Collaborative Joint Content Sharing for Online Social Networks

Filipe Beato and Roel Peeters
KU Leuven ESAT/COSIC & iMinds, Belgium
Email: firstname.lastname@esat.kuleuven.be

Abstract—Online social networks’ (OSNs) epic popularity has accustomed users to the ease of sharing information. At the same time, OSNs have been a focus of privacy concerns with respect to the information shared. Therefore, it is important that users have some assurance when sharing on OSNs: popular OSNs provide users with mechanisms, to protect shared information access rights. However, these mechanisms do not allow collaboration when defining access rights for joint content related to more than one user (e.g., party pictures in which different users are being tagged). In fact, the access rights list for such content is represented by the union of the access list defined by each related user, which could result in unwanted leakage. We propose a collaborative access control scheme, based on secret sharing, in which sharing of content on OSNs is decided collaboratively by a number of related users. We demonstrate that such mechanism is feasible and benefits users’ privacy.

I. INTRODUCTION

In the past years, Online Social Networks (OSNs), such as Facebook or Google+, have played an important role in changing society, offering people the ability to share content, and build communities around shared interests. At the same time, alongside with the widespread popularity of OSNs, privacy concerns [1], [15] have prompted interest from both research community [2], [9], [13], [16], [21] and media communities [8], [20], [25]. These concerns have been mainly related to access control issues, for instance, who can access the information?

Popular OSNs empower users with some customizable “privacy settings” to take on access rights decisions based on access lists. Limiting access rights to a subset of the users, for instance, Friends, friends-of-Friends or everyone. However, those mechanisms are often difficult and may lead to accidental leakage [7], [25]. According to Gurses and Berendt [20] the access control design in current OSNs represent the principal bottleneck of privacy when sharing information. At the same time, the research community has proposed solutions to fine-tune the access control mechanism that rely on the use of encryption [3], [4], [5], [23] to prevent leakages. However, those solutions aim to protect users information imposing access control by confidentiality, but they do not allow collaboration.

Erickson and Kellogg [14] showed that discussion and collaboration while sharing information represents an important factor and contributor for social privacy. A less studied problem is that of collaboration when sharing joint content. A vast number of shared information in OSNs is related to more than one user (e.g., user tagging). Therefore, it is hard to control who is able to access the information. The approach taken by most OSNs nowadays is to give access rights to the union of all related users, lacking collaboration, and, thus leading to possible unwanted leakages. Take for example Facebook’s Data Use Policy1, which states that “If you tag someone, that person and their friends can see your story no matter what audience you selected. The same is true when you approve a tag someone else adds to your story.”

Motivated by the privacy issues, such as the lack of control of the information flow on the OSNs, and the social design in [14], we propose a collaborative sharing scheme for joint content in OSNs. Specifically, we present a scheme in which content-related users can collaboratively decide to disseminate the related joint information shared in OSNs. Figure 1 gives an example in which a picture related to five different users is shared collaboratively. Our collaborative sharing scheme is based on some social assumptions and makes use of secret sharing [28]. We show that scheme is secure. The efficiency of our scheme is demonstrator with concrete cryptographic algorithms and we demonstrate that our scheme can be implemented on top of existing OSNs.

The remainder of this paper is structured as follows: Section II overviews some preliminaries definitions. Section III describes our system and adversarial models. Section IV details our collaborative access control scheme, while Section V evaluates its security properties. Later, Section VI discusses the practical feasibility of our approach. Finally, Section VII surveys related work and Section VIII concludes the paper.

II. PRELIMINARIES

This section gives an overview of the Online Social Network definitions, notation, and cryptographic building blocks used throughout this paper.

A. Online Social Networks

Following the OSN definition by Boyd [9], a OSN allow individuals to: (1) construct a public or semi-public profile; (2) articulate a list of users with whom they share a connection; and (3) cross those connections and connections created by others to share and obtain information.

Without loss of generality, we consider that users on OSNs adhere to one or more of the following roles:

1) Publisher: user that publishes information α.
2) Viewer: user that accesses and views the posted information α.
3) Tagged: user that is tagged or α-related, i.e., also represented in the published content α.

