Computer Communications xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Practical identity-based private sharing for online social networks

Q1 Filipe Beato*, Stijn Meul, Bart Preneel

ESAT/COSIC - KU Leuven and iMinds, Leuven, Belgium

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Online social networks
Privacy
Identity-based encryption

ABSTRACT

Online Social Networks (OSNs) constitute vital communication and information sharing channels. Unfortunately, existing coarse-grained privacy preferences insufficiently protect the shared information. Although cryptographic techniques provide interesting mechanisms to protect privacy, several issues remain problematic, such as, OSN provider acceptance, user adoption, key management and usability. To mitigate these problems, we propose a practical solution that uses Identity-Based Encryption to simplify key management and enforce data confidentiality. Moreover, we devise an Identity-Based outsider anonymous private sharing scheme to disseminate information among multiple users. Furthermore, we demonstrate the viability and tolerable overhead of our solution via an open-source prototype.

1. Introduction

Online Social Networks (OSNs), such as Facebook, Google+, and Twitter present a significant growth and have become a prominent communication channel for many millions of users. OSNs offer users an efficient and reliable channel to distribute and share information. At the same time, OSNs store large amounts of data which prompts several privacy concerns, in particular as it is possible to infer a considerable amount of sensitive information from the shared and stored content. Although users are allowed to configure "privacy preferences" to limit access and select which users or groups can access the shared content, these preferences are generally too coarse-grained and difficult to configure [1]. In addition, these preferences do not exclude providers along with the dangers of data beaches and leaks [2] nor government. As proved by recent events like the PRISM project [3] and the iCloud breach [4].

All these worrisome issues motivate the need for effective techniques to properly protect user's privacy in OSNs. Several solutions have been proposed advocating the use of cryptographic mechanisms to address the privacy issues, either by an add-on atop of existing OSNs [5–8], or by complete new privacy-friendly architectures [9], mainly decentralized [10,11]. In general, those solutions suffer from user adoption and key management issues as users are required to register and then share, certify and store public keys [12]. Günther et al. [13] formalize cryptographic models for private profile management achieving confidentiality and unlinkability, however their sharing information protocols similar complex key management do not protect privacy of the recipients. Completely new architectures represent a difficult step for users as the trade-off of moving away from the commonly used social ecosystem compared with the risk of losing interactions is high. Arguably, current centralized OSNs are here to stay and will continue to be actively used by millions of people. In light of recent events, such as Edward Snowden's whistle-blowing on US surveillance programs [3], OSN providers have an interest to maintain their users and a privacy-friendly image. Hence, it is important to protect user's sharing information, such as text and media content, as well as the identity of the recipients as it can contain private and sensitive information.

Main Idea. Identity Based Encryption (IBE) [14] solutions overcome the key management problem as the public key of the user can be represented by any valid string, such as the email, unique `id` and username. Therefore, by using a OSN username any savvy and concerned user can share encrypted content with other users who are not using the solution, thereby motivating curious ones to use the system as well. However, IBE-based systems require a trusted central Private Key Generator (PKG) server to generate the private parameters for each user based on the PKG master secret. Consequently, such an architecture only shifts the trusted party from the OSN to the PKG. This problem can be mitigated if the master secret is divided among multiple PKGs following a Distributed Key Generation (DKG) [15] protocol based on Verifiable Secret Sharing (VSS) [16]. A DKG protocol allows n entities to jointly generate a secret requiring that a threshold t of the n entities does not get compromised. In fact, each entity holds only a share of the master secret, that can be reconstructed by atleast t shares.

Many OSN users are represented on several OSNs, and potentially hold multiple public keys. In this way, the multi-PKG setting could

Q2 * Corresponding author. Tel.: +3216321818.
E-mail addresses: filipe.beato@esat.kuleuven.be (F. Beato), stijn.meul@esat.kuleuven.be (S. Meul), bart.preneel@esat.kuleuven.be (B. Preneel).

http://dx.doi.org/10.1016/j.comcom.2015.07.009
0140-3664/© 2015 Published by Elsevier B.V.

