A Layered Secret Sharing Scheme for Automated Profile Sharing in OSN Groups

Guillaume Smith[1,2,3(✉)], Roksana Boreli[1,2], and Mohamed Ali Kaafar[1,4]

[1] NICTA, Sydney, Australia
[2] University of New South Wales, Sydney, Australia
[3] ISAE - Université de Toulouse, Toulouse, France
[4] INRIA, Paris, France
guillaume.smith@nicta.com.au

Abstract. We propose a novel Layered secret sharing scheme and its application to Online Social Networks (OSNs). In current, commercially offered OSNs, access to users' profile information is managed by the service provider e.g. Facebook or Google+, based on the user defined privacy settings. A limited set of rules such as those governing the creation of groups of friends as defined by the user (e.g. circles, friend groups or lists) allow the users to define different levels of privacy, however they are arguably complex and rely on a trusted third party (the service provider) to ensure compliance. The proposed scheme enables automated profile sharing in OSN groups with fine grained privacy control, via a multi-secret sharing scheme comprising layered shares, created from user's profile attributes (multiple secrets), that are distributed to group members; with no reliance on a trusted third party. The scheme can be implemented via e.g. a browser plugin, enabling automation of all operations for OSN users. We study the security of the scheme against attacks aiming to acquire knowledge about user's profile. We also provide a theoretical analysis of the resulting level of protection for specific (privacy sensitive) attributes of the profile.

1 Introduction

Secret sharing schemes have been extensively used in a number of cryptographic and distributed computing applications [3]. Using secret sharing, a dealer shares a secret (any content, be it a numerical value, text string or a file) in a secure way, by creating a number of shares (cryptographic derivatives of the secret) and distributing them to a set of participants. A participant can access the secret by combining a specific number of shares, that has to be greater than the threshold.

In this paper, we consider the use of secret sharing to enable user-controlled and privacy preserving sharing of an Online Social Network (OSN) user's profile amongst groups of their direct connections (friends). In most commercially available OSNs, an OSN user currently stores their private data using the trusted OSN server. Access to their profile is managed by a policy, regulating which attributes (e.g. their location, date of birth, interests, etc.) may be obtained by

I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 487–499, 2014.
DOI: 10.1007/978-3-319-11569-6_38

their friends or other people. A number of challenges related to privacy in OSNs have been identified by researchers [10], including the reliance on the trusted third party (TTP) to manage the privacy control, the arguable complexity of how those controls are implemented in current OSNs and the limited flexibility of access controls (we note Google+ circles and Facebook lists are a step in the direction of improving the flexibility). This motivates our interest in using secret sharing to enable direct and fine grained user control of the accessibility of their private data.

We specifically consider the OSN user's requirements to share multiple profile attributes (secrets) with differing levels of security and privacy (i.e. thresholds), within groups of friends, without relying on a TTP. Additionally, an OSN user may not wish to disclose his level of desired privacy protection, and/or the difference in the number of attributes he is sharing with different friends (or groups of friends). The limitations of the currently proposed secret sharing schemes relate to both threshold flexibility and disclosure of scheme's parameters [12,20], which could be used to re-identify anonymous data [11].

We propose a novel Layered secret sharing scheme, that recursively embeds shares related to individual secrets in layers with increased protection, and encrypts each layer with a shares-based key. Each secret is then protected by a selected single secret-sharing scheme with its own threshold, and the number of secrets is hidden within the Layered scheme, which ensures that the thresholds remain unknown. The main contributions of this paper are as follows:

We **propose a new Layered secret sharing scheme** that enables flexible levels of security and privacy. We introduce **privacy-preserving automated profile sharing in OSN groups** as a possible use of our scheme. By generating Layered shares (comprising a selected set of attributes, corresponding to each group's privacy policy) and distributing a share to each member of the group, an OSN user automatically enforces the deployment of group's privacy policy and enables fine grained policy control within group's members, without relying on a TTP.

