensive proof of the fact that revealing sensitive personal information through Likes may lead to dangerous situations, we describe a recent case reported in the Italian newspaper *Il Fatto Quotidiano* on January 30, 2014 [76]. According to this article, a man from Parma (Italy), for a simple click on Facebook, risked being condemned criminally. The prosecutor of the court of Parma asked the trial for this man, accused of complicity in aggravated defamation, threatening penalty from six months to three years in prison. The reason of this criminal procedure is that the man had intervened in a dispute between two women belonging to a political movement, by just expressing his Like to one of the insults directed to one of the other two contenders. This is just a striking example of how much a simple click of the mouse could result dangerous. A number of further extreme cases like this can be easily found, but many less sever situations may occur in which a simple Like can have negative impact on personal life, on business, or on political and religious activities.

Despite the above privacy threats, at the same time, companies, researchers, governments, will realize better and better that social networks are a precious source of knowledge that can return strategic advantages in many fields. Concerning Likes, which are the subject of our study, this awareness is already spread not only at research level (as witnessed by papers like [44, 11]), but also at business level. Consider for example what an Italian social media analytics company publishes in its web site [75]. They say that for a thorough analysis of investments in advertising on Facebook, the most interesting information is the provenance of Likes (that is, the origin of the fans), in order to assess whether a campaign of advertising has

obtained a strong impact in the growth of a community. They categorize Likes in four types, each subdivided into a number of subcategories. For example, they distinguish between Likes made from mobile devices, from users who selected Likes from their section Like or that of a friend of the timeline, etc. As another example, consider systems like LikeAlizer [56], which claims to help customers to measure and analyze the potential and effectiveness of their Facebook pages. If the customer is a company, services like this may provide useful information about the impact of the Facebook presence of the company, for example w.r.t. competitors. In this case, the analysis of the impact versus market segments could be extremely important for the company. In all the above cases, no identifying data is significant for analysis, so that the solution of above mentioned trade-off we propose in this paper appears fully applicable. Consider also that the balancing between privacy goals and will of users to be explicitly associated with Likes is completely left to the user, who may choose also to link all personal data with no restriction.

In sum, we argue that in our study, besides dealing with a number of research problems regarding both (1) how to satisfy in the considered distributed evaluation process all the security and privacy properties (that is, uniqueness, secretness, individual verifiability, uncloneability, and robustness of the evaluation process and analysis-preserving unlinkability with identifying information) and (2) how to implement this in a social network, we provide a framework that appears interesting also from a business perspective.

3. Related Work

In this section, we discuss the papers related to our proposal. These papers are grouped by the topics they address, which are selective disclosure, P2P infrastructure, social media, and e-voting.

Selective disclosure. Selective disclosure is the topic most related to our paper. The possibility of a user to disclose some selected attributes when she clicks on a Like evokes what is provided by selective disclosure approaches. There are several techniques used to support partial attribute disclosure. One design for creating attributes that can be selectively revealed as part of a credential borrows directly from the idea of bit commitment [59, 70], which allows a user to commit to a value without having to immediately

reveal it. Thus, when sensitive attributes in a credential are replaced with commitments to those values, we refer to those commitments as private attributes. Most of the proposal regarding selective disclosure need that one party issues a digital signature in which the signed message includes information about the user (that is, attributes). Efficient implementation of these concepts make use of group signatures [43, 50], e-cash [20, 48], anonymous credentials [21, 22, 23], and traceable signatures [42, 74]. These schemes use as building blocks signed attributes and protocols that selectively reveal these attributes or prove properties about them. Their implementation typically encode attributes as a discrete logarithm or, more generally, as an element (exponent) of a representation of a group element, resulting in protocols where the number of group elements transmitted and the commutations performed are linear in the number of encoded attributes. The focus of Bertino et al. [8] is to deeply analyze the impact of protected attribute credentials on trust negotiations, and to devise new strategies allowing interoperability between users adopting various credential formats. Furthermore, the authors adopt a multi-bit hash commitment technique for attribute encoding, as the length of attributes will likely be longer than one bit. Privacy enhanced credentials are different from selective disclosure credentials in that the credential contents are explicitly separated from the credential itself. The system of Holt et al. [37] uses bit commitments to create selective disclosure credentials with a limited amount of data the holder must reveal. Credential sets accomplish this with the help of bit commitment that allows the user to commit to a value without revealing it. Observe that the above approaches are not resolutive in our case, because the secret used by the user to enable the disclosure of the chosen attribute would allow third parties to trace the user herself, thus breaking anonymity.

P2P infrastructure. A crucial component of our proposal, allowing us to obtain the above goal, is to distribute the social network (actually, only the most voting functionalities) over the users, in a Peer-to-Peer (P2P) fashion. Even though, to the best of our knowledge, the problem of resource evaluation in social networks guaranteeing privacy requirements has not been investigated in the literature, the idea of implementing social networks using a P2P paradigm is not new. Fang Wang et al. [85] propose to use a structure of P2P social networks that captures social associations of distributed peers in resource sharing. Peer social networks appear to be mainly composed of

pure resource providers that guarantee high resource availability and reliability. In particular, they report on properties such as degree distribution, clustering coefficient, average path length, betweenness and degree-degree correlations. Sonja Buchegger et al. [17] envision a paradigm shift from client-server to a P2P infrastructure coupled with encryption so that users keep control of their data and can use the social network also locally, without Internet access. Moreover, this paper identifies the core functionalities necessary to build social networking applications and services, and also gives evidence on how the proposed system can be used to form and maintain P2P social networks. PeerSoN [18] is a P2P approach coupled with encryption. The authors design a two-tiered architecture and protocols that recreate the core features of a social network in a decentralized way. This paper focuses on the P2P infrastructure for social networks, outlining the challenges and possibilities of the decentralized paradigm. Cutillo et al. [28] propose a social network based on a P2P architecture to solve privacy issues. The authors' solution leverages the trust relationships that are part of the social network application itself. Privacy in basic data access and exchange operations within the social network is achieved by a multi-hop routing among nodes that trust each other in the social network. In [55], a P2P social network named Social Tube has been designed to facilitate users to share their personal videos or interesting videos they found with their friends. P2P-iSN [51] is a P2P architecture that allows users from different social networks to communicate without involving the social network they belong to. The most recent proposal taking into account the aspect of privacy of users' data stored in company-owned servers, is [7], which exploits an android application using a P2P network to send all shared data. Our proposal shares with the above ones the use of an underlying P2P network to avoid that the social network provider can access sensitive users' data stored on its servers.

Social Media. As social media use has become a principal activity in people's life, there has been an increasing debate about whether or not privacy can be considered a "not realistic" requirement in the information age. Social media benefits have received great attention from the literature, which has given less importance to the risk for privacy derived from such benefits. This issue is discussed in [53], in which the authors examine the benefits and outcomes of interactional privacy and the design of social media interfaces

that are responsive to both relational and privacy needs. In [82], the authors present a model including network composition, disclosures, privacy-based strategies, and social capital. Results indicate that: (i) audience size and diversity impact disclosure and use of advanced privacy settings, (ii) privacy concerns and privacy settings impact disclosure in varying ways, and (ii) audience and disclosure characteristics predict bridging social capital. In [80], Ur et al. survey existing on cross-cultural privacy issues, giving particular weight to online social networking sites. They propose a framework for evaluating the extent to which social network privacy options are offered and communicated. The framework can enable service providers to identify potential gaps in supporting user privacy. It focuses on cultural issues, discusses legal issues in cross-cultural privacy and, finally, delves into user expectations regarding the data-sharing practices and the communication of privacy information. Wisniewski et al. [86] observe that users gain the most benefits when social network sites give them the privacy they desire. By applying structural equations modeling Facebook users, they found that users whose privacy desires were met reported higher levels of social connectedness than those who achieved less privacy than they desired. Moreover, social connectedness played a pivotal role in building social capital. The conclusion of the study is that social networks should aim to achieve privacy fit with user needs to enhance user experience and ensure sustained use. From this point of view, our paper addresses this issue, giving a user flexible tools to prevent privacy violations derived from the expression of Likes.

E-voting. From the side of the specific activity of resources evaluation, our paper is clearly related to the topic of e-voting, for which a wide literature exists. Chaum [25] introduced the notion of *mix-net* as a tool for achieving anonymity in e-mail and in electronic elections. A mix-net consists in a sequence of servers, called mixes. Each server receives a batch of input messages and produces as output the batch in permuted (mixed) order. Such mix-nets are sometimes called *mix cascades* or *shuffle networks*. An observer should not be able to tell how the inputs correspond to the outputs. This property provides voter privacy in an electronic election. Damgård et al. proposed an electronic vote protocol that utilizes the generalized Pallier's cryptosystem [29]. Some approaches to electronic voting based on homomorphic encryptions have been proposed in [4] and [39]. These systems preserve the receipt-freeness property. This means that voting systems do

not generate a receipt when the voter expresses her vote because it could be used by another party to coerce the voter. Unfortunately, receipt-freeness and incoercibility (providing the adversary does not access the registration phase) have in these papers a high price in terms of verifiability and scalability. Also the usage of anonymous broadcast channel makes the scheme impractical. Zwierko et al. [89] proposed an agent-based scheme for secure e-voting. This scheme is based on an authentication protocol with revocable anonymity and can be implemented in a network of stationary and mobile electronic devices. A scheme able to preserve the privacy of both the voters and also the candidates is presented in [46]. This scheme does not require any trusted third party and is based on distributed ElGamal encryption and mix-match. The voters can compute the result by themselves without disclosing their will and the vote of the losing candidates. In [45], an electronic voting scheme capable of providing receipt-freeness is proposed. Receipt-freeness is achieved by distributing the voting procedure between the voter and a smart card. The need of an extra-device is a serious drawback of this proposal.

From the analysis above, it arises that the requirements of e-voting systems [19, 64, 78] are stricter than those necessary in our scenario. Indeed, we argue that, in our cases, properties like eligibility, fairness, and receipt-freeness, would introduce an intolerable price in terms of usability and invasiveness. In other words, the above properties are not coherent with the security level we need in our case. This, combined with the fact that existing e-voting systems guarantee the above features with a significant price in terms of complexity of the solution, requires us to find a new ad-hoc lightweight solution. This is just the goal of this paper, which implements also a different notion of secreteness, as we link Likes to non-identifying attributes.

We recall that *eligibility* means that only those who are authorized to vote can vote and the system has to provide means to validate a voter and a permitted number of votes, *fairness* ensures that no intermediate result can influence the remaining voters, *receipt-freeness* claims that the voter is not able to prove any coercer how she had voted. Eligibility is not necessary in our context because in a social network everyone should be able to perform her evaluation. Moreover, fairness is not required because the resource evaluation in social networks is inherently incremental. Finally, receipt-freeness is unproportionate for evident reasons. Even though we have shown that

these properties are excessive in our context, we have to consider that some fundamental properties need to be satisfied. They are uniqueness, secretness, verifiability, uncloneability, robustness and scalability (as observed in the Introduction).

