NOYB: Privacy in Online Social Networks

Saikat Guha, Kevin Tang, and Paul Francis
Cornell University, Ithaca
{saikat, francis}@cs.cornell.edu, kt258@cornell.edu

ABSTRACT

Increasingly, Internet users trade privacy for service. Facebook, Google, and others mine personal information to target advertising. This paper presents a preliminary and partial answer to the general question "Can users retain their privacy while still benefiting from these web services?". We propose NOYB, a novel approach that provides privacy while preserving some of the functionality provided by online services. We apply our approach to the Facebook online social networking website. Through a proof-of-concept implementation we demonstrate that NOYB is practical and incrementally deployable, requires no changes to or cooperation from an existing online service, and indeed can be non-trivial for the online service to detect.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer Communication Networks—*Distributed Systems*; E.3 [**Data**]: Data Encryption

General Terms

Design, Security

Keywords

NOYB, Privacy, Cloud Computing

1. INTRODUCTION

The Internet was originally designed to connect endhosts. All data was stored and processed in these endhosts, and the network simply provided transit. Reasoning about privacy largely involved which users and endhosts were processing the data. But that is no longer the case. Today, the Internet *is* the computer. Data is stored and processed in the "cloud". Service providers, such as Google and Facebook, are the faceless entities that control the cloud, and the user, for better or for worse, is merely on for the ride.

WOSN'08, August 18, 2008, Seattle, Washington, USA.

Of course, there are good reasons for cloud computing. Gmail, Google's web-based email service, for instance, provides searchable, any-time any-where any-device access to the user's email through nothing more than the universal web browser. Facebook, an online social networking website, allows groups of friends to stay up-to-date with developments in each other's lives. Meanwhile, under the covers, the cloud ensures data durability, disaster recovery, platform independence, and provides integration with other services, all completely transparent to the user.

In today's economic climate, advertisers, rather than the users, pay for the cloud, and consequently, it is the interests of the advertisers that takes priority over age old principles such as that of least privilege [18]. The cloud, today, collects massive amounts of private information to provide highly targeted advertisements. Not surprisingly, lapses in security, poor judgement, or the lack of judicial oversight leaves users vulnerable: identity theft is rampant [19], information leakage is common [13], and the government is given an avenue to impinge on civil liberties by sidestepping users and going directly after cloud [2].

Recent work in programming language techniques [4] demonstrate that it is possible to build online services that guarantee conformance with strict privacy policies. However, such approaches require buy-in from the service provider who, arguably, *need* the private data to generate revenue, and therefore have the incentive to do precisely the opposite. The research question that we seek to explore therefore is to what extent a user can ensure his own privacy while benefiting from existing online services.

The user unilaterally encrypting his data preserves privacy, however, doing so precludes search, and more importantly, breaks targeted ads. In order to preserve its bottomline, a cooperative service provider may re-engineer the service to provide privacy, or push ad targeting to the client. An adversarial service provider not willing to expend this effort, on the other hand, can simply deny service to unprofitable users. To account for the latter case, a solution must protect privacy conscious users from being (easily) discovered.

NOYB, short for *none of your business*, is based on the observation that some online services, notably social networking websites, can operate on "fake" data. If the operations performed on the fake data by the online service can be mapped back onto the real data, the user can, to a degree, make use of the service. Furthermore, privacy can be preserved by restricting the ability to recover the real data from the fake data to authorized users only. This ob-

servation leads naturally to our solution: user data is first encrypted, and the ciphertext encoded to look like legitimate data. The online service can operate on the ciphered data, however, only authorized users can decode and decrypt the result.

More broadly, NOYB proposes a new way of thinking about how to achieve privacy in online services whereby the user devises a transformation under which much of the functionality of the service is preserved, but which can only be undone by authorized users. The transformation is weaker than traditional encryption in that strictly more information is revealed to an adversary, but with the benefit that the victim can fly low under the adversary's radar by making it hard for the adversary to find the victim amongst ordinary users. Such an approach can be deployed incrementally by small groups of users without buy-in from the service provider.

Overall this paper makes three contributions. First, we present a general cipher and encoding scheme that preserves certain semantic and statistical properties such that online services can process the data oblivious to the encryption. Second, we show how to apply this general approach to Facebook. And third, we report on our proof-of-concept implementation which demonstrates that NOYB is practical, feasible, and incrementally deployable by endusers without the need for additional infrastructure.