1https://www.facebook.com/about/privacy/your-info-on-fb
In order to protect access rights of published content, popular OSNs offer a mechanism that resolves to an access control list defined by the publisher. The publisher can specify a set of viewers for a certain published content α to be set to his only Friends, or friends-of-Friends. However, when the content contains tagged users then the viewers set is represented by the union of the access rights of all the associated users, i.e., publisher and all tagged users. For example, user \( u_1 \) publishes α in the OSN and tags users \( u_2 \) and \( u_3 \). Thereby, all users in the set \( \{ F_{u_1} \cup F_{u_2} \cup F_{u_3} \} \) can view the content \( \alpha \). We define the set \( S = \{ u_2, u_3 \} \) as the α-related users. The general notation is summarized in Table I.

### TABLE I: Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u )</td>
<td>user identifier</td>
</tr>
<tr>
<td>( \alpha, \beta )</td>
<td>Information content, in plain and ciphered, exchanged among users (e.g., photo, comment, post)</td>
</tr>
<tr>
<td>( F_u )</td>
<td>Set of connections (e.g., “Friends”, “friends-of-Friends”, “Circles”) of the user ( u )</td>
</tr>
<tr>
<td>( S )</td>
<td>Set of collaborators, i.e., “tagged” or α-related users.</td>
</tr>
<tr>
<td>( (pk_u, sk_u), k_u )</td>
<td>Asymmetric key pair (public and private keys), and symmetric key from user ( u )</td>
</tr>
<tr>
<td>( E_k(\cdot), D_k(\cdot) )</td>
<td>Symmetric authenticated encryption and decryption under key ( k )</td>
</tr>
<tr>
<td>( Enc_{pk_u}(\cdot), Dec_{sk_u}(\cdot) )</td>
<td>Asymmetric encryption and decryption under the ( pk_u ), and ( sk_u )</td>
</tr>
<tr>
<td>( (t, n), f(\cdot) )</td>
<td>Threshold ( t ) out of ( n ) shares, and the Polynomial ( f(\cdot) ) of degree ( t-1 )</td>
</tr>
<tr>
<td>( s_0, \ldots, s_n )</td>
<td>Shares and the encryption of the shares of the ( n ) α-related users</td>
</tr>
</tbody>
</table>

### B. Cryptographic Background

We review the main cryptographic building blocks that are used for our collaborative access control.

**Authenticated Encryption:** A symmetric encryption method that guarantees, simultaneously, data confidentiality (i.e., secrecy) and integrity (i.e., tamper resistance). To realize an authenticated encryption scheme, is sufficient to use either a block cipher under a mode of operation, e.g., AES-CCM [31], or a dedicated design, e.g., AEGIS [32]. In general, an authenticated encryption scheme works as follows. The encryption \( E_k(m) \) of a message \( m \) under the secret \( k \) is the tuple \( \langle C, T \rangle \), with \( T \) the authenticity tag. To decrypt a ciphertext \( D_k(C, T) \), the message is first authenticated by computing \( T' \) and verifying that \( T = T' \), before it is returned.

**Secret Sharing:** Protecting a symmetric secret \( k \) by sharing it in the format of shares \( s_i \) among a number \( n \) entities, in such a way that any subset of size greater or equal than a threshold \( t \) can reconstruct the secret \( k \), while no information about the secret can be learnt from less than \( t \) shares. The scheme proposed by Shamir [28] is based on polynomial interpolation. The dealer generates a random secret \( k \) along with a polynomial \( f(x) \) of degree \( t-1 \), such that \( f(0) = k \). The secret \( k \) is then shared as evaluations of a polynomial \( f(x) \), represented as shares \( \{ s_i \}_{i=0, \ldots, n} \). Further, any player holding \( t \) or more shares is able to reconstruct the polynomial \( f(x) \) of degree \( t-1 \), and subsequently the secret \( k \). This is done by constructing the Lagrange multipliers \( a_i \) in \( t \) points of \( f(x) \), as follows:

\[
k = \sum a_i s_i \quad \text{for} \quad a_i = \prod_{j \neq i} \frac{j}{j - i}
\]

### III. Model

In this section we present our ecosystem, the associated threat model, and the access rights goals.
A. System Model

We consider any OSN that follows the definition of Boyd [9]. Furthermore, we assume that each user \( u \) holds a public-private key pair \((pk_u, sk_u)\), where \( pk_u \) is known to everyone on the OSN (e.g., available on the user profile page), whereas the \( sk_u \) is only known to \( u \). In addition, each \( u \) holds a collaborative sharing key \( k_u \) that is shared only with the user defined set \( F_u \). Note that, \( u \) can form different \( F_u \), and that \( F_u \) can be represented by the full group of Friends, friends-of-Friends, or any subset of those.