It is worth noting that a light-weight e-voting system suitable for our application cannot be obtained by trivially relaxing the existing e-voting systems through the disabling of some components, since the elimination of even one of the three components above results in the loss of some basic requirement. Indeed, the elimination of the mix-nets implies that voter anonymity (that is, secretness) is compromised, since the relationship existing between the final vote and its voter is not obscured. Furthermore, an important function of the mix-nets is to ensure that no item is processed more than once, so that its elimination affects also the uniqueness of the vote. The elimination of proofs determines the failure of robustness and uncloneability. Indeed, it is only by means of proofs that a dishonest voter is not able to clone a vote. Not being aware of the key that generated the proof, the dishonest voter cannot generate a bogus vote with a valid proof, and therefore the vote will not be counted. Also verifiability would be compromised, since, in absence of proofs, any interested party could not check both if the ballot has been modified and if information about vote has been leaked.

The above proposals obviously does not permit any kind of analysis about the population of users who express preferences, since votes are always completely unlinkable to any attribute of the voter. Thus, the goal of this paper cannot be reached by using (as it is) any existing e-voting system.

Finally, we observe that an abridged version of this paper appeared in [14]. It is worth noting that this paper includes significant new material w.r.t. the conference version. Indeed, while in [14] the protocol has been presented just in abstract form, this paper adopts mathematical tools to make concrete the protocol. Consequently, security analysis has been significantly deepened and detailed in this paper, and implementation issues have been addressed.

4. Background

In this section, we briefly recall some notions representing the background necessary to understand the technical aspects of the paper. Such

notions are *digital signature*, *blind* and *partially blind signature*, and *distributed hash table*.

4.1. Digital Signature

A *digital signature* is a cryptographic primitive to guarantee data integrity, entity authentication and signer's non-repudiation [15]. It relies on a public key infrastructure where each user has a 'private' signing key and a corresponding 'public' verification key. A user uses its private-key to sign a document and anyone can use its public-key to verify the signature of the document but no one else can forge a signature of a document.

Formally, a *digital signature* Σ is a quadruple of algorithms $\Sigma = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sig}, \mathsf{Vf})$, where

- the setup (parameter generation) algorithm $\mathsf{Setup}()$ outputs the system parameter $PP \leftarrow \mathsf{Setup}(\lambda)$; using security parameter λ,
- the key generation algorithm $\mathsf{KeyGen}()$ outputs a public/private key pair $(pk, sk) \leftarrow \mathsf{KeyGen}(PP)$ given the system parameters PP as input,
- the signature algorithm $\mathsf{Sig}()$ outputs a signature $\sigma \leftarrow \mathsf{Sig}(sk, m)$ with respect to the secret key sk and a message $m \in \{0,1\}^*$,
- and the deterministic, signature verification algorithm $\mathsf{Vf}(pk, m, \sigma)$ outputs true or false.

For consistency, we require that for all $PP \leftarrow \mathsf{Setup}(\lambda)$, $(pk, sk) \leftarrow \mathsf{KeyGen}(PP)$, and $m \in \mathcal{M} \leftarrow \{0,1\}^*$,

$$\mathsf{Vf}(pk, m, \mathsf{Sig}(sk, m)) = \mathsf{true}\,.$$

The system parameter PP contains the security parameter λ itself, and other public information common to all users of the system, like description of cryptographic groups used in the signature scheme.

4.2. Blind Signature

A *blind signature* [24] scheme is a variation of a digital signature typically deployed in privacy related protocols where the signer and the message author (user) are different parties and allows the signer to issue signatures for the user such that

1. the signer learns nothing about the message being signed and should not be able to link or trace an 'unblinded' signature to the user; and
2. the user cannot compute any additional signature without the help of the signer even after getting from the signer many blind signatures.

Formally, a *blind digital signature* Γ is a quadruple of algorithms (Setup, $\mathsf{KeyGen}, \mathsf{Sig}, \mathsf{Vf}$), where Setup, KeyGen and Vf are defined as in a digital signature and the signature issuing algorithm Sig is an interactive protocol executed in three steps, *blinding*, *signing* and *unblinding*, between the signer and the user and outputs a signature $\sigma \leftarrow \mathsf{Sig}(sk, m)$ with respect to the secret key sk of the signer and a message $m \in \{0,1\}^*$ from the user.

The consistency is defined as in digital signature.

4.3. Partially Blind Signature

Partially blind signatures [2] are a generalization of a blind signature allowing the signer to explicitly add to the blind signature some pre-agreed information in unblinded form (for example, an expiry date in the context of electronic cash).

Formally, a *partially blind signature* Π is a quadruple of algorithms ($\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sig}, \mathsf{Vf}$), defined as in a blind signature and except that the interactive signature issuing protocol Sig has an extra step *initialization* during which the signer and the user agree upon the auxiliary information that may be added to the blind signature in such a way that if this information is changed at all verification will fail.

An example instantiation of a partially blind signature [87] based on the intractability of the discrete log problem is as follows. Let G be a cyclic group with prime order q, and g a generator element in G whose order is q. We assume that any polynomial-time algorithm solves $log_g h$ in $\mathbb{Z}_q$ only with negligible probability when h is selected randomly from G. Let $T : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $F : \{0,1\}^* \rightarrow G$ be public cryptographic hash functions. Let $x \in \mathbb{Z}_q$ be a secret key and $y = g^x$ be the corresponding public key. First, the signer and the user agree on the common information *info*. Then, they execute the signature issuing protocol on the blind message m based on the following phases:

P_1 (*Initialization.*) The signer randomly selects $r, u \in \mathbb{Z}_q^*$ and computes $z = F(info)$, $d = g^r z^u$. The signer sends d to the user as a commitment.

P_2 (*Blinding.*) The user randomly selects $t_1, t_2, t_3 \in \mathbb{Z}_q$ as blind factors, and computes $z = F(info)$, $\alpha = d \cdot g^{t_1} y^{t_2} z^{t_3}$ and $\varepsilon = T(\alpha \| info \| z \| m)$ where α and z are being considered in their binary representation. Finally, the user sends $e = \varepsilon - t_2 - t_3 \mod q$ to the signer.

P_3 (*Signing.*) The signer sends back the pair (v, w) to the user, where $v = e - u \mod q$ and $w = r - v \cdot x \mod q$.

P_4 (*Unblinding.*) The user computes $\sigma = w + t_1 \mod q$, $\rho = v + t_2 \mod q$ and $\delta = e - v + t_3 \mod q$. It outputs (σ, ρ, δ) as the resulting signature on m and *info*.

P_5 (*Verification.*) The verifier computes $z = F(info)$, $\alpha = g^{\sigma} y^{\rho} z^{\delta}$ and accepts the signature as valid if and only if $\rho + \delta = T(\alpha \| info \| z \| m)$.

4.4. Distributed Hash Table

A *distributed hash table* (DHT) enables us map a bit sequence to a node of the social network usually using a hash function $\mathcal{H}$ [77, 67]. DHT is typically used in P2P systems to provide lookup services. In these systems, when a peer P_1 shares a resource R, the DHT allows us to obtain the address of a peer P_2 able to route a request for R to P_1. In particular, a random ID is assigned to each peer and an ID (derived from the hash of the resource name) is assigned to each resource. The peer having the ID closest to the ID of the resource stores the information about the peers providing such a resource. The above indexing is dynamically maintained, according to the continuous joining and leaving of nodes in the system.

4.5. Obscuring Operator

We introduce the *obscuring operator* which we denote by the symbol $\odot$. This binary operator satisfies the following four properties. Given any integer values B, E, E_1, E_2:

1. $B \odot E$ can be efficiently computed.
2. $(B \odot E_1) \odot E_2 = B \odot (E_1 \cdot E_2)$, where $\cdot$ denotes the (integer) product operator.
3. Given $Z = B \odot E$, it is infeasible to guess B without the knowledge of E.
4. Given $Z = B \odot E$, it is infeasible to compute E from the knowledge of just Z and B.

Symbol	Description
$\|\|$	The concatenation operator
R	The resource to be evaluated
ID_R	The resource ID
V	The user who evaluates R
CA	The certification authority
CE	The attribute certificate
TTP	The Trusted Third Party
CU_j	The j-th credential (provider) user
$\mathcal{H}, \mathcal{T}, \mathcal{F}$	Cryptographic hash functions
$\mathcal{PBS}_X\{info, m\}$	The partially blind signature
$\mathcal{PBS}_X^{P_i}\{info, m\}$	The partially blind signature at the phase P_i

Table 1: Notations.

A good candidate for this operator is the modular exponentiation function in $\mathbb{Z}_m$, where m is a prime number and $\mathbb{Z}_m$ is the set of integers modulo m. Indeed, if we define this operator as $B \odot E = B^E \mod m$, where $0 < B < m$, it is easy to verify that it satisfies the above four properties. In particular,

1. $B^E \mod m$ can be computed efficiently using fast modular exponentiation;
2. $(B^{E_1})^{E_2} \mod m = B^{E_1 \cdot E_2} \mod m$;
3. Since there are $\phi(m)$ generators of $\mathbb{Z}_m^*$, for any $Z = B^E \in \mathbb{Z}_m^*$ there are at least $\phi(m)$ possibilities for B (and E) and so guessing B from Z without any knowledge of E is infeasible.
4. *Discrete log assumption.* Given a multiplicative group G, the cyclic subgroup $\langle g \rangle$ generated by $g \in G$, and $a \in \langle g \rangle$, it is infeasible to find an integer x such that $g^x = a$.
   In particular, the computation of the discrete logarithm is infeasible, that is, given $Z = B^E \in \mathbb{Z}_m^*$, it is infeasible to compute E from the knowledge of just Z and B.

For the purpose of this article, the obscuring operator $\odot$ will be the modular exponentiation function for a suitable modulus m, a realistic lower bound for which is the maximum size of the attribute field of the certificate. It is also possible to use different values of m for each attribute in which case such values will have to be saved in the certificate (see Section 6).

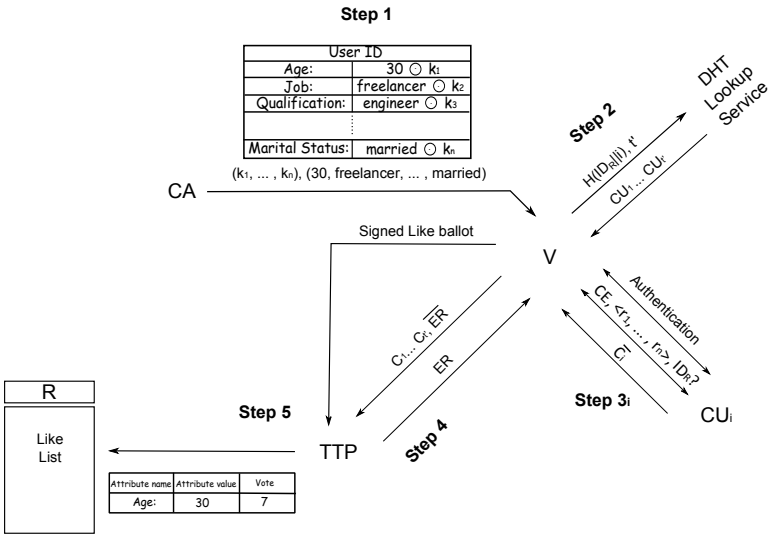

Figure 1: The scenario in which the basic entities operate.