We **analyse the security of the scheme** for attacks that have a goal of illegitimately acquiring knowledge about users OSN profiles. Consequently, we show that no new Layered shares can be attained by any of the attacks that include a varying level of background knowledge. We provide an **analysis of the number of Layered shares** that an **attacker** who is at an arbitrary number of hops in the social graph from the node sharing his profile **may acquire**, which can be used to enable the profile owner to set the required level of protection offered to specific (privacy sensitive) attributes of the profile.

2 Background and Related Work

In a secret sharing scheme, a dealer securely shares a secret with a group of participants, by generating n shares using a cryptographic function. These are distributed to n participants; by aggregating shares, participants can gain access to the secret when the number of combined shares reaches the threshold t.

We note that the threshold needs to be distributed jointly with the shares. The most commonly used scheme, proposed by Shamir [14], is based on polynomial interpolation. It has the following properties:

Property S.1. *Shamir's scheme is information theoretically secure, that is, for a* (t, n) *scheme, knowledge of less than* t *shares does not provide any information about the remaining shares or the secret.*

Property S.2. *The secret is equivalent to a share, as it is a specific point on the curve defined by the polynomial used to both generate and decode shares.*

Shamir's scheme introduces a significant overhead as shares are of the same size as the secret [13]. Ramp schemes [19] have therefore been proposed to provide a trade-off between the size of the shares and the security of the scheme.

A generalized ramp scheme is defined by (t, L, n), where the parameter L relates to security, i.e. no information about the secret can be obtained if a participant has $x \leq (t - L)$ shares. In all ramp schemes, the size of the shares for a secret S, is $\frac{|S|}{L}$. [19] introduces the concepts of weak and strong ramp schemes. A strong ramp scheme with $L = t$ (used in our work) has the following properties:

Property R.1. *The ramp scheme is computationally secure, i.e. for a* (t, n) *scheme, possession of* $x < t$ *(less than* t *shares) does not provide any information about any remaining (unknown) shares and any part of the secret.*

Property R.2. *Knowledge of the secret in the ramp scheme enables reconstruction of all the related shares (coordinates used for shares computation are known).*

2.1 Sharing Multiple Secrets

A naive approach to share multiple secrets is to use, multiple times, a single secret sharing scheme. That may prove onerous in both terms of communication and computational overhead as each participant needs, for each shared secret, to receive and store a corresponding share [4]. Considering privacy, this approach will expose the number of shared secrets that is equal to the number of shares.

Multiple secret sharing schemes (e.g. [7,12,20]) have then been introduced with the objective of allowing one share per participant for all the secrets shared by a single user. From a profile sharing perspective, and more specifically when considering the different privacy levels a user may consider for each attribute, these schemes do not provide the desirable flexibility, as a profile's owner would not be allowed to parametrize the level of privacy for each secret (attribute).

A specific class of schemes consists of designs that allow a different threshold per secret (e.g. [7,17,18]). When used in profile sharing, each participant will have a priori knowledge about the number of shared secrets along with their corresponding thresholds, which doesn't fit the needs of users to not reveal how much of their personal information they are actually sharing.

2.2 Profile Sharing in OSNs

A possible approach to selective sharing of different information on popular OSNs, e.g. Facebook or Google+ is to define groups of friends (lists or circles) and set up a different privacy policy for each group. Alternatively, in order to prevent a central entity from accessing user data, [1] proposes an architecture where user data is encrypted and stored on an untrusted server. Other proposals e.g. *PeerSon*, [5], aim to distribute the online social network itself.

In practice, users may not be easily convinced to change from the well-established centralized OSNs only due to privacy concerns. References [8] and [2] e.g. specifically address user's privacy in the context of major OSNs, by providing browser-based plugins in order to enable user's control over their personal data.

Although there is user support for manually configured group based access control in OSNs, researchers have shown [6] that the majority of OSN users do not utilize this feature. Consequently automated algorithms for grouping friends (*e.g.* based on common interests, family relationship, etc.) have been studied e.g. in [21]. However, although the access control can be defined for groups (circles) of friends, all users in the same group share a common access policy. Our proposed Layered secret sharing scheme, as we will show, enables a finer grained access control policy that is based on the number of common friends, and automates the enforcement of this policy in each circle. This is implemented without having to rely on a trusted third party (i.e. the OSN operator).