B. Adversarial Model

We assume that the adversary is capable to access the encrypted version of the joint content \( \beta \), and can interact with the collaborative users (i.e., \( \alpha \)-related users) to retrieve shares. The publisher is assumed to be honest as he already holds the information. We do not consider denial of service attacks, since we assume that the main motivation of an adversary is to learn, and, thus, will not win on removing information. However, once the content is distributed, there is no way to prevent a malicious viewer from storing or re-distributing the content. In this case, such user is said to break the social contract established along with the friendship relation.

C. Goals

For a model that provides a collaborative sharing where information disseminated is private to prying eyes, as illustrated in Fig. 1, we need to attain the following goals:

– Collaborative Access Rights: Access to joint content \( \alpha \) is only granted to viewers who are connected to at least a threshold \( t \leq n \) of \( n \) \( \alpha \)-related collaborating users (i.e., \( \alpha \)-related users choosing to collaborate in the sharing of content \( \alpha \)).

– Content Confidentiality: Joint content \( \alpha \) should be published in encrypted format \( \beta \), such that adversaries are not able to infer any information nor access to the content \( \alpha \).

IV. COLLABORATIVE SHARING SCHEME

We propose a collaborative sharing scheme using secret sharing, where access rights are only passed to other profiles if, and only if, a threshold of \( t \leq n \) users out of all \( n \) \( \alpha \)-related users collaborate (i.e., from the set \( S \)).

We categorize our scheme to be composed by a setup algorithm and three protocols, as follows:

– Setup(\( \lambda \)) : A randomized algorithm that generates the user \( u \) initial parameters \( I \leftarrow \{pk_u, sk_u\}, k_u \) from a security parameter \( \lambda \).

– Publish(\( \alpha \)) : A randomized protocol that generates a fresh secret key \( k \), the encryption \( \beta \) of \( \alpha \), the polynomial \( f(\cdot) \) of degree \( t-1 \), and the shares for the \( n \) \( \alpha \)-related users (s.t., \( t \leq n \)).

– Collaborate(\( \gamma_i, \beta \)) : Used by each of the \( n \) \( \alpha \)-related users from the set \( S \) to distribute their shares.

– Retrieve(\( \beta, \delta_0, \ldots, \delta_t \)) : Extracts the content \( \alpha \) from the ciphertext \( \beta \) using \( t \) shares to reconstruct the secret \( k \). Decrypt can also return \( \perp \) if \( \beta \) is malformed.

We will now discuss these algorithms and protocols separately. Figure 2 provides a detailed description of the three protocols.

A. Setup

Each user \( u \) runs the setup algorithm once to generate a public/private key pair \((pk_u, sk_u)\) alongside with the random collaborative sharing key \( k_u \). The public key \( pk_u \) and collaborative sharing secret \( k_u \) are made available, for instance, on his/her profile. While \( pk_u \) is publicly accessible, \( k_u \) is restricted to the set \( F_u \). For new users who are added to \( F_u \) (e.g., during the acceptance of a friendship request), the procedure of making these two keys available has to be repeated. Whenever \( u \) composes an additional group \( F_u' \), a different \( k_u' \) is generated and made available to the respective \( F_u' \).

B. Publishing content

A user \( u \) wants to publish some joint content \( \alpha \) on the OSN, such that, \( \alpha \) is related to a number of other users. For instance, apart from \( u \) the set \( S = \{u_0, u_1, u_2\} \) is also being presented in \( \alpha \), e.g., “tagged”, and therefore they undertake a collaborative sharing approach. To publish \( \alpha \), \( u \) generates a random key \( k \) used for the encryption of \( \beta \leftarrow E_k(\alpha) \). Then, \( u \) constructs a polynomial \( f(x) \) of degree \( t-1 \), such that, \( f(0) = k \) and \( t \) represents the minimum number of users required to collaborate. After, \( u \) generates \( n \) different shares, one per \( \alpha \)-related user, encrypting them along with \( k \) using the \( \alpha \)-related user’s public keys \( \gamma_i \leftarrow pk_{ui} \). The reason why \( k \) is shared along with the shares, is that \( \alpha \)-related users should be able to make an informed decision about collaborating in the content sharing (they should be able to review \( \alpha \)). Finally, \( u \) publishes the encrypted content \( \beta \) alongside with the set of encrypted shares \( G \) and \( S \). Figure 2a illustrates this protocol.