5. Notations

The protocol we propose relies on an underlying distributed social network. We consider given a DHT lookup service allowing us to map a bit sequence to a node of the social network. Throughout the paper, we use the following notations. We consider given an online content R identified by ID_R (e.g., the `URL` of the resource). We denote by V the user who expresses a Like on R. CA is a certification authority granting the attribute certificate CE. We denote by TTP a *functionally* Trusted Third Party [58] and by CU_j the j-th *credential (provider) user* (the role of these entities will be clarified in Section 6). An attribute AT_i is a pair (AN_i, AV_i) where AN_i is the attribute name and AV_i is the attribute value. We also denote the attribute name AN_i by $AT(i).n$ and the attribute value AV_i by $AT(i).v$. Given a cyclic group G, we denote by g a generator element in G. $\mathcal{H}$, $\mathcal{T}$ and $\mathcal{F}$ are public cryptographic hash functions such that $\mathcal{T} : \{0,1\}^* \rightarrow \mathbb{Z}_q$, $\mathcal{F} : \{0,1\}^* \rightarrow G$ and $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^*$. Given a signer X and a message M composed of two parts, say *info* and m, we denote by $\mathcal{PBS}_X\{info, m\}$ the partially blind signature of the message M and by $\mathcal{PBS}_X^{P_i}\{info, m\}$ the output of the partially blind signature at the phase P_i (see Section 4). The notations are summarized in Table 1.

6. Like Expression Protocol

In this section, we describe the protocol allowing a user to express a Like about an online content. Recall that the aim of this protocol is to produce digital records whose analysis does not break users' privacy. Generalizing the current concept of Likes, we assume that a Like is represented by a numeric score, thus allowing a more granular evaluation of online contents. For example, in the case of the Facebook Likes, we have just one possible score value. The entities involved in the Like expression protocol are:

- The user V who expresses the Like.
- The certification authority CA.
- The credential (provider) user CU.
- The Trusted Third Party TTP.

The protocol is composed of five steps, which are *Certificate Issuing*, *CU Identification*, *Credential Issuing*, *Like Click*, and *Like Displaying*.

In the Certificate Issuing step, the certification authority CA issues the attribute certificate CE to the user V. The CU Identification and the subsequent Credential Issuing steps are devoted (i) to deterministically identify those users of the social network who are responsible for generating credentials and (ii) to send them to the user V. Credentials are the "tickets" that the user spends in the Like Click step and are valid only one time per Like. In the last step, TTP has the role of collector of Likes and verifier of their correctness. Moreover, TTP enables the update of the overall score of the on-line content on the basis of the processed Likes.

As said above, credentials for a Like expression are produced collaboratively by several credential users. Some credential users might be corrupted by an adversary, but we assume an honest majority of credential users at all times. This is a common assumption in this context [27, 89, 34], which we call here *CU-collusion assumption*. As a consequence, our technique is parametric with respect to the value t. It is chosen in such a way that the likelihood that t randomly selected users misbehave is negligible.

The communication among the entities mentioned above is based on message exchange and we assume the communication channel to be authen-

ticated and confidential, which can be achieved using asymmetric keys[1].

It could happen that the communication itself identifies a party even though the message is actually anonymous. This concerns a level that belongs to the network communication protocol (not to the social network or the application), and involves external entities, such as the network provider. To address this problem, existing e-voting systems usually adopt mix-nets [25], which are one of the existing anonymity systems [32]. Even though the purpose of our work is not focused on how to implement anonymity, for which there is a large literature [32], as we refer to a P2P-social-network model, we argue that a system like Tor [31] can be directly exploited by integrating it into the social network itself. On the other hand, the real-life applicability of Tor witnessed by the existence of more than one million daily users in 2014 [79], is another point in favor of this choice. In principle, other systems, such as [72, 73], or the development of an ad-hoc system could be considered.

We are ready to present how the evaluation process proceeds. It consists of five steps, which are *Certificate Issuing*, *CU Identification*, *Credential Issuing*, *Like Click*, and *Like Displaying*. Observe that, anonymous communication introduced above is necessary for the communication between users and TTP at the *Like Click* and the *Like Displaying* steps. The protocol, which is sketched in Fig. 1, proceeds as follows:

1. *Certificate Issuing.* In this first step, CA generates the certificate CE for the user V. This certificate contains the user's identifier ID and a list of n attributes of the user. The attributes encode (even sensitive) information about the user, set according to a given predetermined policy. For example, the policy could be stated by the social network provider, which agrees with the certification authority a standard attribute set to include in the certificate. The user's ID appears as plaintext in the certificate whereas the attributes are obscured, in such a way that a third party cannot know their values by accessing the certificate. Without loss of generality, we assume that the values of the attributes are integers.

   The value of each attribute is obscured by means of the obscuring

[1] Authentication is performed through any secure public-key-based authentication protocol. For the sake of presentation, we do not treat this aspect here.

Step	Messages
Certificate Issuing	$CA \rightarrow V : CE, (k_1, \cdots, k_n), (AV_1, \cdots, AV_n)$

Credential Issuing	*For each* $1 \leq j \leq \bar{t}$: {
	$CU_j \rightarrow V : \widetilde{ID_V} = \mathcal{H}(ID_V \| \| S_{CU_j}), AT(i) = \{AN_i, AV_i \odot (k_i \cdot r_i)\}$
	$CU_j \rightarrow V : \overline{c} = g^r F(\langle \widetilde{ID_V}, AT(i) \rangle)^u$
	$V \rightarrow CU_j : \overline{c} = T(\overline{c} \cdot g^{k_1} y_{CU_j}^{k_2} \overline{b}^{k_3} \| \langle \widetilde{ID_V}, AT(i) \rangle \| \overline{b} \| ID_R) - k_2 - k_3 \mod q$
	$CU_j \rightarrow V : C_j = (\overline{v}, \overline{w}) = \mathcal{PBS}_{CU_j}^{P_3} \{\langle \widetilde{ID_V}, AT(i) \rangle, ID_R\}$
	}

Like Click	$V \rightarrow TTP : \overline{C}_j = (\overline{\sigma}, \overline{\rho}, \overline{\delta}) = \mathcal{PBS}_{CU_j}^{P_4} \{\langle \widetilde{ID_V}, AT(i) \rangle, ID_R\}$, *for each* $1 \leq i \leq \bar{t}$ $V \rightarrow TTP : \langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}) \rangle$ $TTP \rightarrow V : \tilde{c} = g^r F(\langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}) \rangle)^u$ $V \rightarrow TTP : \tilde{e} = T(\tilde{c} \cdot g^{k_1} y_{TTP}^{k_2} \tilde{b}^{k_3} \| \langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots,$ $(AT(f(h)).n, B_{f(h)}) \rangle \| \tilde{b} \| (r \| s)) - k_2 - k_3 \mod q$ $TTP \rightarrow V : ER = (\tilde{v}, \tilde{w}) = \mathcal{PBS}_{TTP}^{P_3} \{\langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots,$ $(AT(f(h)).n, B_{f(h)}) \rangle, (r \| s)\}$
Like Displaying	$V \rightarrow TTP : SB = (\tilde{\sigma}, \tilde{\rho}, \tilde{\delta}) = \mathcal{PBS}_{TTP}^{P_4} \{\langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots,$ $(AT(f(h)).n, B_{f(h)}) \rangle, (r \| s)\}$

Table 2: The formal description of the protocol.

operator $\odot$ defined in Section 4.5, which is the modular exponentiation function for a suitable modulus m, a realistic lower bound for which is the maximum size of the attribute field of the certificate.

To obscure the values of the attributes, CA selects a random vector of integers $(k_1, \ldots, k_n)$ and computes $AV_i \odot k_i$. For each attribute $AT_i = (AN_i, AV_i)$, the pair $(AN_i, AV_i \odot k_i)$ is included in the certificate CE. Therefore, in the certificate, instead of the plain value AV_i, only the obscured value $AV_i \odot k_i$ is inserted. At the end of this operation, CA signs the certificate and sends it to V together with the vectors $(k_1, \ldots, k_n)$ through a secure channel.

2. *CU Identification.* V contacts $\bar{t} = 2 \cdot t + 1$ other users who play the role of credential users and they will generate the credentials (we recall that t is a parameter of our technique chosen in such a way that the likelihood that t randomly selected users misbehave is negligible). The i-th credential user CU_j, where $1 \leq j \leq \bar{t}$, is selected by exploiting the DHT lookup service with input $\mathcal{H}(ID_R||j)$, where $||$ is the concatenation operator and, recall, ID_R is the ID of the resource and $\mathcal{H}$ is a cryptographic hash function.
3. *Credential Issuing.* In this phase, V obtains the $\bar{t}$ credentials needed to express a Like, each from a different credential user. A number of operations are repeated for each credential. At the j-th iteration, with $1 \leq j \leq \bar{t}$, the user contacts CU_j. V sends the certificate CE issued at Step 1 to CU_j together with a random integer vector $\langle r_1, \ldots, r_n \rangle$