3 Layered Secret Sharing Scheme

A dealer in our scheme generates Layered shares, which consist of a number of encrypted layers, each comprising (standard single secret sharing) shares related to a specific secret. Figure 1 shows the components of a Layered share.

The notations used in this paper are listed in Table 1. A dealer shares k secrets with n participants P_j $j = 1, ..., n$. The secrets S_i, $i = 1, ..., k$, have t_i corresponding thresholds. We assume these are ordered in terms of increasing magnitudes, with $(t_1 \leq t_2 \leq ... \leq t_k)$ and that the secret S_1 is the system secret,

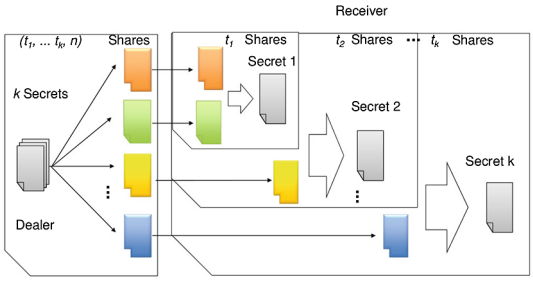

Fig. 1. Components in the Layered secret sharing scheme

Table 1. Notations used to describe the Layered secret sharing scheme

n	Number of Layered shares and participants
P_j	Participant j for $j = 1, ..., n$
k	Number of secrets shared (including the system secret)
S_i	Secret i, for $i = 1, ..., k$
$H(.)$	One-way hash function, publicly known
$[.]_K$	Symmetric encryption using the key K
$[.]^K$	Symmetric decryption using the key K
$[A\|\|B]$	Concatenation of content A and B
s_j^i	Share j related to the secret S_i
L_j^i	Layer i of the Layered share, containing the shares related to S_i to S_k
K_i	Key derived from the secret S_i
t_i	Threshold for the secret S_i
b_i	Number of additional shares needed to compute the key K_i

a random variable used for verification of Layered shares. For the use case of OSN profile sharing, S_2 is an OSN group's identifier.

Layered shares L_j^1, $j = 1, ..., n$, are generated according to steps outlined in Function *MakeLayered*. The secrets are included recursively within the Layered shares, starting from the secret with the highest threshold (S_k). At the $(k-i)^{th}$ step, to include the secret S_i in L_j^{i+1} for $j = 1, ..., n$, the dealer computes $n + b_i$ standard shares $(s_1^i, ..., s_{n+b_i}^i)$ with a threshold t_i using the Function *MakeSingle*. *MakeSingle* utilizes either Shamir's scheme [14] or the ramp scheme from [19], with the choice depending on the size of S_i and the required security level.

b_i additional shares $s_{n+1}^i, \cdots, s_{n+b_i}^i$ are generated, then concatenated and hashed, using e.g. *SHA-2* to obtain the key K_i (ensuring the size of concatenation is at least the size of the encryption key, e.g. *AES* needs 128 or 256 bits keys).

Function `MakeLayered`$(S_1, ..., S_k, t_1, ..., t_k)$

Initialization;
for $j \leftarrow 1$ ***to*** n **do**
  $L_j^{k+1} \leftarrow \varnothing$;
Generate n Layered shares by constructing k individual layers;
for $i \leftarrow k$ ***to*** 1 **do**
  MakeSingle produces $n + b_i$ *(single secret sharing) shares according to the selected secret sharing mechanism*
  $(s_1^i, ..., s_{n+b_i}^i) \leftarrow MakeSingle(S_i, t_i, scheme)$;
  $K_i \leftarrow H([s_{n+1}^i || \cdots || s_{n+b_i}^i])$;
  for $j \leftarrow 1$ ***to*** n **do**
    $L_j^i \leftarrow [t_i || s_j^i || [L_j^{i+1}]_{K_i}]$;

For $j = 1, ..., n$, each share s_j^i related to secret S_i is concatenated with the corresponding encrypted layer $[L_j^{i+1}]_{K_i}$: $L_j^i = [t_i||s_j^i||[L_j^{i+1}]_{K_i}]$. These steps are repeated until all secrets are included in the layers.