C. Collaborate with shares

The process of collaboration resumes to the distribution of shares. \( u_i \) as a \( \alpha \)-related user is empowered to contribute with the distribution of his personal share \( s_i \) after having reviewed \( \alpha \). Subsequently, \( u_i \) publishes \( \delta_i \), the encryption of his share using his/her collaborative sharing key, for instance, in the form of a comment. This allows other users in \( F_u \) to retrieve this encrypted share needed to retrieve the content in a later stage. Figure 2b gives an overview of the protocol.

D. Retrieving content

In order to retrieve \( \alpha \), any user \( u_j \) accessing \( \beta \) is required to obtain at least \( t \) shares, to then be able to reconstruct the secret \( k \). However, to get hold of \( t \) shares, \( u_j \) needs to have access to the collaborative sharing key from \( t \) collaborating users. This means that, in order to view \( \alpha \), one needs to be a member of at least \( t \) sets \( F_{ui} \) of collaborating users \( u_i \) to have access to the corresponding collaborative sharing keys \( k_{ui} \). Recall that these keys were made available to the users in \( F_{ui} \) during setup. The authenticity of the decrypted shares is guaranteed by the used authenticated encryption scheme. The protocol to retrieve the content is depicted in Fig. 2c.
E. Selection of the Threshold

In our scheme the set of authorized viewers is constructed implicitly, the threshold \( t \) represents an important parameter. The threshold value holds how private is the content shared. Hence, deciding the threshold is a matter of design that needs to be decided either by the OSN (top down) as part of their service, or based on the trust necessary and the sensitivity of the information (a subjective value to be decided in a given application domain). Higher trust or lower sensitivity is likely to imply a low threshold, whereas lower trust or greater sensitivity would imply a high threshold. Different trust algorithms or interfaces to solve this problem are described in [17], [24], and is a topic for future research.

However, it should be noted that a too high threshold could lead to a deviation from the purpose of sharing, as the content will only be accessible by a very limited amount of viewers. For example, the threshold being 2 or 3 will probably suffice to stop one’s coworkers of viewing pictures of a party with old friends, while at the same time not denying access to other old friends that might not be friends with all the tagged users.

VI. Security Evaluation

To achieve content confidentiality, adversaries that do not know user collaboration keys \( k_u \) should not be able to infer any information on the joint content \( \alpha \). Authenticated encryption schemes, used to encrypt the joint content or shares, provide security against chosen ciphertext attacks (IND-CCA). However, for the adversary not to learn the joint content, the public key encryption scheme also needs to be semantically secure (IND-CPA), since the encrypted message contains the secret \( k \) that can be used to decrypt the encrypted joint content \( \beta \). The public key encryption scheme does not need to be secure against chosen ciphertext attacks, because the encrypted message \( \gamma_i \) contains the secret \( k \). A user \( u \) that wants to collaborate will always verify the joint content, authenticated decrypted using this key, before publishing the share to his/her connections \( \mathcal{F}_u \). As a consequence, attackers that want to use this user as a decryption oracle for a share \( s_i \), need access to this user’s collaboration key \( k_u \) and provide authenticated encrypted content that verifies when using the secret \( k \). The second condition implies that the attacker already knows \( k \) and has access to the joint content \( \alpha \) or forwarded the original \( \beta \) and \( \gamma_i \) to the user. As such substitution attacks (by another user of the OSN that acts a publisher or the OSN provider itself), where the attacker substitutes \( \beta \) or any of the \( \gamma_i \), are ruled out.