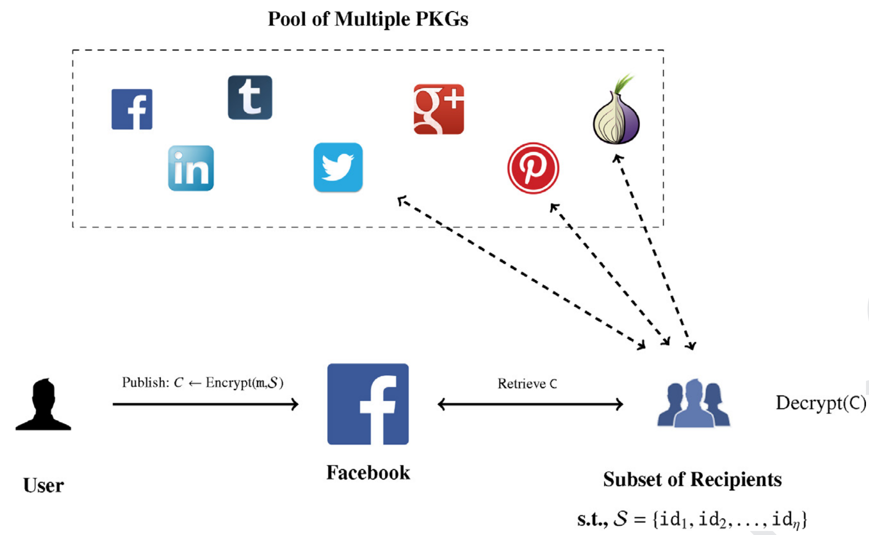

Fig. 1. Multiple (n, t)-PKG IBE for OSNs overview, for a message $\mathtt{m}$ published for the set $\mathcal{S}$ for $t = 3$.

be supported and maintained by several OSNs, in particular if considering the collaboration between competing OSN providers to be difficult and orthogonal to their business model. Fig. 1 shows an overview of the proposed model, in which users authenticate to t-PKGs of their choice; to retrieve private keys. This action is performed after the reception of encrypted content. For an additional level of security, PKG servers can also be represented by governmental entities from different continents, with no incentives to collaborate, thus overcoming more powerful adversaries using legal measures [17] that may at least affect t-PKGs.

Contribution. In this paper, we propose a novel practical solution using IBE with multiple semi-trusted PKGs on top of current OSNs. We highlight that multi-PKGs can be supported by several OSNs in view of business competition. We present an IBE broadcast encryption protocol with a multi-PKG model to support multiple recipients. Using a broadcast IBE-based mechanism users can share content with multiple recipients, thus, enforcing data confidentiality while hiding the recipient set. Furthermore, this solution is implemented on top of the Scramble Firefox extension [6], requiring a relatively small overhead.

Roadmap. The remainder of this paper is organized as follows. Section 2 gives a brief overview of the cryptographic background. Next, Section 3 presents the model followed by the description of the suggested solution in Section 4. Section 5 describes the implementation details, while Section 6 reviews related work. Finally, Section 7 summarizes and concludes the paper.

2. Background

In this section we briefly overview the cryptographic tools and building blocks used in this paper. For ease of explanation we omit the definitions of the underlying cryptographic primitives. This section can, however, be skipped with no loss of continuity.

2.1. Identity based encryption

The concept of Identity Based Encryption (IBE) was introduced by Shamir [14], with the main idea of using any string as the public key. IBE requires no certificates as users can rely on publicly known identifiers such as an e-mail address or a telephone number, thus, reducing the complexity of establishing and managing a public key infrastructure. Boneh and Franklin proposed the first practical IBE using bilinear pairings [18], later extended by Gentry [19].

A generic IBE scheme is composed of four randomized algorithms:

$\mathtt{IBE.Setup}(\lambda)$: On the input of a security parameter λ, outputs a master secret *msk* and the master public parameters $mpk \leftarrow$ *params*.
$\mathtt{IBE.Extract}(params, msk, \mathtt{id})$: Takes the public parameters *params*, the master secret *msk*, and an $\mathtt{id}$ and returns the private key $sk_{\mathtt{id}}$.
$\mathtt{IBE.Encrypt}(params, \mathtt{m}, \mathtt{id})$: Returns the encryption $\mathtt{C}$ of the message $\mathtt{m}$ on the input of the *params*, the $\mathtt{id}$, and the arbitrary length message $\mathtt{m}$.
$\mathtt{IBE.Decrypt}(params, sk_{\mathtt{id}}, \mathtt{C})$: Reconstruct $\mathtt{m}$ from $\mathtt{C}$ by using the secret $sk_{\mathtt{id}}$ and the public parameters. Otherwise return $\perp$.

The $\mathtt{IBE.Setup}$ and $\mathtt{IBE.Extract}$ algorithms are executed by a trusted Private Key Generator (PKG) server, whereas $\mathtt{IBE.Encrypt}$ and $\mathtt{IBE.Decrypt}$ are performed by two players, e.g., Alice and Bob. Consequently, key escrow is performed implicitly in the classic IBE scheme as the PKG holds the master secret key. The correctness property holds with overwhelming probability for all $sk_{\mathtt{id}} \leftarrow \mathtt{IBE.Extract}(params, msk, \mathtt{id}_i)$, such that, $\mathtt{m} = \mathtt{IBE.Decrypt}(sk_{\mathtt{id}}), (\mathtt{C} \leftarrow \mathtt{IBE.Encrypt}(\mathtt{m}, \mathtt{id}_i))$.