Having said that, we do not answer the question whether modeling the service provider as an adversary, and correspondingly buying into heavyweight mechanisms such as a PKI, is necessary. But the existence of our solution that preserves, to a large extent, both user privacy and service functionality under such an extreme model suggests the bar can be set high for other solutions that trade off complexity for a more cooperative service provider.

2. MOTIVATING EXAMPLE

Facebook is a popular online social network. A user's Facebook profile contains a wealth of personal information, including name, photo, date of birth, contact information, sexual orientation and relationship status, political and religious views, personal interests, hobbies, education history and more. This information is made available to members of the user's social network, allowing friends to stay in touch and up-to-date with each other's lives. At the same time, Facebook generates revenue by targeting ads to highly specific demographics (e.g. single males between 18–24 years of age in New York City).

Registering on Facebook under a pseudonym, or obfuscating one's personal information is forbidden by Facebook's terms of service [9]. Indeed, Facebook has banned users in cases where it identified violations of these terms [16]. Although, negative publicity has forced Facebook to reinstate some users [17].

Based on the study conducted by Acquisti and Gross [1], Facebook users *are* concerned about who can access their personal information. While most users (60%) trust their friends almost completely with their personal information, significantly fewer (18%) trust Facebook (the company) to the same degree, and even fewer (6%) trust strangers. Yet, under the covers, Facebook allows any application developer (a stranger) access to a user's profile [10]. While a privacy conscious user may choose to not use Facebook, [1] finds that peer pressure drives membership, and unawareness of the true visibility of profiles lulls users into revealing personal information to untrusted parties.

Facebook provides two primary services in the minds of users [1]: first, advertising information about oneself, typically restricted to one's extended social network, in order to attract dates, and second, to find classmates. Since the ability to search inherently requires revealing information to Facebook, in this paper we focus on preserving user privacy while allowing users to advertise their private information to friends.

3. GOALS AND ASSUMPTIONS

Motivated by the example in Section 2, NOYB strives to achieve the following goals.

Privacy Preserving: NOYB preserves privacy defined as contextual integrity [12]. In such a framework, pieces of a user's information are scattered but public; it is the inability of an adversary to *combine* these pieces that defines privacy. For instance, it may be public knowledge that there exists some user named Alice, and some user aged 25, but an adversary is not able to conclude with any certainty that Alice is 25. Only trusted parties, as designated by the user, typically excluding the untrusted online service, are able to combine the user's information.

Incrementally Deployable: NOYB can be deployed incrementally (by small groups of users) without any cooperation from the online service. In this deployment mode, a (hopefully large) fraction of the functionality provided by the online service is preserved. If the service cooperates, however, a greater fraction of the functionality may be preserved, and NOYB can be rolled out to all users.

Hard to Detect: In the case where an online service is hostile to NOYB, NOYB users blend into the crowd. The implicit assumption, as substantiated in Section 2, is that the service pays a high penalty for falsely accusing non-users. Making it difficult for the service to find NOYB users through automated means serves as a deterrent against punitive action as long as the fraction of users using NOYB is not significant.

4. NOYB: NONE OF YOUR BUSINESS

This section starts with an overview of NOYB, followed by a detailed description of its operation.

4.1 Basic Primitive

NOYB operates by first partitioning private information into multiple *atoms*, and then replacing each atom with its encryption. A simplistic approach would be to encrypt each atom, and share the key with other users authorized to view that atom. While such a scheme does not reveal any user information to the online service, an online service can easily find users encrypting their data by looking for the traits of the cipher-text. Steganography can help avoid detection, but it is impractical to store cipher-text inside Facebook atoms as the cover-text, for example, as the atoms tend to be small.

Instead of this simple encryption scheme, NOYB substitutes the user's atom with another user's atom picked pseudorandomly. In essence, if all the atoms of the same class compose a *dictionary*, NOYB encrypts the *index* of the user's atom in this dictionary, and uses the ciphered index to pick the replacement atom from the dictionary (Figure 1).