For the used \((t, n)\)-secret sharing scheme, at least the threshold number \( t \) of shares are needed to reconstruct the shared secret \( k \). Shamir’s secret sharing scheme is information theoretic secure, meaning that even adversaries with unlimited computational power cannot infer any information about \( k \) when knowing less than \( t \) shares. This means that until \( t \) tagged users collaborate, the joint content remains fully confidential (also to the OSN). At the moment that \( t \) or more tagged users collaborate, the ability to view the joint content \( \alpha \) boils down to being able to collect \( t \) shares, which means having access to \( t \) collaboration keys \( k_u \) of tagged users who collaborated in the sharing of the joint content. Given that none of the users breaks the social contract, collaborative sharing is achieved: only users related to enough collaborating tagged users can see the content \( \alpha \). At this point, if one also wants to hide the joint content from the OSN provider, the collaboration keys \( k_u \) cannot be known to the OSN provider. This means that one has to either share these keys outside the OSN or publish these keys, encrypted under the public keys of all users in \( \mathcal{F}_u \).

A. Practical Design

Our scheme implementation can be fulfilled by a browser extension, e.g., Firefox or Chrome extensions. The development of browser extensions for popular browsers is done in Javascript, which represents a bottleneck for performance of cryptographic operations. To address this issue, we divide
our design into a client side responsible for the user human interactions, and a server side responsible for cryptographic request. The communication between both is performed locally in the same machine through a socket communication. Such design enables efficiency as it allow the cryptographic library to be implemented efficiently [27]. In addition, it eases the process of possible migrations to different browser platforms.

Public keys are made publicly available by each user on their OSN profile, while the collaborative secret $k$ is shared only with $F_u$. The browser extension can automatically retrieve the tuple $(pk_u, k_u)$ from the OSN, and store it locally. Later, to engage in the scheme, the user publishes the encryption of the content $\beta$ along with the encrypted shares and the list of tagged users $S$. Later, the $\alpha$-related users in $S$ collaborate by disseminating their shares as comments. In order to be aligned with the general Terms of Services of popular OSNs, where encryption is not allowed, $\beta$ can be published in an external storage (e.g., Dropbox), and the respective link in the OSN. The browser extension automatically translates the encrypted content, retrieves the $t$-collaborative shares, decrypts the respective shares, and subsequently displays the content.

B. Performance

For a security parameter $\lambda = 128$ bit, the size of the secret key in authenticated encryption scheme will be 128 bit. This means that the size of shares, using Shamir’s secret sharing scheme, will also be 128 bit. For the public key encryption scheme we make use of elliptic curve cryptography (ECC) with the curve25519 [6], which is among the fastest curves for $\lambda$ of 128. According to the benchmark tool Supercop\(^2\), a single core desktop machine requires for an elliptic curve point multiplication on this curve about 60 ms, while a smartphone processor needs 200 ms. An additional benefit of this specific curve is that each 256 bit number is a valid point on the curve. This makes that we can encrypt a 256 bit message $M = k|s_i$ as follows: $R = rP, M \oplus rY$, with $r$ a random number, $P$ the generator of the curve and $Y$ the public key of the intended recipient of the message. This encryption scheme is semantically secure (IND-CPA) under the computational Difﬁe-Helmann (CDH) assumption.

For the publisher, the computational effort and communication overhead can be reduced by almost 50% by choosing one random number for all public key encryptions at the time of publishing content $\alpha$. As such we only need $n + 1$ EC point multiplications (transfer $n + 1$ EC points) instead of $2n$ EC point multiplications (transfer $2n$ EC points). For efficiently evaluating the polynomial one can make use of Horner’s rule, resulting in $t - 1$ multiplications and $t - 1$ additions for a polynomial of degree $t - 1$. The tagged user willing to collaborate, first decrypts, using his private key, his share together with the symmetric encryption key that was used to encrypt the content. After approval of the authenticated decrypted content, the user then authenticated encrypts his share under the collaborative secret. To view content, one needs to collect $t$ shares (for which one has to do $t$ authenticated decryptions), then combine these shares using Lagrange interpolation and finally use the resulting key to decrypt the content. In case the collaboration keys are not cached by the browser extension, one needs an extra transfer of $t \times 16$ bytes. An overview of the computational effort (ordered according to efficiency) and communication overhead (in comparison with posting the content $\alpha$ directly on the OSN) is depicted in Table II.