The n Layered shares are distributed by the dealer, with a single unique share and the share's index (j, L_j^1) sent to each participant over a secure channel. As in other secret sharing mechanisms, participants aggregate their shares to gain access to the secret(s) S_i, with the number of shares as per the threshold t_i.

To describe the decoding process by which a participant recovers one or more secrets, we assume that he has acquired x Layered shares, L_z^1 where $z = 1, ..., x$ (to simplify the notation, without loss of generality we assume that Layered shares are received in the order of their indexes). We also assume that the participant has previously decoded i secrets $S_1, ..., S_i$ with thresholds $t_1, ..., t_i$ and their corresponding keys $K_1, ..., K_i$, where i is defined by $t_i \leq x - 1 < t_{i+1}$, using the previously received $x - 1$ Layered shares ($i = 0$ if no secrets have been decoded). Thus he has $x-1$ layers L_z^{i+1}, where $z = 1, ..., x-1$ and a Layered share L_x^1 that is yet to be processed. Function *DecodeLayered* describes the decoding process. In the first step, the "old" layers related to the previously decoded secrets need to, recursively, be removed and decrypted (using $K_1, ..., K_i$) from L_x^1. If $x == t_{i+1}$, function *DecodeSingle* can then be used to decode the secret S_{i+1} and also to compute as previously the key K_{i+1}. Finally the layers related to the secret S_{i+1} are removed from the x Layered shares and the following parts $[L_z^{i+2}]_{K_{i+1}}$ are decrypted, using the key K_{i+1}.

Function `DecodeLayered`(L_x^1)

Remove the layers related to the previously decoded secrets;
for $y \leftarrow 1$ ***to*** i **do**
  $[s_x^y||[L_x^{y+1}]_{K_y}] \leftarrow L_x^y$;
  $L_x^{y+1} \leftarrow [[L_x^{y+1}]_{K_y}]^{K_y}$;
Decode the secret S_{i+1} if there is a sufficient number of shares;
if $x == t_{i+1}$ **then**
  for $j \leftarrow 1$ ***to*** x **do**
    $[s_j^{i+1}||[L_j^{i+2}]_{K_{i+1}}] \leftarrow L_j^{i+1}$;
  DecodeSingle outputs the secret S_{i+1} and the shares $s_{n+1}^{i+1}, \cdots, s_{n+b_i}^{i+1}$;
  $(S_{i+1}, s_{n+1}^{i+1}, \cdots, s_{n+b_i}^{i+1}) \leftarrow$ `DecodeSingle`$(scheme, t_{i+1}, s_1^{i+1}, ..., s_x^{i+1})$;
  Generate the encryption key from the share s_{n+1}^{i+1};
  $K_{i+1} \leftarrow H([s_{n+1}^{i+1}|| \cdots ||s_{n+b_i}^{i+1}])$;
  Remove the layer related to S_{i+1} for all (available) Layered shares;
  for $j \leftarrow 1$ ***to*** x **do**
    $L_j^{i+2} \leftarrow [[L_j^{i+2}]_{K_{i+1}}]^{K_{i+1}}$;

3.1 Using Layered Shares for Profile Sharing in OSN Groups

To initialize the scheme, the dealer must first define circles of friends, C_i, and the attributes (secrets) he intends to share amongst the friends in each circle. He should also allocate friends to circles, either manually or using any of the proposed mechanisms, e.g. [21]. Then, for each circle and each shared attribute, the dealer should choose a privacy level for accessing the attribute (*i.e.* the threshold for each secret), that will, in our scheme, be equivalent to the number of common friends that a user in this circle needs to have with the profile owner. We acknowledge that this criteria may not be universally applicable to all OSN groups. However, we believe, as per [21], that the number of common friends represents a relevant and easy to understand metric for the OSN users.

The dealer then generates Layered shares (using *MakeLayered*) and distributes them to friends in a specific circle over a secure channel. Subsequently, each friend contacts their friends and collects the available Layered shares from the friends in common with the dealer (that also belong to the same circle). Any friend can then decode the accessible attributes according to the number of collected Layered shares, using *DecodeLayered*.