through a secure channel. CU_j computes $AT(j) = \{AN_i, AV_i \odot (k_i \cdot r_i)\}$ with $1 \leq i \leq n$. Observe that the right-hand element of the pair is the further obscuration of the i-th attribute value by means of the random value r_i, that is, $(AV_i \odot k_i) \odot r_i = AV_i \odot (k_i \cdot r_i)$.
CU_j computes also $\widetilde{ID_V}$ as $\mathcal{H}(ID_V || S_{CU_j})$, thus by applying the cryptographic hash function $\mathcal{H}$ to the concatenation between the user's identifier ID_V (uniquely associated with the profile registration data), and a static secret S_{CU_j} owned by CU_j. The use of S_{CU_j} ensures that none but CU_j is able to link $\widetilde{ID_V}$ to the user. However, this value is the same each time the same user requires a credential from the same credential user.
At this point, CU_j generates the j-th credential $C_j := \mathcal{PBS}^{P_3}_{CU_j}\{\langle \widetilde{ID_V}, AT\rangle, ID_R\}$, where $\langle \widetilde{ID_V}, AT\rangle$ is the pre-agreed information and ID_R is the part signed in blind form. To generate C_j, CU_j randomly selects a secret key $\overline{x} \in \mathbb{Z}_q$ and calculates the correspondent public key $y = g^{\overline{x}}$ (we recall that g is the generator element in the cyclic group G). Then, CU_j and V exchange the pair $\langle \widetilde{ID_V}, AT\rangle$, V computes $\overline{b} = F(\langle \widetilde{ID_V}, AT\rangle)$ and sends it to CU_j.
To verify the correctness of the message, CU_j computes $\overline{b}$ and compares it with the received value. Then, CU_j selects two nonces $r, u \in \mathbb{Z}^*_q$ and sends $\overline{c} = g^r \overline{b}^u$ to V as a commitment. After this, V selects three nonces $k_1, k_2, k_3 \in \mathbb{Z}_q$ as blind factors, and computes $\overline{\alpha} = \overline{c} \cdot g^{k_1} {y_{CU_j}}^{k_2} \overline{b}^{k_3}$, $\overline{\varepsilon} = T(\overline{\alpha} \| \langle \widetilde{ID_V}, AT\rangle \| \overline{b} \| ID_R)$, $\overline{e} = \overline{\varepsilon} - k_2 - k_3 \mod q$. V sends $\overline{e}$ to CU_j.
CU_j sends back the pair $(\overline{v}, \overline{w})$ to V, where $\overline{v} = \overline{e} - u \mod q$ and $\overline{w} = r - \overline{v} \cdot \overline{x} \mod q$. The pair $(\overline{v}, \overline{w})$ is $\mathcal{PBS}^{P_3}_{CU_j}\{\langle \widetilde{ID_V}, AT\rangle, ID_R\}$. This way, the credential user CU_j is aware about the identity of V, but not about the content being evaluated. Finally, $C_j := \mathcal{PBS}^{P_3}_{CU_j}\{\langle \widetilde{ID_V}, AT\rangle, ID_R\}$, is sent to V by the credential user CU_j. Observe that no anonymous communication is needed for all messages exchanged between V to CU_j, because CU_j is aware about the identity of V.

4. *Like Click.* This step starts after the user has collected the $\overline{t}$ credentials. The first task done by V is to unblind the above credentials. V computes $\overline{\sigma} = \overline{w} + k_1 \mod q$, $\overline{\rho} = \overline{v} + k_2 \mod q$ and $\overline{\delta} = \overline{e} - \overline{v} + k_3 \mod q$. It outputs $(\overline{\sigma}, \overline{\rho}, \overline{\delta}) = \mathcal{PBS}^{P_4}_{CU_j}\{\langle \widetilde{ID_V}, AT\rangle, ID_R\} = \overline{C_j}$ as the resulting signature on the message ID_R and the pre-agreed common

information $\langle \widetilde{ID_V}, AT \rangle$. Then, V submits all the credentials to TTP. At this point TTP checks that at least $t+1$ credentials:

(a) are authentic and intact, by verifying their digital signature;
(b) contain ID_R (that is, they refer to same content);
(c) have been issued by the correct credential user, as computed in Step 2;
(d) are *fresh* credentials. A fresh credential $\mathcal{PBS}^{P_4}_{CU_j}\{\langle \widetilde{ID_V}, AT \rangle, ID_R\}$ is a credential that is not identical to the credentials received in the past. To detect a possible re-submission of the credential, TTP uses a database containing all past credentials. The failure of this test means that the user has already evaluated the content R. Thus, in this case, no further operation is performed by TTP.

If all these tests succeed, then the procedure continues as follows. Let $h \leq n$ be the number of attributes that V decides to disclose. V computes $T = \{(B_i, e_i)\}$ with $i \in [1, h]$, where B_i is the value of a chosen attribute, say A_x, with $x \in [1, n]$, and e_i is equal to $k_x \cdot r_x$. At this point, TTP has to verify that the chosen attributes are consistent with AT. To do this, the following function $f : \{1, \cdots, h\} \rightarrow \{0, \cdots, n\}$ is introduced. It is defined as follows: $f(i) = j$ if there exists unique $j \in [1, n]$ such that $AT(j).v = B_i \odot e_i$; $f(i) = 0$ otherwise. Note that the aim of f is to map each disclosed attribute to one obscured attribute in AT.

If there exists $p \in [1, h]$ such that $f(p) = 0$, then the protocol aborts. Indeed, this is the case in which a disclosed attribute corresponds to either none or multiple obscured attributes, which are both proofs of inconsistency. Otherwise, TTP signs by a partially blind signature the *evaluation record* $ER = \mathcal{PBS}^{P_4}_{TTP}\{\langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}) \rangle, (r||s)\}$, where $(AT(f(i)).n, B_{f(i)})$ is the pair $\langle$attribute name, attribute value$\rangle$ disclosed by V, s is the Like score expressed by the user about the online content and r is a 128-bit random sequence generated by V to identify this Like. Both s and r are blindly signed, so that they keep hidden from TTP. The protocol requires that TTP uses the same key pair to sign each evaluation record. The reason of this will be clarified in Section 8. At this point, TTP randomly selects $\tilde{x} \in \mathbb{Z}_q$ that is a secret key, the corresponding public key is $y = g^{\tilde{x}}$. Then, TTP and V exchange the pair $\langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}) \rangle$, which is the

common information. V computes $\tilde{b} = F(ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}))$ and sends it to TTP. To verify the correctness of the common information, TTP computes $\tilde{b}$ and compares it with the received value. Then, TTP selects two nonces $r, u \in \mathbb{Z}_q^*$ and computes $\tilde{c} = g^r \tilde{b}^u$. TTP sends $\tilde{c}$ to V as a commitment. After this, V selects three nonces $k_1, k_2, k_3 \in \mathbb{Z}_q$ as blind factors, and computes $\tilde{\alpha} = \tilde{c} \cdot g^{k_1} {y_{TTP}}^{k_2} \tilde{b}^{k_3}$, $\tilde{\varepsilon} = 2T(\tilde{\alpha} \| \langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}) \rangle \| \tilde{b} \| (r||s))$, $\tilde{e} = \tilde{\varepsilon} - k_2 - k_3 \mod q$. V sends $\tilde{e}$ to TTP. Finally, TTP sends back the *evaluation record* $ER = (\tilde{v}, \tilde{w}) = \mathcal{PBS}_{TTP}^{P_3}\{\langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}) \rangle, (r||s)\}$ to V, where $\tilde{v} = \tilde{e} - u \mod q$ and $\tilde{w} = r - \tilde{v} \cdot \tilde{x} \mod q$.

5. *Like Displaying.* The evaluation record obtained by the user is unblinded, by producing the message $SB = \mathcal{PBS}_{TTP}^{P_4}\{\langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}) \rangle, (r||s)\}$, which we call *Signed ballot*. The user unblinds the evaluation record as follows. V computes $\tilde{\sigma} = \tilde{w} + k_1 \mod q$, $\tilde{\rho} = \tilde{v} + k_2 \mod q$ and $\tilde{\delta} = \tilde{e} - \tilde{v} + k_3 \mod q$. It outputs $(\tilde{\sigma}, \tilde{\rho}, \tilde{\delta})$ as the resulting signature on the message $(r||s)$ and the pre-agreed common information $\langle ID_R, (AT(f(1)).n, B_{f(1)}), \cdots, (AT(f(h)).n, B_{f(h)}) \rangle$. Then, V sends the signed Like ballot to TTP. To prevent timing attacks, an unpredictable delay before sending the ballot to TTP is introduced. Now, TTP verifies the ballot signature and checks that another ballot with the same random r has never been received, detecting a cloned ballot. If both checks succeed, then the new Like is accepted and delivered to the social network provider, which will update both the overall score of the corresponding online content and the list of the attributes associated with each Like (called *Like list*).

The formal description of the message exchanged in the protocol, based on a common syntax [25], is reported in Table 2.

7. Implementation Issues

In this section, we deal with the implementation of our proposal, by describing the adopted cryptographic hash function and partially blind signature, the underlying distributed social network, the data structures and algorithms to use.

We start by describing the choice of the cryptographic hash functions $\mathcal{H}, \mathcal{T}, \mathcal{F}$ introduced in Table 1. Recall that a cryptographic hash function is a function taking an input x and generating an output y (called *digest*, guaranteeing the following basic properties: *collision resistance*, *preimage resistance*, and *second-preimage resistance*. The first property is that it is computationally infeasible to find any pair x and x' of distinct inputs producing the same digest. The second property asserts that given y, it is computationally infeasible to find any preimage x such that its digest is y. Finally, the third property states that it is computationally infeasible to find any second input which has the same output as any specified input. In the last years, many proposals of cryptographic hash functions appeared in the literature. MD5 [66] takes as input a message of arbitrary length and produces as output a 128-bit digest. B. den Boer et al. [30] found that the round function of MD5 is not collision resistant. Another function, called RIPEMD-160, was developed in the framework of the EEC-RACE project Race Integrity Primitives Evaluation [1]. RIPEMD-160 is standardized by ISO/IEC in 1997 (Part 3 of ISO/IEC 10118). Some weaknesses of RIPEMD-160 have been recently detected in [57].

The Secure Hash Algorithm (SHA) is a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) [60]. SHA-256 generates an almost unique, fixed size 256-bit (32-byte) digest and is one of the strongest hash functions available. This makes it suitable for password validation, challenge hash authentication, anti-tamper, digital signatures. Among the various cryptographic hash functions, we choose SHA-256 as implementation of $\mathcal{H}, \mathcal{T}, \mathcal{F}$ because it satisfies the characteristics required by law in many applications.

Now, we describe how we implement partially blind signatures introduced in Section 4. A number of partially blind signature schemes using different assumptions exist. The most relevant are: Abe and Fujisaki's scheme [3], based on RSA, [2, 87], relying on the discrete logarithm problem, and Fan and Lei's scheme [33], based on the quadratic residues problem. Among these schemes, our implementation follows the approach OR-Schnorr, presented in [87], which provides a rigorous proof of security based on the discrete logarithm problem. According to the notations introduced in Section 4, in our implementation, we have a 256-bit prime q and a 1024-bit prime such that $p = k \cdot q + 1$ and an order q cyclic group $G_q \subset \mathbb{Z}_p^*$ which

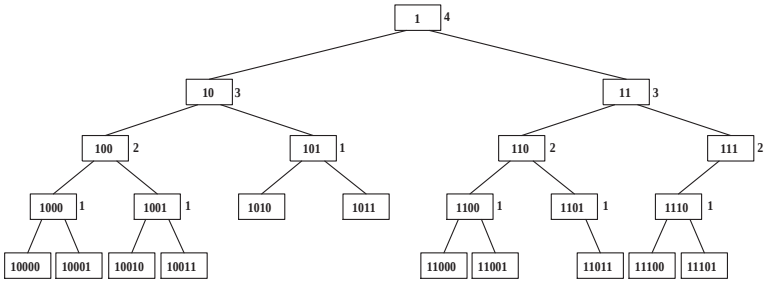

Figure 2: An example of a 5-level LBT.

resists methods for solving the discrete logarithm problem. We obtain a generator element g of G_q, by randomly choosing h in the range $1 < h < p$ in such a way that $h^k \not\equiv 1 \pmod{p}$. As a consequence, a generator of G_q is equal to $g = h^k \mod p$.

