POSTER: PRETTY GOOD FACEBOOK PRIVACY

Alexandra Dirksen*, Sebastian Gajek*, Martin Johns†, Robert Michael*
* *University of Applied Sciences Flensburg, Germany*
† *SAP SE, Germany*

1. Introduction

Online Social Networks (OSN) have a significant impact on the daily life of a digital society. According to estimates, the number of worldwide social network users reached 1.96 billion and is expected to grow to some 2.5 billion by 2018 [1]. Many online social network users are unaware or ignore the fact that they disclose various forms of private information. The concentrated availability of user-specific data raises privacy concerns. It can be mined to infer social, economic or psychological profiles of the user without her witness. Users are left with the decision to either tolerate the disclosure of privacy or to boycott social networks at all. The latter is particularly problematic given the widespread proliferation of social networks.

2. Our Contribution

The phenomenon of OSNs attracted many researchers to study concepts of improving the privacy situation within those networks. There have been many approaches with different impact on user privacy (e.g. [2], [3], [4]). The motivation of our work is the protection of the *content*, such as postings, images, videos, or documents, users publish on the network. To this end, we design a content privacy layer and implement a prototype for Android mobile devices on top of the Facebook social network, dubbed *pretty good facebook privacy (PGfbP)*. The design of PGfbP puts much emphasis on the following features:

- Simplified key distribution and management without the involvement of a third party.
- Group communication security with symmetric-key predicate encryption supporting fine-grained access control (e.g. expressed in terms of Boolean formulas).
- Protection of the encryption and decryption interfaces against exfiltrating Web attacks.
- Preservation of the Facebook Web experience through the integration of all OSN features and usual look and feel.
- Independence of Facebook's official SDK [5] or any API which might limit the functionality of PGfbP in future.

3. Particular Problems

We solve several interesting challenges both at the cryptography and Web application layer to achieve our privacy goal of leaking no information about the content published by users in presence of a honest-but-curious online social network. Our work showcases that a cryptographically secure solution (with a security proof) alone is insufficient when being implemented in a concrete environment. Specifically in a Webview [6] environment on Android mobile phones, one has to take care of the isolation of encryption/decryption events.

3.1. Cryptographical Layer

A key tool in our solution is a *symmetric-key predicate encryption (SPE)* system [7]. In a SPE every user distributes a decryption key with user-specific permissions. Content posted to the network is encrypted together with a ciphertext policy defined by the encrypting user. In case of Facebook the friend list mechanism (e.g. friend, family or acquaintance) acts as the policy. A user in possession of a token is only able to decrypt ciphertexts which comply with the policy. This approach has three major benefits:

- When using a standard symmetric encryption scheme the ciphertext length amounts to $O(|m| \cdot |p|)$ where $|m|$ is the content length and $|p|$ the policy length (linear in the number of recipients). Our SPE considerably shrinks the communication overhead and produces ciphertexts of length $O(|m| + |p|)$.
- Our SPE system builds upon the key and data encapsulation paradigm: a predicate-based key encapsulation mechanism encrypts an ephemeral key under a predicate structure implementing the policy, while a standard symmetric-key encryption scheme (e.g. AES-GCM) encrypts the content under the ephemeral key. This way, content of arbitrary length (e.g., videos) can be encrypted efficiently.
- Our scheme satisfies not only content hiding, but also the stronger notion of policy privacy. Users with a valid decryption token learn the decrypted content, however they do not learn the set of users entitled to decrypt the content.

3.2. Web Application Layer

We implemented our privacy layer as an Android application running on top of Webviews. They provide a simple interface to Web objects and may be seen as a tiny Web browser within the mobile app. In PGfbP we use Webviews to interface the Facebook Web site in order to extract and insert the privacy-sensitive content. Unfortunately, the naive deployment of the Webview concept makes our mobile app susceptible to attacks through server-sided injection of JavaScript (and related Web techniques). For example, an attacker can intercept the encrypted text after it is placed in the Webview. It is also possible to intercept the plaint text from the Webview, before is is encrypted. The reason is the same origin policy. It enforces a weak isolation of Web objects. In a nutshell, the same origin policy says Web objects have reckless access to objects in the same domain. Consequently the OSN has uncompromising access to all Web objects as the sole domain provider. To prevent the OSN from bypassing our privacy measures we implemented the following isolation techniques:

- When the app detects some encrypted content, it first extracts the ciphertext from the HTML. Next, it decrypts the ciphertext within the native app and re-embedds the content plaintext in an *iFrame*. The latter isolates the content plaintext and prevents exfiltration through malicious scripting code.
- When it comes to encryption of user input our application intercepts the event and relays the input to a native Android textbox. It is then encrypted before being forwarded to the Webview.

References

[1] "Statista: Statistics and facts about social media usage," (Date last accessed 01-March-2017). [Online]. Available: https://www.statista.com/topics/1164/social-networks/

[2] S. Jahid, P. Mittal, and N. Borisov, "Easier: Encryption-based access control in social networks with efficient revocation," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, 2011, pp. 411–415.

[3] "Snapencryption: A new plugin to protect private snapchat pictures, developed at tu darmstadt," (Date last accessed 01-March-2017). [Online]. Available: https://www.trust.informatik.tu-darmstadt.de/news-events/news/einzelansicht/artikel/snapencryption-a-new-plugin-to-protect-private-snapchat-pictures-developed-at-tu-darmstadt/

[4] W. He, D. Akhawe, S. Jain, E. Shi, and D. Song, "Shadowcrypt: Encrypted web applications for everyone," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 1028–1039.

[5] "Facebook android sdk," (Date last accessed 01-March-2017). [Online]. Available: https://developers.facebook.com/docs/android/

[6] "Android developer: Android framework," (Date last accessed 01-March-2017). [Online]. Available: https://developer.android.com/index.html

[7] E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in *Theory of Cryptography Conference*. Springer, 2009, pp. 457–473.

[8] A. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2012, pp. 318–335.

[9] D. F. Aranha and C. P. L. Gouvêa, "Relic is an efficient library for cryptography," (Date last accessed 01-March-2017). [Online]. Available: https://github.com/relic-toolkit/relic