We envisage the profile sharing application to be implemented as a browser plugin, where all the operations related to the shares are automated and secure. We note that for the scheme to be operational, all participating OSN users need to have the plugin installed.

For new friend connection requests, the dealer first needs to select the appropriate circle for this new friend. Then, he needs to generate a new Layered share and index and send it over a secure channel to the new friend.

Finally, if a user wishes to remove a friend from a circle or change his circle, he has to renew all Layered shares for this circle, to ensure that his Layered share will not be usable in future exchanges. An alternative would be to use the proxy re-encryption approach from [9], that does not require share renewal.

4 Security Analysis for the OSN Application

4.1 Adversary Model

The threat scenario consists of adversaries that will attempt to access more information than what they may obtain legitimately i.e. from the Layered shares acquired either directly from the dealer or from the dealer's friend connections. This includes an honest but curious adversary who is an insider (a user) within the system and has access to a number of Layered shares, or a malicious user who obtains all information by monitoring communications of other nodes, by compromising nodes, *etc.*

4.2 Privacy of the Scheme

We assume the attacker has x Layered shares, $x \in [0, n]$. This will enable him access to corresponding secrets S_1 to S_i, decoded from those shares, with i

defined by the threshold $t_i \leq x < t_{i+1}$. The attacker also may have background knowledge of n_S, $n_S \in [0, k]$ secrets, their position in the layers and related thresholds. In the following, we consider the different scenarios related to the position of known secrets and the resulting gain by the attacker.

We consider the use of both Shamir's and the ramp scheme and utilize the properties of these schemes, defined in Sect. 2, in the security analysis.

The Attacker has Access only to Layered Shares: The privacy protection our scheme provides is equivalent to that of the underlying scheme, as defined by properties S.1 and R.1, respectively for Shamir's and the ramp scheme.

Attacker has Access to both Layered Shares and Background Knowledge: We first consider the cases when one of the following secrets, S_{i+1} or S_{i+2} is known, then generalise the analysis to the attacker having access to any number of secrets (not acquired from the available Layered shares).

The adversary knows secret S_{i+1} in addition to Layered shares: *For Shamir's scheme*, knowledge of the secret is equivalent to having access to an additional share related to S_{i+1} (see property S.2). If the new number of shares is below t_{i+1}, as per the property S.1, the attacker cannot progress further. If the threshold was reached, the attacker can recompute all related shares; consequently (from the additional shares), he can acquire the key K_{i+1} required to decrypt the next layer of shares within the available Layered shares.

For the ramp scheme, knowledge of the secret and the threshold t_{i+1}, as per property R.2, enables the attacker only to reconstruct the key K_{i+1} and access the next level of shares in the Layered scheme. Similarly to the case when Shamir's scheme is used, the attacker can, as the maximum gain, succeed in having access to x shares in the next layer of the Layered shares.

The adversary knows secret S_{i+2} together with Layered shares: Here, the adversary aims to access the unknown secret S_{i+1} and potentially further information about the remaining secrets.

For Shamir's scheme (re. S.2), there is no additional gain.

For the ramp scheme, (as per R.2) the attacker can reconstruct all the shares related to S_{i+2}. This would enable him access to K_{i+2} for decrypting the next level of shares in the Layered scheme, related to the secret S_{i+3}. But first the attacker needs to reconstruct the key K_{i+1} to be able to decode the Layered shares L_j^{i+1} and then the Layered shares L_j^{i+2} using the key K_{i+2}.

A plaintext attack, in which the adversary partially knows the decrypted text string, i.e. shares related to S_{i+2}, could be used to gain K_{i+1} (we note that, to the best of our knowledge, there has been as yet no successful plaintext attack on *AES*). Assuming this is successful, the attacker would obtain K_{i+1}, i.e. a cryptographic hash of b_{i+1} shares related to S_{i+1}. Assuming the attacker has sufficient computational resources to derive these shares from the one-way hash function, the resulting gain would be b_{i+1} additional shares related to S_{i+1}; if the threshold t_{i+1} is reached, the attacker will gain the additional secret S_{i+1}.