As a further contribution, we focus on the underlying distributed social network which our protocol relies on. Among the possible ways for implementing the DHT lookup service necessary to map a bit sequence to a node of the social network (see Section 5), we choose to implement this service by following the approach presented in [16], named *Tree-Based DHT Lookup Service* (TLS). TLS implements a logical network with tree topology allowing sharing information embedded into nodes. The basic data structure of TLS is a hash table distributed on a binary tree, named LBT, in which every credential user account is associated with a node of the binary tree. A given node N belonging to the level $x-1$ of the tree is identified by an x-bit sequence, as reported in Figure 2 (for more details about TLS, the reader is referred to [16]).

We recall that at Step 2 of the Section 6, given a resource identified by ID_R, we computed $h = \mathcal{H}(ID_R||i)$ which is the input of the DHT lookup service. In our implementation, the credential user associated with h is the leaf node whose ID is a prefix of h. For example, if $h =< 10000 \cdots >$, then the credential user which generates the credential is that associated with the left-most leaf node (that is, that with ID=10000).

Now, we discuss about the data exchange among actors (users, certification authority, credential users and TTP). This is obtained by the use of XML documents.

Concerning the certificates generated by the certification authority and issued to users, we extend the standard X.509. The resulting certificate is

```
<complexType name="certificateType"> <sequence> <element
name="serialNumber" type="integer"/> <element name="issuer"
type="tns:issuerType"/> <element name="start" type="date"/> <element
name="end" type="date"/> <element name="publicKey"
type="hexBinary"/> <element name="IDv" type="integer"/> <element
name="propertyList" type="tns:propertyListType"/> </sequence>
</complexType>

<complexType name="issuerType"> <sequence> <element
name="commonName" type="string"/> <element name="organizationalUnit"
type="string"/> <element name="organization" type="string"/>
<element name="country" type="string"/> </sequence> </complexType>

<complexType name="propertyListType"> <sequence> <element
name="property" type="tns:propertyType" minOccurs="0"
maxOccurs="unbounded"/> </sequence> </complexType>

<complexType name="propertyType"> <attribute name="name"
type="string" use="required"/> <attribute name="value"
type="hexBinary" use="required"/> <attribute name="key"
type="hexBinary"/> </complexType>
```

Listing 1: The complex types `certificateType`, `propertyListType` and `propertyType`.

the XML complex type `certificateType` (Listing 1). In addition to the traditional fields (e.g., serialNumber, issuer, start, end, publicKey), each certificate has a list of the properties associate with the user (`propertyList`). Each property is a pair ⟨name,value⟩ (e.g., the name of a property could be 'gender'). Recall that, for privacy reasons, the value of all properties in the certificate are obscured. As a consequence, the XML type of such values is `hexBinary`. Finally, the XML attribute `key` maps the key necessary to decrypt obscured attribute.

Now, we introduce the XML complex type called `credentialType`, which represents the credential generated by the credential user and issued to the user. Its structure is reported in Listing 2. To prevent linkability, a credential does not have an identifier: it contains an integer value identifying the credential user ($\texttt{ID}_{\texttt{CU}}$), an integer value identifying the resource ($\texttt{ID}_{\texttt{r}}$), the hash value of the concatenation between the user's identifier and a static secret ($\texttt{H\_ID}_{\texttt{V}}$), and a list of the properties associated with the user (`AT`). Recall that these properties have the same name as the properties of the certificate, but their values have been obscured (see Step 3 of Section 6).

As for the ballot generated by TTP, its structure, called `ballotType`, is reported in Listing 3. The ballot stores a reference to the resource's identifier ($\texttt{ID}_{\texttt{r}}$) and the blindly signed value (thus, the XML attribute `partiallyBlindType` is equal to true) of a 128-bit random sequence (`r`) and of the score

```
<complexType name="credentialType"> <sequence> <element name="IDcu"
type="integer"/> <element name="IDr" type="partiallyBlindType"/>
<element name="H_IDv" type="hexBinary"/> <element name="AT"
type="tns:propertyListType"/> </sequence> </complexType>

<complexType name="partiallyBlindType"> <simpleContent> <extension
base="hexBinary"> <attribute name="partiallyBlind" type="boolean"/>
</extension> </simpleContent> </complexType>
```

Listing 2: The complex type `credentialType` and `partiallyBlindType` in the XML Schema `like`.

```
<complexType name="ballotType"> <sequence> <element name="IDr"
type="integer"/> <element name="r" type="tns:partiallyBlindType"/>
<element name="s" type="tns:partiallyBlindType"/> <element
name="disclosedPropertyList" type="tns:propertyListType"/>
</sequence> </complexType>
```

Listing 3: The complex types `ballotType` in the XML Schema `like`.

specified by the user (`s`). Moreover, the user includes in the ballot the properties to disclose (`disclosedPropertyList`): for each property, the XML attribute `value` is provided in plaintext (that is, not obscured) and the XML attribute `key` contains the secret necessary to decrypt obscured attributes. The data structures described above are contained in the XML root element `like` (Listing 4).

After describing the data structures used in the implementation of our protocol, we discuss the algorithm performing the Like click step (see Section 6), which is the only one requiring a more detailed explanation. Its pseudocode is shown in Algorithm 1. The algorithm receives as input the XML document, say `like.xml`, presented to TTP by the user. This document contains the certificate issued by the Certification Authority, the credentials collected by the user, and the ballot that TTP will sign if the check on the presented credentials succeeds. The output is an XML document, say `ballot.xml`, which is the signed ballot. Observe that, in our pseudocode, we use XQuery expressions [83] to extract and manipulate data from the XML documents and SQL expressions to insert and query data from the database, called `DB`, which supports our implementation.

The algorithm proceeds as follows. Each credential presented by the user is analyzed. The XQuery expression at line 2 returns all the XML elements `credential`, children of the root `like` in the document `like.xml`. First, the authenticity and integrity of the signature on the credential is verified by

```
<?xml version="1.0" encoding="UTF-8"?> <schema
xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.example.org/like"
xmlns:tns="http://www.example.org/like"> <element name="like">
<complexType> <sequence> <element name="certificate"
type="tns:certificateType" minOccurs="0"/> <element
name="credential" type="tns:credentialType" minOccurs="0"
maxOccurs="unbounded"/> <element name="ballot" type="tns:ballotType"
minOccurs="0"/> </sequence> </complexType> </element> ... </schema>
```

Listing 4: The XML Schema `like`.

using the method `checkSignature` (line 3). A fake credential is skipped and no further analysis is carried out on it – this is implemented by the `continue` statement (line 4). Then, it is checked that the credential has been issued from the credential user for the resource declared by the user in the ballot (line 6). To detect a possible re-submission of the credential, it is checked by an SQL query (line 9) whether the credential is included into the table `UsedCred` of the database `DB` storing references to all credentials received in the past from TTP. This table consists of two columns, the former identifies the credential user that issued the credential and the latter represents the resource's identifier. If the table already includes this credential, the next credential is processed. Otherwise, the reference to the current credential is inserted into the table of already presented credentials. The variable *cont* is incremented by 1 (line 14) each time a credential is valid (that is, succeeds all the tests above).

Now, it is verified (lines 15-29) the consistency of the attributes that the user decides to disclose (`/like/ballot/disclosedPropertyList/property`) with the attributes contained in the credential (`/like/credential/AT/property`). An inconsistency occurs if a disclosed property (i) appears in the credential with a different value (`wrong = true`) or (ii) does not appear in the credential (`found = false`). The variables `wrong` and `found` are used to detect such cases and, as optimization, to break the `for` loop. The method `decrypt` receives as input an obscured value and a key and returns the plaintext value of the attribute if the key is correct, null otherwise. If an inconsistency is found, then the property incorrectly disclosed from the user is removed from the ballot (line 27).

Finally, if at least $\bar{t}/2 + 1$ credentials are valid (line 32), TTP proceeds by blindly sign the ballot according to Step 4 at the end of Section 6.

The output of the algorithm is an XML document `ballot.xml` contain-

Algorithm 1 *Like*

Input *like.xml*: an XML document
Output *ballot.xml*: an XML document
Variable *cont*: an integer
Variable *wrong*: a boolean
Variable *found*: a boolean

```
 1: cont := 0
 2: for $c in document("like.xml")/like/credential do
 3:     if checkSignature($c) = false then
 4:         continue
 5:     end if
 6:     if $c/ID_r != document("like.xml")/like/ballot/ID_r then
 7:         continue
 8:     end if
 9:     if (SELECT * FROM BD.BurnedCred
           WHERE ID_cu = $c/ID_cu AND ID_r = $c/ID_r) != null then
10:         continue
11:     else
12:         INSERT INTO DB.BurnedCred VALUES ($c/ID_cu, $c/ID_r)
13:     end if
14:     cont++
15:     wrong := false
16:     for $dp in document("like.xml")/like/ballot/disclosedPropertyList/property do
17:         found := false
18:         for $p in document("like.xml")/like/credential/AT/property do
19:             if $dp/@name = $p/@name then
20:                 found := true
21:                 if found AND $dp/@value != decrypt($p/@value, $dp/@key) then
22:                     wrong := true
23:                     break
24:                 end if
25:             end if
26:         end for
27:         if wrong OR !found then
28:             REMOVE $dp
29:         end if
30:     end for
31: end for
32: if cont > t̄/2 then
33:     <like>
34:       blindSignature(document("like.xml")/like/ballot)
35:     </like>
36: end if
```

ing the ballot and its partially blind signature.

8. Security Analysis

In this section, we analyze the security of the proposed model. We first present our security model by identifying all the expected security and privacy properties and an exhaustive attack model. Then, we formally prove that our protocol is secure according to this security model.

8.1. Security Model

In this section, we state the security properties required of the proposed "Like" protocol. We denote by S the so obtained security model. We remark that we consider plausible attacks from most enabled adversaries who are "enabled insiders", namely, the credential users, the TTP and their collusion. Hence, a protocol which is secure against such adversaries is also secure against any other adversary who is an outsider or a normal user in the system.

8.1.1. Security Property 1 – Uniqueness

SP1 (Security Property 1) is defined as follows: *An online content can scored only once by a user. That is, once an online content has been scored by a user, the user itself should not be able to score that online content again.*

The attack/failure model we consider to describe how this property can be threaten is the following:

- **Attack AU1**: An adversary tries to resubmit to TTP for the second time a signed Like ballot.
- **Attack AU2**: An adversary attempts to resubmit the same credentials to TTP in order to be authorized to express a Like for the second time.