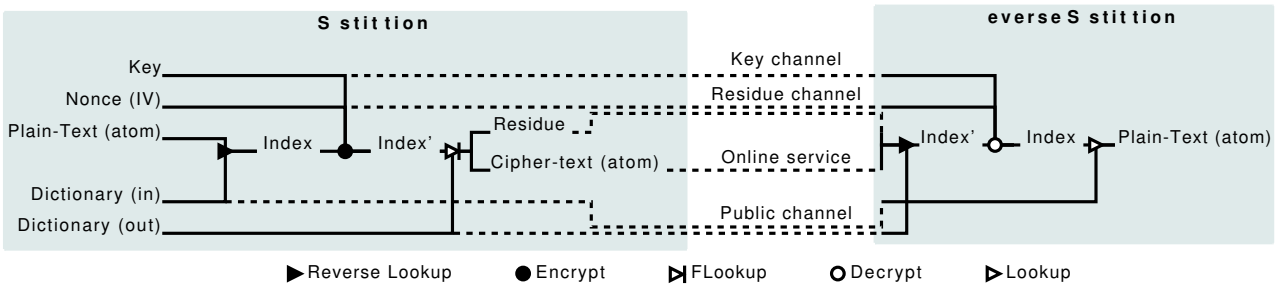

Figure 1: The basic NOYB primitive is a pseudorandom substitution cipher applied to each atom of private data. Dictionaries are maintained for each class of atoms; input and output dictionaries may be the same. The pseudorandom sequence is generated by a symmetric-key cipher. Key management is out-of-band. Nonce is unique per-update. Residue channel may be in-band, possibly covert (e.g. steganography), or out-of-band (e.g. peer-to-peer pub-sub network).

The benefit over the previous approach is that NOYB is harder to detect as the cipher atom is a legitimate atom, albeit for some random user. To illustrate, assume Alice's name, sex and age (Alice, F, 25) is partitioned into two atoms: (Alice, F) and (25). The first atom is substituted with (Bob, M) say, and the second with (28) say, from Bob and Charlie respectively based on the encrypted indices. Alice's friends can reverse the encryption to recover Alice's information. And while Alice's atoms may similarly show up in other users' profiles, an adversary cannot piece together her atoms.

4.2 Key Management

Existing key management protocols (e.g. [3]) are used to distribute keys to users authorized to view the private information; however, the much of this complexity is hidden from the user by the NOYB application. The key management protocol is implemented out-of-band (OOB) — i.e. does not exchange messages using the online service to which NOYB is being applied. This is pursuant with our goal of preventing the online service from identifying NOYB users by identifying (high entropy) key management messages. The choice of the OOB channel depends on the trust model, and may range from unencrypted emails exchanged through one or more trusted third-parties, to a peer-to-peer network secured with a PKI.

The number of keys depends on the online service. For instance in the case of Facebook, each user maintains one master key, from which it generates keys to encrypt each profile field. The master key is distributed to the user's social network (friends, friends of friends etc.). Revocation is handled by negotiating a new key, however, only updates to the profile are encrypted with the new key. By not re-encrypting the profile with the new key at the time of revocation, NOYB avoids the flurry of updates across multiple fields that may otherwise be detected by the online service. The tradeoff, however, is that multiple keys may be needed to decrypt a profile. The number of decryption keys needed is a function of how often profile fields are updated in relation to how often new keys are negotiated, and is upper bounded by the number of fields.

4.3 Dictionary

The dictionary used for substitution is a mapping from a sequential index to unique atoms and their corresponding frequency. The dictionary supports three functions: first, given an index it returns the corresponding atom (Lookup); second, given an atom it returns the corresponding index (Reverse Lookup); and third, given a (random) number in the range $[0, 1)$, it returns an atom with the largest index such that the sum of frequencies of all atoms with lower indices is less than the given number (FLookup) — in essence, preserving the marginal distribution of the cipher atoms given the uniformly distributed ciphered index. The joint distribution of atoms can be preserved by using different input and output dictionaries based on the values of other atoms (Figure 1).

Dictionaries are public, and include atoms from both NOYB users and (some) non-users. The data from non-users thwarts the service from using the dictionary to discover users, as well as improves the confusion property of the pseudorandom cipher. While we assume for simplicity that there is one global dictionary per class of atoms, private dictionaries may be used by groups of friends. For public dictionaries, external mechanisms are used to disseminate the mappings.

Finally, dictionaries must support dynamic updates to allow new atoms to be added over time. One concern in this regard is that adding new atoms must not break the existing index-to-atom mappings that may be in use. Consequently, the dictionary is append-only. Updates to atom frequencies, however, are in-place since the frequencies are used only for encryption. Updates are performed anonymously over secure channels to defend against timing and eavesdropping attacks.