The adversary knows any number of subsequent secrets: For any case where the next layer of shares (in the Layered scheme) is protected by Shamir's scheme, and assuming the secret is known to the attacker, the maximum possible

gain can be the availability of the key to decrypt the next layer. When the next layer is protected by the ramp scheme, the certain gain is the same key. When the second-next layer is based on Shamir's secret sharing, assuming a known secret, there is no possible gain; when it is based on the ramp scheme, again with a known secret, both the previous layer and the following layer may be decrypted. We note that no new Layered shares can be gained by the attacker with an arbitrary level of background information.

4.3 Analysis of Attacker's Access to Layered Shares

We now provide an analysis of the number of Layered shares that an adversary potentially may have access to, in the OSN scenario where a user is sharing his profile with direct friends.

We assume that an adversary is a node belonging to the social graph of the user and that he obtains information (shares) from his direct connections with a probability p_{j-1}, where $j > 1$ is the distance in hops between the attacker and the profile's owner. A node at a distance j from the owner is also referred as a j-degree friend. Second, we assume that the probability that an adversary will obtain shares decreases with the distance from the node sharing his profile and we only consider the path with the shortest distance as relevant in obtaining information (assuming that the probability to obtain shares on a longer path is negligible compared to the shortest path).

We assume that the direct friends of the owner follow the protocol and, although they may be tricked by the attacker, they would release only their own (single) Layered share, with a probability p_1. All other nodes (on the social graph of the profile owner) who have shares are adversaries. Therefore an attacker at a distance $j+1$ will receive all available Layered shares from a j-degree friend of the owner ($j > 1$) with a probability p_j.

In the following analysis, we derive the probability that an attacker amongst the $j+1$-degree friend with $j \geq 1$ will obtain x Layered shares, assuming the owner has n direct friends: $P_j(X = x \mid N = n)$. To this purpose, we consider two cases: we provide a detailed analysis for a close attacker, with $j = 1$ and we generalize the analysis for $j \geq 2$.

First, we consider an *attacker who is at a distance two from the profile owner.* Let $\mathscr{A}_1$ be the set of common friends of the attacker and the profile's owner of size A_1. The attacker will successfully acquire x Layered shares from $\mathscr{A}_1$ if he receives them from exactly x elements in $\mathscr{A}_1$ with a probability p_1, therefore:

$$P_1(X = x \mid N = n) = \sum_{a_1=x}^{n} P(A_1 = a_1 \mid N = n) \binom{a_1}{x} p_1^x (1-p_1)^{a_1-x} \quad (1)$$

The distribution $P(A_1 = a_1 \mid N = n)$ can be computed for $a_1 \in [0, n]$ from the social graph of the selected OSN.

For the generic case of a *distant adversary* who is a $j+1$-degree friend with $j \geq 2$, we assume that the nodes at a distance j for $j \geq 2$ receive independent sets of Layered shares (although any share can be in multiple sets), in line with

the assumption that the further two nodes are from the profile's owner, the less chance they have to have common friends. Let A_j be the random variable describing the size of the intersection between the direct friends of the attacker and the j-degree friends of the profile's owner. We can then calculate:

$$P_j(X = x \mid N = n) = \sum_{a_j=0}^{\infty} P(X = x \mid N = n \wedge A_j = a_j) P(A_j = a_j \mid N = n) \tag{2}$$

where $P(A_j = a_j \mid N = n)$ can be computed from the OSN's social graph.