| TABLE II: Computational effort and communication overhead |
|---------------------------------|----------|
| **Computational effort** | **Publish** | **Collaborate** | **Retrieve** |
| EC multiplication | $n + 1$ | $t + 1$ | $t + 1$ |
| Auth. en-/decryption | $t$ | $t$ | $t$ |
| Scalar multiplication | $n(t - 1)$ | $t$ | $t$ |
| Scalar addition | $n(t - 1)$ | $t$ | $t$ |
| Bitwise exclusive OR | $n(t - 1)$ | $t$ | $t$ |
| Comm. overhead [byte] | $(n + 1) + 32$ | $48$ | $t + 16$ |

Note that elliptic curve multiplications require an order of magnitude more computational effort than authenticated encryption schemes. If the user’s device supports native AES instructions, AES-based authenticated en-/decryption (e.g., AES-CCM) might even be more efficient. The computational cost for scalar arithmetic and bitwise exclusive ORs is negligible. Therefore, we say that the computational effort for the publisher rises with the number of tagged users $n$. The computational effort for the viewer is minimal (no costly public key operations) and independent of $n$, it solely depends on the selected threshold $t$.

VII. Related Work

Several solutions have been proposed focusing mainly on the problem of privacy as confidentiality, and who can access the content [3], [4], [5], [18], [22], [23], [29]. Those solutions, however, model the OSN to be adversarial and are built as add ons, enforcing access control to published content by means of encryption. Whilst NOYB [18] protects profile information by using fake information, others solutions [3], [4], [5], [22], [23], [29] apply different cryptographic mechanisms to enforce confidentiality and access control to the published content. For instance, Scramble! [5] utilizes the OpenPGP standard for encryption, and Persona [3] implements attribute base encryption with fine grained policies. Further, Günther et al. [19] proposed another building block for privacy in social interactions by a cryptographic solution for private profile management that achieves confidentiality and unlikability. However, while it is possible for users to define access control lists per published content, such solutions empower the publisher with complete control not considering other $\alpha$-related users. Recently, Zhu et al. [33] suggested a new collaborative framework for a private OSN. Their system relies on the group-oriented convergence cryptosystem and focus on communities that are jointly build. Although the initialization of the community is done in collaboration by the members, the content access and delegation is not jointly decided. Other solutions, such as Diaspora [12] and Safebook [11] take a more drastic approach proposing new decentralized architectures to replace existing OSNs. Whereas, HummingBird [10] suggests a centralized twitter alike system that requires authorization for following tweets. Unfortunately, such solutions rely not only on the availability of peers but also on the assumption that a large mass of users move to a new system, in detriment to existing ones where all their friends are.

The concept of secret sharing has been previously explored on the context of distributed networks, such as mobile ad hoc

\(^2\)http://bench.cr.yp.to/supercop.html
networks [26] for trust distribution and to prevent access from unauthorized nodes. Later, secret sharing has been similarly applied to decentralized OSNs. Vu et al. [30] introduced the use of secret sharing on distributed OSNs to protect data and secure backup against untrusted peers. In this way, the issue of data backup storage within nodes is mitigated. Further, Nojoumian et al. [24] propose a secret sharing scheme where the distribution of shares is based on the reputation and social interactions of the players in OSNs. While this solution does not force collaboration on content related to more than one user, it can be used as an extension to define the allocation of shares.

VIII. CONCLUSION

This paper presents a solution for a collaborative sharing scheme for OSNs based in secret sharing, where each content related user can collaborate by disseminating their share within the OSN. Only users that obtain the threshold $t$ of shares can view the content. and, thus, the set of authorized viewers is constructed implicitly. At the same time, it guarantees the shared content to be protected from any curious viewers without knowledge of at least $t$ shares. Finally, our approach makes collaboration mandatory and in the process provides related users with information about co-related users privacy preferences. We have evaluated and presented concrete examples that demonstrate the practical feasibility of our approach. We also identified some important open challenges that call for further research. Items for future work include further analyzing the choice of threshold and the number of shares per related user. As well as, increasing the collaboration rounds efficiently to better coordinate how far in the OSN topology information can be disseminated.

ACKNOWLEDGMENTS

We gratefully acknowledge Seda Gurses for the initial discussion that helped bootstrap the research of this paper. We would like to thank the anonymous reviewers for their valuable input. This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007) and by the IWT SBO SPION project. Filipe Beato is funded by the FCT fellowship SFRH/BD/70311/2010.

REFERENCES