  Attack AU3: An adversary tries to obtain for the second time fresh credentials for the same content.
- **Failure FU1**: Two different users generate a Like ballot with the same random r (which, we recall, is used as identifier of the Like), thus producing a false positive in validity check performed by TTP.

8.1.2. Security Property 2 – Secretness and Unlinkability

SP2 (Security Property 2) is defined as follows: *The score given by a user to an online content is secret and any information about it cannot be linked to the user who expressed it.*

The attack model we consider to describe how this property can be threaten is the following:

- **Attack AS1**: A credential user may try to find information about the user who expressed the Like.

- **Attack AS2**: A credential user may try to guess the online content evaluated by the user.
- **Attack AS3**: TTP may try to link the user and its score.
- **Failure AS4**: A collusion between TTP and a credential user may try to link the user and its score.

8.1.3. Security Property 3 – Individual Verifiability

SP3 (Security Property 3) is defined as follows: *The result announced by the TTP must be individually verifiable. That is, each individual user can verify that her score is fair and its Like has been counted.*

The failure model we consider to describe how this property can be threaten is the following:

- **Failure FI1**: Two different users generate a Like ballot with the same random r, thus compromising the individual verifiability of the corresponding published scores.

8.1.4. Security Property 4 – Uncloneability

SP4 (Security Property 4) is defined as follows: *The proposed scheme must be uncloneable and bogus ballots must be detectable.*

The attack model we consider to describe how this property can be threaten is the following:

- **Attack AC1**: The adversary tries to forge a valid ballot by starting from a legal one.

8.1.5. Security Property 5 – Robustness

SP5 (Security Property 4) is defined as follows: *Any malicious behavior by (at most t) credential users must be detectable.*

Due to CU-collusion assumption (see Section 6), the attack model we consider to describe how this property can be threaten is the following:

- **Attack AR1**: t or less credential users collude to break security properties of the protocol.

8.2. Security Analysis

In this section, we analyze the security of our protocol, which is also based on the difficulty of solving the discrete logarithm [6]. We consider separately all the security properties we have to guarantee, which are uniqueness, secretness, individual verifiability, uncloneability, and robustness.

The following theorem states that the protocol satisfies the property SP1 of the security model S.

Theorem 1 (Uniqueness). *The protocol is resistant to attacks AU1, AU2, and AU3. The probability of occurrence of failure FU1 is* $P(FU1) \approx 1 - e^{-u^2/2D}$*, where* u *is the number of users and* D *is the cardinality of the domain of the random* r *included in the Like ballot.*

Proof. Resistance to Attack AU1. Recall that AU1 is the Like replay attack where an adversary tries to resubmit to TTP for the second time a signed Like ballot. Such an attack is infeasible since TTP can detect any double submission of a Like because it stores all submitted Like ballots and each one is identified by r.

Resistance to Attack AU2. Recall that AU2 occurs when a credential replay attack is performed. That is, an adversary attempts to resubmit the same credentials to TTP in order to be authorized to express a Like for the second time. Such an attack is infeasible since TTP detects the double credential submission because it stores all submitted credentials (Step 4.(d)). Also in the case the attacker requires a new certificate to CA, the user's ID is the same, thus resulting in the failure of the attack because $\widetilde{ID_V}$ depends on only ID_V.

Resistance to Attack AU3. Recall that AU3 occurs when a user does not cast its vote but an adversary tries to score an online content more than once on behalf of the user more than once. This may happen if the adversary tries to obtain for the second time fresh credentials for the same content. Two cases may hold. The contacted credential user, say CU_j, is the one returned by Step 2 of the protocol, that is, CU_j is identified by $\mathcal{H}(ID_R||j)$. In this case, the credential C_j issued from CU_j to V contains the same value $\widetilde{ID_V} = \mathcal{H}(ID_V||S_{CU_j})$ which depends only on the user's identifier ID_V and on the static secret S_{CU_j} held by CU_j. As a consequence, TTP can detect the double submission. The second case occurs when the attacker contacts a provider CU_x different from that is returned by the lookup service with

input $\mathcal{H}(ID_R||j)$, $j = 1, \ldots, \bar{t}$ at Step 2 of our protocol. TTP rejects such submission during the check done in Step 4.(c) and such an attack fails.

Occurrence of failure FU1. Recall that FU1 occurs whenever two different users generate a Like ballot with the same random r, thus producing a false positive in the validity check performed by TTP. It is easy to realize that the occurrence probability $P(FU1)$ can be approximated by $1 - e^{-u^2/2D}$, as FU1 can be modeled as birthday attack.

Remark. In our case, even hypothesizing an unrealistically high number of users, say $u = 10^{12}$, since r is a 128-bit sequence, $P(FU1) < 10^{-15}$ and hence it is negligible in practice. Moreover, we observe that depending on what we use as user's identifier ID_V, we may ensure the Like uniqueness w.r.t. only the user's profile in the social network or the Like uniqueness w.r.t. the physical person identity. The former is obtained for example if we use as identifier the `URL` of the profile, while the latter is achieved if we adopt a secure PKI certifying the ownership of the public key registered in the profile and by using as ID_V some personal identifier (like, for example, the VAT number). About the possibility that two obscured values $AV_1 \odot (k_1 \cdot r_1)$ and $AV_2 \odot (k_2 \cdot r_2)$ in AT collide, the probability of this event is negligible thanks to the randomness of r_1 and r_2 assuming that the number of bits of such random values is sufficiently large.

The following theorem states that the protocol satisfies the property SP2 of the security model S.

Theorem 2 (Secretness and Unlinkability). *The protocol is resistant to attacks AS1, AS2, AS3, and AS4.*

Proof. Resistance to Attack AS1. Recall that SU1 occurs when a credential user try to find information about the user who expressed the Like. By contradiction, suppose that a credential user CU guesses whether the real value of the obscured attribute $A' = A^k \mod m$ is equal to V. This means that it has found a value k' such that $V^{k'} \mod m = A'$ which corresponds to find the discrete logarithm of A', which is infeasible. Analogously, no other entity of the scenario is able to guess the value of non-disclosed attributes.

Resistance to Attack AS2. Recall that SU2 occurs when a credential user may try to guess the online content evaluated by the user. Consider that the score s is initially sent to TTP by V in the message $\tilde{e} = \tilde{\varepsilon} - k_2 - k_3 \mod q$, where $\tilde{\varepsilon} = T(\tilde{\alpha}\|ID_R\|\tilde{b}\|(r||s))$. Anyway, this message does not give TTP the possibility to know s, because the cryptographic hash function

T is applied to the concatenation of s with other values and T is one-way. For the same reason, also r cannot be used to link the voter. Thus, once the Like ballot has been produced, it cannot be linked to the user. Similarly, the credential users know only the user's public key and have no possibility to guess the score of the online content being evaluated. The only remainder possibility is that a credential user CU_j may try to guess the online content evaluated by the user. Recall that CU_j receives the message $\overline{e} = T(\overline{\alpha}\|a_i\|\overline{b}\|ID_R) - k_2 - k_3 \mod q$ and is aware about the identity of the user. In this attack, the credential user tries to guess ID_R knowing that it has been selected on the basis of the online content. To do this, CU_j should be able to invert the composition of two hashes, namely the DHT lookup function and the function $\mathcal{H}$. However, even though the inversion of the DHT lookup function is feasible because it is not a cryptographic hash function, this does not occur for $\mathcal{H}$, considering that ID_R is the `URL` of the corresponding online content. Thus, this attack is prevented.

Resistance to Attack AS3. Recall that SU3 occurs when TTP tries to link the user and its score. We prove that no link between the certificate and the credentials issued to a user exists. Indeed, the user ID is not included in the credential and any attribute $AV_i \odot k_i$ in the certificate is transformed into $AV_i \odot (k_i \cdot r_i)$. Thanks to the further obscuration performed by r_i, there is no possibility to link the credential to the attribute certificate (and then to the user). The only information known by TTP is the ID of the content and the disclosed attributes. TTP cannot link the user and the preference score of her Like ballot thanks to the use of the partially blind signature (at Step 4). In particular, the score s is initially sent to TTP by V in the message $\tilde{e} = \tilde{\varepsilon} - k_2 - k_3 \mod q$, where $\tilde{\varepsilon} = T(\tilde{\alpha}\|ID_R\|\tilde{b}\|(r||s))$. Anyway, this message does not give TTP the possibility to know s, because the cryptographic hash function T is applied to the concatenation of s with other values and T is one-way. For the same reason, also r cannot be used to link the voter. Thus, once the Like ballot has been produced, it cannot be linked to the user.

Resistance to Attack AS4. Recall that SU4 occurs when a collusion between TTP and a credential user try to link the user and its score. Clearly, the collusion between TTP and a credential user allows them to link the pre-agreed information ID_R to the identity of the user (that is, attributes), via the credential, because the credential user is aware about the identity of

the user. As a consequence, in this case both TTP and the credential user become able to link the user identity to the online content being evaluated, but they cannot guess the score of the online content. The only remainder possibility is that the collusion uses covert channels [63]. It is a matter of fact that covert channel can be used to break unlinkability of protocols guaranteeing anonymity (see, for example, [5]). In the cover channel that we can figure out in this case, TTP acts as an attacker by using a different pair of asymmetric keys for every user it wants to trace. This results in a linkage between the evaluation record and the signed Like ballot. If this covert-channel attack is done together with a collusion between TTP and a credential user described above, then TTP becomes able to link the user with her Like score. Indeed, thanks to the collusion TTP, it links the user with the message ID_R, while thanks to the covert-channel attack it links ER with the Like score s. However this attack is prevented. Indeed, the protocol requires all evaluation records are signed by TTP with the same key pair[2]. Thus, the user can detects the attack by comparing the key pair used by TTP to sign her Like ballot and that used to verify any Like ballot published in the Like list related to any online content.

The following theorem states that the protocol satisfies the property SP3 of the security model S.

Theorem 3 (Individual Verifiability). *The probability of occurrence of failure FU1 is* $P(FU1) \approx 1 - e^{-u^2/2D}$*, where* u *is the number of users and* D *is the cardinality of the domain of the random* r *included in the Like ballot.*

Proof.. The proof can be done as in item (4) of the proof of Theorem 1. As remarked at the end of Theorem 1, in real-life cases the probability stated in the theorem above is negligible, thus individual verifiability is satisfied.

The next theorem states that the protocol satisfies the property SP4 of the security model S.

Theorem 4 (Uncloneability). *The protocol is resistant to attack AC1.*

Proof. Recall that AC1 occurs when the adversary tries to forge a valid ballot by starting from a legal one. We prove that the bogus ballot is always detected. We observe that a valid Like ballot is accompanied by the TTP's

[2]In a real-life implementation of the protocol, we could allow key substitution, but we require anyway that keys are long-term, still preventing the covert-channel attack.

signature and thus any modification made to it is detected by the signature verification, or it amounts to a successful forgery of the underlying signature scheme which is proved to be secure [87]. Obviously, it cannot be duplicated thanks to the presence of the bit-sequence r identifying the Like ballot. About a possible collision between the 128-bit sequence r in two different ballots, we have proved that the probability of this event is negligible.