4.4 Pseudorandom Substitution

The semantic security of the substitution cipher is derived from the security provided by the encryption process that lies at the core of the algorithm (Figure 1). As illustrated in the figure, a dictionary lookup is used to transform the atom to an index. The index is encrypted using the symmetric key and random nonce; we assume existing ciphers for this purpose. The output of the underlying cipher is used to select the cipher-text atom using the FLookup primitive of the output dictionary. By design, the distribution of the cipher text matches that of the plain text, which helps protect against certain frequency analysis attacks. The information discarded by the FLookup process is encoded as a *residue*, and is transmitted separately along with the nonce to assist in the reverse substitution process. Message integrity is ensured through a MAC computed over the output of the underlying cipher across all atoms, and transmitted along-

side the residues. If the underlying cipher provides semantic security, the pseudorandom substitution is also semantically secure.

4.5 Communication Channels

NOYB sends data across four channels, of which the key management channel and the public channel for exchanging dictionaries were discussed earlier. The online network itself serves as the third channel over which the cipher-text is exchanged. We now discuss the fourth channel, called the residue channel.

The residue channel is used to exchange small amounts of data for each update. As mentioned, this data typically consists of nonces, residues, and the message integrity. While the residue channel, like the key management channel, may be OOB, it is possible to use the online service itself to send the data in-band. This is accomplished by steganographically encoding the data within other cover data [11], for instance within the user profile photo in Facebook. The in-band channel allows members with the necessary keys to decrypt the private information without engaging in any additional communication. The small size of the residual data allows the greatest flexibility in the choice of steganographic technique in order to avoid detection. Correlated updates may be delayed, or updates batched in order to avoid detection.

5. ATTACKING NOYB

In this section we discuss attacks on NOYB components. While some aspects of security are ultimately rooted in components external to NOYB, such as the key management algorithm, the underlying cipher, and the steganographic scheme used, we consider only attacks on mechanisms described in this paper.

The first line of defense for NOYB users is hiding in the crowd of ordinary users. Several NOYB mechanisms are geared towards this goal. First, cipher-text is encoded as legitimate atoms lacking any identifying tags. Second, the marginal distribution of the cipher-text atoms, and to some extent the joint distribution, matches that of legitimate atoms, which protects against Bayesian detection methods. Third, the partitioning into atoms preserves semantic relationships in the private information (e.g. name and sex are kept together) to make it harder to develop heuristics to semantically detect encrypted information. Fourth, steganography is using sparingly to minimize the chances of detection. Fifth, the public dictionary is padded with information of non NOYB users to increase the rate of false positives if an adversary were to pick users matching dictionary atoms at random. And sixth, communication across different channels is decorrelated to avoid timing attacks. That said, in the event the online service does discover a NOYB user, the service does not learn the private information, and can at best deny the user service if the terms of service so allow.

An attacker may attempt to expose NOYB users by polluting the dictionary with atoms that do not normally exist, and looking for these fake atoms in the user's (encrypted) private information. To avoid falling into this trap, dictionaries are maintained by groups of users or trusted third-parties that authorize updates only from certain members. However, these third-parties are not trusted to enforce privacy.

The message integrity protects against attackers that modify the cipher-text. The use of standard ciphers protects against known plain-text attacks. In the event the keystream used to encrypt an atom is revealed, the random nonce ensures the keystream cannot be used to decrypt other atoms thereby limiting the breach.

Finally, we do not directly protect against an attacker using external databases to check for consistency. If the structure of these external databases are public, however, the partitioning phase ensures all the interrelated fields are contained inside the same atom such that the atom is internally and externally consistent.

6. USING NOYB IN FACEBOOK

We now delve into the finer details of applying NOYB to Facebook. As mentioned previously, our goal is to preserve user privacy while allowing users to make use of Facebook to advertise themselves to their extended social network.

Facebook profiles contain over 40 fields of personal information; we mention some illustrative examples of how these are partitioned into atoms that are small enough to not leak much information, and yet large enough to be internally consistent. For instance, the name and sex of a person are contained within a single atom for consistency. Similarly, the street address, city, state, country and the area codes of telephone numbers are contained within another atom. For fields that contain lists, such as personal interests, favorite music, movies and tv shows, each list element is a separate atom. Other fields such as birthdate, political and religion views, etc. each represent a separate atom. However, not all fields are partitioned into atoms.

The key omissions are the phone number (excluding the area code), the user part of email addresses, and instant messaging handles. This is because these elements are expected to be unique; the substitution process cannot guarantee that cipher-atoms will also be unique, and indeed due to the birthday paradox, it is likely that even a small userbase will generate some duplicates that may be noticed. Fortunately, there is little internal structure to these fields, which allows the substitution to be applied at the character level, with the dictionary comprising the alphabet (with character frequencies) [15].