The probability that the attacker receives the Layered shares from k nodes amongst the A_j is equal to $\binom{a_j}{k} p_j^k (1-p_j)^{a_j-k}$. Then the probability that one of the k nodes has x_y Layered shares for $y \in [1,k]$ is equal to $P_{j-1}(X = x_y \mid N = n)$. Finally we need to evaluate the probability that the size of the union of any k sets of size x_y amongst n of Layered shares is equal to x, denoted as $P(|U_1^k| = x \mid x_y)$. The derivation of this probability is included in our technical report[1]. We finally derive the probability that the attacker receives x Layered shares:

$$P(X = x \mid N = n \wedge A_j = a_j) = \sum_{k=1}^{a_j} \binom{a_j}{k} p_j^k (1-p_j)^{a_j-k} \times$$
$$\sum_{y=1}^{k} \sum_{x_y=0}^{n} P(|U_1^k| = x \mid x_y) \prod_{z=1}^{k} P_{j-1}(X = x_z \mid N = n) \tag{3}$$

5 Implementation Considerations

To evaluate the practicality of our proposal, we need to ensure that the number of Layered shares that friends of an OSN user can obtain, is always greater than the minimum threshold required to protect against attacks. To obtain representative OSN parameters, we use the Facebook New Orleans dataset [16], comprising around 60,000 nodes, to derive the distributions required for calculating $P_j(X \geq x \mid N = n)$, as per Sect. 4.3. We choose three levels of resilience: such that less than, respectively, 1 %, 0.1 %, and 0.01 % of the adversaries at selected distances from the dealer are able to access data if corresponding thresholds t_{min}^j are chosen. We also choose two values for the probabilities p_j (5 % and 10 %) that a friend of the dealer will leak information to an adversary and for an adversary to forward all Layered shares to another adversary.

Figure 2 shows the calculated minimum thresholds for selected parameters. This figure also shows the average number of Layered shares that friends of a dealer will be able to acquire (the number of friends is varied between one and 100, the average number of Facebook friends [15]). We can observe that even with a high (10 %) leakage from dealer's friends and other adversaries, the number of available shares that the friends can acquire is greater than the minimum

[1] http://www.nicta.com.au/pub?id=7318

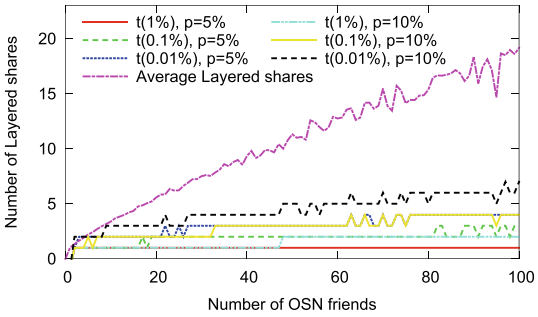

Fig. 2. Thresholds required to ensure a chosen level of protection against attackers at distance two from the dealer.

threshold required for the highest level of protection (99.99 %) against adversaries. We can also confirm the potential for having flexible privacy protection for shared attributes: an OSN user with 40 friends can have 5 different levels of protection, while the average Facebook user who has 100 friends may to have around 10 different levels of shared data protection.

We also compare the implementation complexity of Layered shares and the naive approach, i.e. when the same number of secrets is shared with a single secret sharing scheme. The comparison is based on the amount of time required to perform the generating and decoding of shares (Layered or single). We use the open source software: *ssss*[2] version 5, for secret sharing utilizing Shamir's scheme, *vdmfec*[3] for the ramp scheme and *openssl*[4] for symmetric encryption. We note that *vdmfec* implements a Reed-Solomon based weak ramp scheme, however we consider this sufficiently close in encoding/decoding complexity to a strong ramp scheme (considered in the rest of this work) for a rough estimate.

We consider an example use of the Layered scheme in Facebook, where an average user has around $n = 100$ friends. We assume that a user is sharing $k = 10$ attributes in a selected group (circle), with secret sharing thresholds ranging between 5 and 50 (with increments of 5).

Our results show that there is a very low overhead in the time taken to generate Layered shares (compared to generating single shares for the same secrets and thresholds) – up to 5 %.

6 Conclusion

We have proposed a new Layered multiple secret sharing scheme that fully preserves privacy of the data shared by the dealer, by protecting the number of secrets he is sharing and their thresholds. We consider the use of this scheme for

[2] http://point-at-infinity.org/ssss/
[3] http://freecode.com/projects/vdmfec
[4] http://www.openssl.org/

profile sharing in OSNs, that would enable direct access to selected attributes of the profile by friends, with access levels determined by the number of friends in common with the profile owner. We evaluate the security of the proposed scheme and analyse the level of threat posed by OSN users who are on the social graph of the user sharing his profile. Finally, we evaluate the complexity of critical components of the scheme by experimental evaluation. Future work includes extensions to handle revocation of shares (un-friending) and changes to the relationship status, without the need to re-distribute new shares.