The next theorem states that the protocol satisfies the property SP5 of the security model S.

Theorem 5 (Robustness). *The protocol is resistant to attack AR1.*

Proof. Recall that AR1 occurs when t or less credential users collude to break security properties of the protocol. We prove that, whenever at most t credential users misbehave, their malicious behavior is detected by TTP. Indeed, the user has to provide $\bar{t} = 2 \cdot t + 1$ credentials and, thus, at least $t + 1$ of them are correct, thanks to the CU-collusion assumption. As a consequence, fake credentials are detected because they are in the minority during the verification phase at Step 4.(a)–(d).

9. Efficiency and Scalability

In this section, we show that the proposed solution is efficient and scalable. First, we observe that our protocol presents good scalability, because the number of users involved in the generation of a single score is independent of the overall number of users. In particular, the Like expression done by V involves a limited number $(2 \cdot t + 1)$ of other users (who play the role of credential users). This allows us to state that the approach is feasible also for social networks with a large number of users and/or online contents. The overall scalability of the system is clearly affected also by the scalability of the underlying anonymous communication system. But, it is well known that highly scalable anonymous communication systems exist [32]. Scalability is also ensured by the adoption of a truly P2P approach to distribute the evaluation process over social network users. In particular, the look-up service (TLS) used in our implementation is proven in [16] to have good performances, also in comparison with the most known lookup proposals. Specifically, the strong advantage obtained is to pull down the insertion/deletion cost from the state-of-the-art $O(\log^2 n)$ to $O(\log n)$. Per-

Technique	Join/Leave	Space	Hops
CHORD [77]	$O(\log^2 n)$	$O(\log n)$	$O(\log n)$
CAN [65]	$O(d)$	$O(nd)$	$O(dn^{1/d})$
Pastry [67]	$O(\log^2 n)$	$O(\log n)$	$O(n \log n)$
Tapestry [36]	$O(\log^2 n)$	$O(\log n)$	$O(\log n)$
TLS	$O(\log n)$	$O(\log n)$	$O(\log n)$

Table 3: Performances of the adopted look-up service TLS.

formances of other operations demonstrate the high efficiency of TLS, as shown in Table 3.

Therein, the column *Join/Leave* reports the costs of peer inserting/deleting, *Space* concerns to the storage information amount required for each peer, *Hops* is the routing cost per message, n is the number of peers in the system and d is the number of dimensional coordinates used in CAN [65]. Moreover, we observe that TLS supports broadcasting in $O(\log n)$ time by exploiting the tree structure.

Concerning our cryptographic protocol, we observe that it was already shown in [87] that the underlying partially blind signature [87] was more efficient than the only other state-of-the-art partially blind signature [2]. In [87], they had proposed another scheme which seemed to be more efficient but its security was based on the ROS problem [71] which has since been proven unsecure [84]. Since [87], there have been some other schemes which have been proposed [26, 49, 38, 35, 61, 68, 9] but those too have been either proven to be insecure [26, 49, 88, 41] or too inefficient [9] or are not applicable [35] to our proposed scheme.

We present a comparison of the state-of-the-art partially blind signatures in the following table. In the comparison table, $\lambda = \log q$. **ROM** stands for the random oracle model and **SM** stands for the standard model. **DLP** represents the discrete logarithm problem in the group $\mathbb{Z}_q^*$ and **FAC** represents the factoring problem for a composite integer $n \approx O(q)$. **2SDH** represents the 2SDH assumption which is a significantly stronger assumption than the discrete logarithm problem [62]. **Exp** denotes the exponentiation modulo p. **Pair** denotes the pairing computation for a cryptographic pairing which provides an equivalent security as that of the DLP on $\mathbb{Z}_q^*$. We note that one such pairing operation costs more than 4 times than a modular exponentiation [35]. For comparison, we consider only these two operations since these are the most time-consuming and leave out all other operations whose times

are relatively insignificant.

Scheme	Signer	User	Verification	bit-length	Security
PBS-AO [2]	2 Exp	4 Exp	4 Exp	4 λ	ROM+DLP
PBS-CH [26]	2 Exp	3 Exp	2 Pair ($\approx$ 8 Exp)	2 λ	ROM+DLP
PBS-OS [62]	11 Exp	7 Exp	3 Exp + 2 Pair ($\approx$ 11 Exp)	3 λ	SM+2SDH
PBS-TS [61]	2 Exp	3 Exp	4 Exp	3 λ	ROM+DLP+FAC
PBS-OR [87]	2 Exp	3 Exp	3 Exp	3 λ	ROM+DLP

Table 4: Comparison of the state-of-the-art partially blind signatures.

From the efficiency comparison in Table 4, it is clear that that the underlying partially blind signature [87] used in our scheme is more efficient than other state-of-the-art partially blind signature schemes [2, 26, 62, 61].

10. Discussion and Conclusion

Among digital records of human behavior, social network Likes are probably the most suitable to automatic analysis aimed at predicting even sensitive personal data of users. Thus, Likes induce serious problems of privacy of which the most users are completely unaware. On the other hand, a lot of analysis on Likes could be done without invading users' privacy, by relating Likes to (even sensitive) attributes of users but keeping them non identifiable. Unfortunately, it is not realistic to assume the trustworthiness of the party responsible for the analysis as well as of the social network provider itself. In this paper, we have proposed a solution allowing the user to choose, when submitting a Like, those attributes she wants to relate to the Like, in such a way that no way exists for any third party (including the social network provider) to link the Like to the identity of its author. Specifically, we have defined a cryptographic-based protocol demonstrating its security and implementing it using XML. The advantage of the protocol, w.r.t. a standard anonymization solution, is that privacy-preserving analysis of Likes is allowed. In other words, concerning Likes, we offered a possible balancing between users' privacy requirements and the utility of extracting strategic knowledge from social-network data. Observe that the above goal cannot be reached if digital records of Likes are managed and stored by the social network provider. Indeed, trustworthiness of the social-network issuer cannot be assumed in general. To solve this problem, our solution relies on a DHT-based P2P social network (assumed given). Anyway, we stress the concept that the adoption of our model of Likes does not require a revolution of the current social networks, moving from the centralized model to

the fully distributed one, since we could just distribute the functions related to the evaluation process and not the evaluated contents, possibly relying on cloud computing solutions.

A limitation of our study is that our solution cannot be applied to existing social networks without changes, even though only the functions related to the Like expression (not the contents) should be decentralized. Anyway, this can be done also by relying on (self-managed) cloud computing solutions. Another limitation is that social network providers could be reluctant to implement the above changes given the lucrative opportunity to trace the big data of social activities. Anyway, this limitation does not appear severe if we assume a perspective view. As observed earlier, what today might appear little appealing from a business point of view, tomorrow could become attractive. The core question here is the weight that privacy will have in future business models. We expect that people will perceive as critical the risks related to an uncontrolled exposure of private life over the Internet. As a consequence, the attention of users towards the information leakage related to the use of social networks will increase. In this plausible evolution scenario, a social network offering advanced tools to protect privacy could acquire advantages w.r.t. competitors. Furthermore, our solution preserves a good level business analytics without allowing the tracing of individuals.

Concerning implementation, besides the aspect of decentralization discussed above, we may draw the conclusion that our paper gives all the guidelines for a fully implementation not using proprietary solutions, and relies on cryptographic primitives whose security is well-accepted.

As a future work, we plan to experiment our solution in a real-life domain, in an industrial project where interested companies are involved. Therein, we are designing an ad-hoc prototype social network operating in the e-learning context. Here, the opinion of users (both students and teachers) should be analyzed as useful feedback but the privacy of the evaluation records is very critical.

Acknowledgment

This work has been partially supported by the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research and by the Program "Programma Operativo Nazionale Ricerca

e Competitività" 2007-2013, Distretto Tecnologico CyberSecurity funded by the Italian Ministry of Education, University and Research.

References

[1] Race integrity primitives evaluation (ripe): final report, RACE 1040, 1993.

[2] M. Abe and T. Okamoto. Provably secure partially blind signatures. In *Advances in CryptologyCRYPTO 2000*, pages 271–286. Springer, 2000.

[3] Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In *Advances in CryptologyASIACRYPT'96*, pages 244–251. Springer, 1996.

[4] A. Acquisti. Receipt-free homomorphic elections and write-in ballots. *Tech. Rep. 2004/105*, 2004.

[5] Mikaël Ates, Francesco Buccafurri, Jacques Fayolle, and Gianluca Lax. A warning on how to implement anonymous credential protocols into the information card framework. *International Journal of Information Security*, 11(1):33–40, February 2012.

[6] Eric Bach. *Discrete logarithms and factoring*. Computer Science Division, University of California, 1984.

[7] Dylan Bedrossian, Anas Harb, Rawad Khalil, Ali Chehab, and Hassan Ali Artail. server-less social network for enhanced privacy. *Procedia Computer Science*, 34:95–102, 2014.

[8] Elisa Bertino, Elena Ferrari, and A Squicciarini. Privacy-preserving trust negotiations. In *Privacy Enhancing Technologies*, pages 763–771. Springer, 2005.

[9] Olivier Blazy, David Pointcheval, and Damien Vergnaud. Compact round-optimal partially-blind signatures. In *Security and Cryptography for Networks*, volume 7485 of *LNCS*, pages 95–112. 2012.

[10] F. Buccafurri, L. Fotia, and G. Lax. Allowing continuous evaluation of citizen opinions through social networks. *Advancing Democracy, Government and Governance - Joint International Conference on Electronic Government and the Information Systems Perspective, and Electronic Democracy (EGOVIS/EDEM 2012)*, pages 242–253, 2012.

[11] F. Buccafurri, L. Fotia, and G. Lax. A privacy-preserving e-participation framework allowing citizen opinion analysis. *Electronic Government, An International Journal*, 11:185–206, 2015.

[12] Francesco Buccafurri, Lidia Fotia, and Gianluca Lax. Privacy-preserving resource evaluation in social networks. In *Proceedings of the 2012 Tenth Annual International Conference on Privacy, Security and Trust (PST 2012)*, pages 51–58. IEEE Computer Society, 2012.

[13] Francesco Buccafurri, Lidia Fotia, and Gianluca Lax. Allowing non-identifying information disclosure in citizen opinion evaluation. In *Technology-Enabled Innovation for Democracy, Government and Governance*, pages 241–254. Springer, 2013.

[14] Francesco Buccafurri, Lidia Fotia, and Gianluca Lax. Allowing privacy-preserving analysis of social network likes. In *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*, pages 36–43. IEEE, 2013.