Multiple dictionaries are maintained for atoms that are correlated with a person's age or gender (e.g. movies, tv shows etc.) to preserve the joint distributions. These dictionaries are indexed by the birthdate and sex. When encrypting the movies field, for instance, the input dictionary is that indexed by the real birthday and sex of the user, while the output dictionary is that indexed by the ciphered birthdate and sex. Changes in the birthdate and sex would induce correlated updates of the dependent fields, however, since neither is expected to change, the impact is minimal.

The cipher-text atoms are stored in the user's Facebook profile. Steganography applied to the user's profile photo provides the in-band residue channel, however, the technique must be resilient to the image resampling algorithm used by Facebook. The necessary keys are exchanged over email. Other Facebook users that have the key can locally decode the page, satisfying the goal of allowing users to advertise themselves to their extended social network without revealing their private information to Facebook. However, probabilistic encryption prevents searching for users.

Searching on specific fields may be enabled by trading off some privacy for functionality. The name, for instance, may

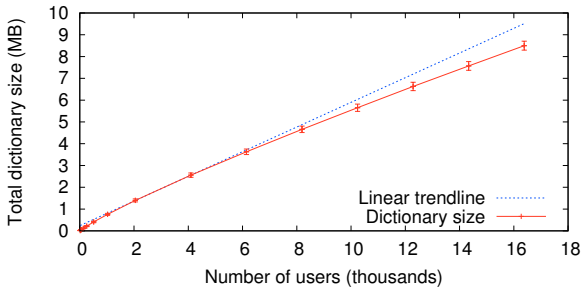

Figure 2: Dictionary grows sublinearly with users.

be encrypted deterministically for each community (e.g. a university) using a predetermined secret. This allows community members to search for a name by searching for the ciphered version without revealing the real name to Facebook.

7. IMPLEMENTATION

We implemented a proof-of-concept version of NOYB as a browser plugin for the Firefox web browser. Our implementation modifies Facebook pages by adding a button that encrypts the user's own profile. A second button added to other user's pages decrypts their profile. The plugin consists of 1400 lines of XUL code, and 500 lines of python code that uses AES in counter mode as the underlying cipher. The dictionaries necessary for the plugin are generated by a service we built; the service samples Facebook profile pages of NOYB users and non-users, and posts the dictionary to a public website that NOYB users can query anonymously. At present, our implementation does not manage keys and instead defers to the user at the time of encryption and decryption to enter the password, however, it is possible to modify our plugin to automatically distribute keys to the friends, and receive keys from friends through web based email services. We have manually verified that the encrypted profiles look plausible without revealing significant private information. We are quick to point out, however, that our experience is limited owing to our small userbase.

A second purpose of our implementation exercise is to study the feasibility of maintaining the dictionaries. Figure 2 shows the size of the dictionary as a function of the number of users sampled. We find that the size grows sublinearly reflecting overlapping values across different users. While the numbers are encouraging in that we expect a dictionary of a million users to initially be manageably small (344 MB), we do not, at present, have data to comment on the size of the dictionary over time as new atoms are appended. Nevertheless we believe that if the size grows by up to 3–4 orders of magnitude over the lifetime of the online service, a distributed peer-to-peer dictionary infrastructure could be necessary.

8. RELATED WORK

User privacy has been an active field of research stretching back to the beginnings of public and commercial adoption of the Internet. Pretty Good Privacy (PGP) [20] applies end-to-end public-key cryptography to emails. TLS [7] applies the same to end-to-end interactive communication channels. While NOYB is similarly end-to-end in the privacy it provides, it is different from the aforementioned systems in that it is not completely opaque to the middle, and can therefore make greater use of the functionality provided by the online service.

A second class of privacy preserving services operate in the middle of the network. Such services include anonymizing proxies [8], and dark nets [5]. These services provide an all-or-nothing model to privacy, where the privacy preserving mechanism, such as stripping of a HTTP cookie, either completely shields the user potentially breaking the application, or, when absent, leaves the user completely vulnerable. NOYB, instead, is tightly coupled with the application allowing fine-grained control over user privacy while balancing the functionality preserved.