References

1. Anderson, J., Diaz, C., Bonneau, J., Stajano, F.: Privacy-enabling social networking over untrusted networks. In: WOSN '09, pp. 1–6. ACM (2009)
2. Baden, R., Bender, A., Spring, N., Bhattacharjee, B., Starin, D.: Persona: an online social network with user-defined privacy. SIGCOMM **39**(4), 135–146 (2009)
3. Beimel, A.: Secret-sharing schemes: a survey. In: Chee, Y.M., Guo, Z., Ling, S., Shao, F., Tang, Y., Wang, H., Xing, C. (eds.) IWCC 2011. LNCS, vol. 6639, pp. 11–46. Springer, Heidelberg (2011)
4. Blundo, C., Santis, A.D., Vaccaro, U.: Efficient sharing of many secrets. In: Enjalbert, P., Wagner, K.W., Finkel, A. (eds.) STACS 1993. LNCS, vol. 665. Springer, Heidelberg (1993)
5. Buchegger, S., Schiöberg, D., Vu, L.-H., Datta, A.: Peerson: P2p social networking: early experiences and insights. In: SNS '09, pp. 46–52. ACM (2009)
6. Cazabet, R., Leguistin, M., Amblard, F.: Automated community detection on social networks: useful? efficient? asking the users. In: ACM WI&C (2012)
7. Chan, C.-W., Chang, C.-C.: A scheme for threshold multi-secret sharing. Appl. Math. Comput. **166**(1), 1–14 (2005)
8. Guha, S., Tang, K., Francis, P.: Noyb: privacy in online social networks. In: WOSN '08, pp. 49–54. ACM (2008)
9. Jahid, S., Mittal, P., Borisov, N.: Easier: encryption-based access control in social networks with efficient revocation. In: ASIACCS '11, pp. 411–415. ACM, New York (2011)
10. Liu, Y., Gummadi, K.P., Krishnamurthy, B., Mislove, A.: Analyzing Facebook privacy settings: user expectations vs. reality. In: 2011 ACM SIGCOMM Conference on Internet Measurement Conference, ACM IMC 2011 (2011)
11. Narayanan, A., Shmatikov, V.: Myths and fallacies of "personally identifiable information". Commun. ACM **53**(6), 24–26 (2010)
12. Pang, L.-J., Wang, Y.-M.: A new (t, n) multi-secret sharing scheme based on shamir's secret sharing. Appl. Math. Comput. **167**, 840–848 (2005)
13. Rabin, M.O.: Efficient dispersal of information for security, load balancing, and fault tolerance. J. ACM **36**(2), 335–348 (1989)
14. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
15. Ugander, J., Karrer, B., Backstrom, L., Marlow, C.: The anatomy of the Facebook social graph. CoRR, abs/1111.4503 (2011)
16. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in Facebook. In: WOSN (2009)
17. Wang, F., Gu, L., Zheng, S., Yang, Y., Hu, Z.: A novel verifiable dynamic multi-policy secret sharing scheme. In: ICACT'10, pp. 1474–1479. IEEE Press (2010)

18. Waseda, A., Soshi, M.: Consideration for multi-threshold multi-secret sharing schemes. In: ISITA 2012, pp. 265–269, Oct 2012
19. Yamamoto, H.: Secret sharing system using (k, l, n) threshold scheme. Electron. Commun. Jpn. (Part I: Commun.) **69**(9), 46–54 (1986)
20. Yang, C.-C., Chang, T.-Y., Hwang, M.-S.: A (t, n) multi-secret sharing scheme. Appl. Math. Comput. **151**(2), 483–490 (2004)
21. Yildiz, H., Kruegel, C.: Detecting social cliques for automated privacy control in online social networks. In: IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 353–359 (2012)