[15] Francesco Buccafurri, Lidia Fotia, and Gianluca Lax. Social signature: Signing by tweeting. In *Electronic Government and the Information Systems Perspective*, pages 1–14. Springer, 2014.

[16] Francesco Buccafurri and Gianluca Lax. Tls: A tree-based dht lookup service for highly dynamic networks. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 563–580. Springer, 2004.

[17] S. Buchegger and A. Datta. A case for p2p infrastructure for social networks-opportunities & challenges. In *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on*, pages 161–168. IEEE, 2009.

[18] S. Buchegger, D. Schiöberg, L.H. Vu, and A. Datta. Peerson: P2p social networking: early experiences and insights. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 46–52. ACM, 2009.

[19] M. Burmester and E. Magkos. Towards secure and practical e-elections in the new era. *Secure Electronic Voting*, pages 63–76, 2003.

[20] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. *Advances in Cryptology–EUROCRYPT 2005*, pages 566–566, 2005.

[21] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *Advances in Cryptology–EUROCRYPT 2001*, pages 93–118, 2001.

[22] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology–CRYPTO 2004*, pages 1–6. Springer, 2004.

[23] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable signatures: New definitions and delegatable anonymous credentials. In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*, pages 199–213. IEEE, 2014.

[24] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[25] D.L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[26] ShermanS.M. Chow, LucasC.K. Hui, S.M. Yiu, and K.P. Chow. Two improved partially blind signature schemes from bilinear pairings. In *Information Security and Privacy*, volume 3574 of *LNCS*, pages 316–328. 2005.

[27] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. *European Transactions on Telecommunications*, 8(5):481–490, 1997.

[28] L.A. Cutillo, R. Molva, and T. Strufe. Privacy preserving social networking through decentralization. In *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on*, pages 145–152. IEEE, 2009.

[29] I. Damgård and M. Jurik. A generalisation, a simpli. cation and some applications of paillier's probabilistic public-key system. In *Public Key Cryptography*, pages 119–136. Springer, 2001.

[30] Bert den Boer and Antoon Bosselaers. Collisions for the compression function of md5. In *Advances in CryptologyEUROCRYPT93*, pages 293–304. Springer, 1994.

[31] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

[32] Matthew Edman and Bülent Yener. On anonymity in an electronic society: A survey of anonymous communication systems. *ACM Comput. Surv.*, 42(1):5:1–5:35, December 2009.

[33] Chun-I Fan and LEI Chin-Laung. Low-computation partially blind signatures for electronic cash. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 81(5):818–824, 1998.

[34] P.A. Fouque, G. Poupard, and J. Stern. Sharing decryption in the context of voting or lotteries. In *Financial Cryptography*, pages 90–104. Springer, 2001.

[35] Zheng Gong, Yu Long, and Kefei Chen. Efficient partially blind signature from lfsr. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD)*, volume 2, pages 717–722, 2007.

[36] Kirsten Hildrum, John D. Kubiatowicz, Satish Rao, and Ben Y. Zhao. Distributed object location in a dynamic network. In *Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures*, pages 41–52. ACM Press, 2002.

[37] Jason E Holt and Kent E Seamons. Selective disclosure credential sets. *IACR Cryptology ePrint Archive*, 2002:151, 2002.

[38] Xiaoming Hu and Shangteng Huang. An efficient id-based partially blind signature scheme. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD)*, volume 3, pages 291–296, 2007.

[39] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 61–70. ACM, 2005.

[40] Justin Jouvenal. A Facebook court battle: Is liking something protected free speech? `http://www.washingtonpost.com/local/crime/a-facebook-court-battle-is-liking-something-protected-free-speech/2012/08/08/538314fe-e179-11e1-ae7f-d2a13e249eb2\_story.html`, 2012.

[41] Baoyuan Kang and Jinguang Han. On the security of blind signature and partially blind signature. In *International Conference on Education Technology and Computer (ICETC)*, volume 5, pages 206–208, 2010.

[42] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In *Advances in Cryptology-Eurocrypt 2004*, pages 571–589. Springer, 2004.

[43] Aggelos Kiayias and Moti Yung. Secure scalable group signature with dynamic joins and separable authorities. *International Journal of Security and Networks*, 1(1):24–45, 2006.

[44] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 2013.

[45] VK Narendira Kumar and B Srinivasan. A practical privacy preserving e-voting scheme with smart card using blind signature. *International Journal of Computer Network and Information Security (IJCNIS)*, 5(2):42, 2013.

[46] PANG Lei, Mao-hua SUN, Shou-shan LUO, WANG Bai, and XIN Yang. Full privacy preserving electronic voting scheme. *The Journal of China Universities of Posts and Telecommunications*, 19(4):86–93, 2012.

[47] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

[48] Bin Lian, Gongliang Chen, and Jianhua Li. Provably secure e-cash system with practical and efficient complete tracing. *International journal of information security*, 13(3):271–289, 2014.

[49] Jian Liao, Yinghao Qi, Peiwei Huang, and Mentian Rong. Pairing-based provable blind signature scheme without random oracles. In *Computational Intelligence and Security*, volume 3802 of *LNCS*, pages 161–166. 2005.

[50] Benoît Libert and Marc Joye. Group signatures with message-dependent opening in the standard model. In *Topics in Cryptology–CT-RSA 2014*, pages 286–306. Springer, 2014.

[51] Phone Lin, Pai-Chun Chung, and Yuguang Fang. P2p-isn: a peer-to-peer architecture for heterogeneous social networks. *Network, IEEE*, 28(1):56–64, 2014.

[52] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. *Journal of cryptology*, 15(3):177–206, 2002.

[53] Heather Richter Lipford, Pamela J Wisniewski, Cliff Lampe, Lorraine Kisselburgh, and Kelly Caine. Reconciling privacy with social media. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*, pages 19–20. ACM, 2012.

[54] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[55] Alireza Mahdian, Richard Han, Qin Lv, and Shivakant Mishra. Results from a practical deployment of the myzone decentralized p2p social network. *arXiv preprint arXiv:1305.0606*, 2013.

[56] Meltwater. Likealyzer: Analyze and monitor your facebook pages. `http://likealyzer.com`, 2015.

[57] Florian Mendel, Tomislav Nad, Stefan Scherz, and Martin Schläffer. Differential attacks on reduced ripemd-160. In *Information Security*, pages 23–38. Springer, 2012.

[58] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC, 1997.

[59] Moni Naor. Bit commitment using pseudorandomness. *Journal of cryptology*, 4(2):151–158, 1991.

[60] National Institute of Standards and Technology (NIST). FIPS-180-2: Secure Hash Standard. `http://www.itl.nist.gov/fipspubs/`, August 2002.

[61] E.S. Ismail N.M.F. Tahat, S.M.A. Shatnawi. A new partially blind signature based on factoring and discrete logarithms. *Journal of Mathematics and Statistics*, 4:124–129, 2008.

[62] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In *Theory of Cryptography*, volume 3876 of *LNCS*, pages 80–99. 2006.

[63] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn. Information hiding-a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[64] J. Pieprzyk, T. Hardjono, and J. Seberry. *Fundamentals of computer security*. Springer Verlag, 2003.

[65] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172. ACM Press, 2001.

[66] Ronald Rivest. The md5 message-digest algorithm. 1992.

[67] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 329–350. Springer-Verlag, 2001.

[68] Markus Ruckert and Dominique Schroder. Fair partially blind signatures. In *Progress in Cryptology – AfricaCrypt 2010*, volume 6055 of *LNCS*, pages 34–51. 2010.

[69] Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information. In *PODS*, volume 98, page 188, 1998.

[70] Bruce Schneier and Phil Sutherland. *Applied cryptography: protocols, algorithms, and source code in C*. John Wiley & Sons, Inc., 1995.

[71] ClausPeter Schnorr. Security of blind discrete log signatures against interactive attacks. In *Information and Communications Security*, volume 2229 of *LNCS*, pages 1–12. 2001.

[72] R. Sherwood, B. Bhattacharjee, and A. Srinivasan. P5: A protocol for scalable anonymous communication. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 58–70. IEEE, 2002.

[73] C. Shields and B.N. Levine. A protocol for anonymous communication over the internet. In *Proceedings of the 7th ACM conference on Computer and communications security*, pages 33–42. ACM, 2000.

[74] Sooyeon Shin and Taekyoung Kwon. Aana: Anonymous authentication and authorization based on short traceable signatures. *International Journal of Information Security*, 13(5):477–495, 2014.

[75] SocialMediaAnalytics. Analytics platforms. `http://www.social-media-analytics.it/social-media-analysis/facebook-da-dove-provengono-i-mi-piace`, 2015.

[76] La Stampa. Clicca 'mi piace' su facebook rischia condanna per diffamazione. `http://www.lastampa.it/2014/01/30/italia/cronache/clicca-mi-piace-su-facebook-rischia-condanna-per-diffamazione-eq6u2JO0SNlAM43L25bhdI/pagina.html`, 2015.

[77] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160. ACM Press, 2001.

[78] Ai Thao Nguyen Thi and Tran Khanh Dang. Privacy preserving in electronic voting. *Electrical and Electronics Engineering Computer and Information Engineering*, page 28, 2014.

[79] Tor Project. Tor Metrics: Users. `https://metrics.torproject.org/users.html`, 2014.

[80] Blase Ur and Yang Wang. A cross-cultural framework for protecting user privacy in online social media. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 755–762. International World Wide Web Conferences Steering Committee, 2013.

[81] Jaideep Vaidya, Christopher W Clifton, and Yu Michael Zhu. *Privacy preserving data mining*, volume 19. Springer, 2005.

[82] Jessica Vitak. The impact of context collapse and privacy on social network site disclosures. *Journal of Broadcasting & Electronic Media*, 56(4):451–470, 2012.

[83] W3Schools.com. Xquery tutorial. `http://www.w3schools.com/xquery/`, 2013.

[84] David Wagner. A generalized birthday problem. In *Advances in Cryptology CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–304. 2002.

[85] F. Wang, Y. Moreno, and Y. Sun. Structure of peer-to-peer social networks. *Physical Review E*, 73(3):036123, 2006.

[86] Pamela Wisniewski, AKM Islam, Bart P Knijnenburg, and Sameer Patil. Give social network users the privacy they want. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1427–1441. ACM, 2015.

[87] Q. Wu, W. Susilo, Y. Mu, and F. Zhang. Efficient partially blind signatures with provable security. *Computational Science and Its Applications-ICCSA 2006*, pages 345–354, 2006.

[88] T.Y. Wu Y.M Tseng and J.D Wu. On the security of an efficient id-based partially blind signature scheme. In *International MultiConference of Engineers and Computer Scientists (IMECS)*, volume 1, pages 417–420, 2008.

[89] A. Zwierko and Z. Kotulski. A light-weight e-voting system with distributed trust. *Electronic Notes in Theoretical Computer Science*, 168:109–126, 2007.