Complementary to NOYB is the large amount of research in ciphers, key management, steganography, and DHTs. NOYB shares a resemblance to the pseudorandom character substitution cipher in [15]. Particularly of use to NOYB are existing and future broadcast key management algorithms [3], strong underlying ciphers [6], resillient steganographic techniques [11], and distributed store-lookup infrastructures [14], which can be used to implement the external mechanisms that NOYB relies on.

9. SUMMARY AND FUTURE WORK

In this paper we have described the NOYB mechanism, which provides fine-grained control over user privacy in online services while preserving much of the functionality provided by the service. We apply this generic approach to the specific case of protecting privacy in Facebook. While we are still exploring the implications of such an approach; based on a proof-of-concept implementation we conclude that the core idea is feasible, and incrementally deployable by groups of users without explicit cooperation from the online service.

Beyond social networks, there are a number of interesting research directions we hope to explore. We only briefly list them here. Foremost among these is to apply NOYB to online services that focus on search. Another is to apply NOYB in a way that allows online services to provide customized content, such as targeted advertisements, to users without the service being able to compromise the private information on which the customization is based. Finally, NOYB appears to be a promising first step towards a new design paradigm of online services where the user plays an active role in performing the sensitive operations on data, while the service takes care of the rest.

Acknowledgements

The authors would like to thank Tom Roeder for his input in the design. Additional thanks to the Facebook users who contributed their profile data for analysis.

10. REFERENCES

[1] Acquisti, A., and Gross, R. Imagined Communities Awareness, Information Sharing, and Privacy on the Facebook . In *Proceedings of the Privacy Enhancing Technologies Symposium (PET '06)* (Cambridge, UK, June 2006).

[2] BBC News. Yahoo 'helped jail China writer' , Sept. 2005.

[3] Boneh, D., Gentry, C., and Waters, B. Collusion Resistant Broadcast Encryption With Short

Ciphertexts and Private Keys. In *Proceedings of the CRYPTO '05* (Santa Barbara, CA, Aug. 2005).

[4] Chong, S., Liu, J., Myers, A. C., Qi, X., Vikram, K., Zheng, L., and Zheng, X. Secure web applications via automatic partitioning. In *Proceedings of the 21th ACM Symposium on Operating Systems Principles* (Stevenson, WA, Oct. 2007).

[5] Clarke, I., Sandberg, O., Wiley, B., and Hong, T. W. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Proceedings of the Privacy Enhancing Technologies Symposium (PET '01)* (San Francisco, CA, Apr. 2001).

[6] Daemen, J., and Rijmen, V. *The Design of Rijndael: AES–the Advanced Encryption Standard*. Springer, 2002.

[7] Dierks, T., and Rescorla, E. RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1, Apr. 2006.

[8] Dingledine, R., Mathewson, N., and Syverson, P. TOR: The Second-Generation Onion Router. In *Proceedings of 13th conference on USENIX Security Symposium* (San Deigo, CA, Aug. 2004).

[9] Facebook Inc. Terms of Use, Nov. 2007.

[10] Felt, A., and Evans, D. Privacy Protection for Social Networking APIs. 2008.

[11] Katzenbeisser, S., and Petitolas, F. Information Hiding Techniques for Steganography and Digital Watermaking. *EDPACS 28*, 6 (2000), 1–2.

[12] Nissenbaum, H. Privacy as Contextual Integrity. *Washington Law Review 79*, 1 (Feb. 2004), 119–158.

[13] PCWorld. Facebook's Beacon More Intrusive Than Previously Thought, Nov. 2007.

[14] Rhea, S., Godfrey, B., Karp, B., Kubiatowicz, J., Ratnasamy, S., Shenker, S., Stoica, I., , and Yu, H. OpenDHT: A Public DHT Service and Its Uses. In *Proceedings of SIGCOMM'05* (Philadelphia, PA, Aug. 2005).

[15] Ritter, T. Substitution cipher with pseudo-random shuffling: The dynamic substitution combiner. *Cryptologia 14*, 4 (1990), 289–303.

[16] Robert Scoble. Facebook disabled my account, Jan. 2008.

[17] Robert Scoble. Facebook lets me back in..., Jan. 2008.

[18] Saltzer, J., and Schroeder, M. The Protection of Information in Computer Systems. *Proceedings of the IEEE 63*, 9 (Sept. 1975), 1278–1308.

[19] Stana, R. M., and Burton, D. R. Identity Theft: Prevalence and Cost Appear to be Growing. Tech. Rep. GAO-02-363, U.S. General Accounting Office, Washington, D.C, 2002.

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