2.2. Anonymous broadcast encryption

The notion of Broadcast encryption (BE) was introduced by Fiat and Naor [20], as a public-key generalization to a multi-user setting. A BE scheme allows a user to encrypt a message to a subset $\mathcal{S}$ of users, such that, only the users in the set $\mathcal{S}$ are able to decrypt the message. The computational overhead of the BE is generally bounded to the size of the ciphertext and the number of recipients. To overcome the overhead issue, the set $\mathcal{S}$ of recipients is generally known. Barth et al. [21] and Libert et al. [22] extended the notion of BE and introduced the notion of Anonymous Broadcast Encryption (ANOBE) scheme, where the recipient set $\mathcal{S}$ remains private even to the members in the set. Fazio and Perera [23] suggested the notion of outsider anonymous BE that represents a more relaxed notion of ANOBE. Thus, a generic broadcast encryption (BE) scheme consists of four randomized algorithms:

$\mathtt{BE.Setup}(\lambda, n)$: On the input of a security parameter λ, generates the public parameters $params \leftarrow (mpk, msk)$ of the system.
$\mathtt{BE.KeyGen}(params, i)$: Returns the public and private key (pk_i, sk_i) for each user i according to the *params*.

BE.Encrypt($mpk, \mathtt{m}, \mathcal{S}$): Takes the set $\mathcal{S} = \{pk_i \dots pk_{|\mathcal{S}|}\}$, s.t., $\mathcal{S} \subset \mathcal{U}$ along with the secret message m and generates C.

BE.Decrypt($mpk, sk_i, \mathtt{C}$): Reconstructs m from C using the private key sk_i if the corresponding public key $pk_i \in \mathcal{S}$. Otherwise, return $\perp$.

Definition 1 (oANOBE)**.** An outsider anonymous broadcast encryption (oANOBE) scheme [23] is a BE with the extra property of recipient privacy, in which the users in the recipient set $\mathcal{S}$ are kept anonymous towards any user not in $\mathcal{S}$.

Definition 2 (ANOBE)**.** A fully anonymous broadcast encryption (ANOBE) scheme [21,22] is a BE with the extra property of recipient privacy, in which the users in the recipient set $\mathcal{S}$ are kept anonymous towards all users including other users in $\mathcal{S}$.

Note that the *pk* can be represented by the id string value from an Identity-Based scheme. Subsequently, the correctness property holds for all $\mathtt{id} \in \mathcal{S}$, such that, $sk_{\mathtt{id}} \leftarrow \mathtt{BE.KeyGen}(params)$, and $\mathtt{m} = \mathtt{BE.Decrypt}(sk_{\mathtt{id}}, (\mathtt{C} \leftarrow \mathtt{BE.Encrypt}(\mathtt{m}, \mathcal{S}))$.

2.3. Distributed key generation

Distributed Key Generation (DKG) was introduced by Pedersen [15] to allow a group of entities to collaboratively setup a secret sharing environment over a public channel. Secret sharing was introduced by Shamir [24] and consists of dividing a secret k into *n* shares among *n* entities, such that, only a subset of size greater than or equal to a threshold *t* can reconstruct k, where $t \geq n$. In practice, a random secret k is generated along with a polynomial $f(x)$ of degree $t - 1$ such that $f(0) = \mathtt{k}$, where the shares s_i are represented by different points on the polynomial. Any entity with *t* or more shares can reconstruct $f(x)$ using Lagrange interpolation, and subsequently find k. Chor et al. [16] suggested Verifiable Secret Sharing (VSS) scheme to allow anyone to verify that the right shares are used. The scheme was later extended by Feldman [25] and Pedersen [15].

For multiple parties to jointly generate a shared secret k, all entities are required to participate in a DKG scheme. Each entity *i* involved generates a different $\mathtt{k}_i$ and $f^i(x)$, distributing their own share and verifying all other shares s_{ij}. Hence, a generic DKG does not require a trusted party, as the master secret is computed as the aggregation of all the polynomials and can only be retrieved by joining *t* shares. A generic DKG protocol consists of two phases:

DKG.Setup(t, n): Every entity *i* generates a random secret $\mathtt{k}_i$ and computes a polynomial of degree $t - 1$. The entity *i* Distributes a valid share s_{ij} over all the other *j* entities, along with the commitment to the share. Each entity *j* verifies the shares and computes the new share $s_j = \sum_i s_{ij}$. The master secret is unknown by each party, and composed by the origin point on the sum of all polynomials $f^i(x)$.

DKG.Reconstruct(t): Each entity *i* broadcasts its share s_i, and with $t \leq n$ shares, one can reconstruct the master secret *s*.

The DKG protocol is secure assuming that no adversary is able to corrupt *t* or more parties. However, the uniformity of a key generated using the Pedersen DKG [15] cannot be guaranteed against a rushing adversary, i.e., adversaries contribute last in each run of the protocol [26]. Despite the biased distribution of the public key, this issue can be mitigated by increasing the security parameter [27].

3. Model

We consider a user *u* to be a member of one or several OSNs, and to be connected with other users in the same OSN by a friendship relationship [28]. Inherently, *u* aims to interact and share information m with other users in the same OSN. Each user holds a public and private key-pair, the public key is represented by the user id, whereas the private key is given by an Identity-Based server. The latter is composed of multiple PKG servers. Each user can be registered in multiple OSNs accumulating different ids, and thus different public keys. We assume the authentication between users and identity servers is performed under an authenticated channel, such as TLS, and uses a token similar to open id, such as, Facebook OAuth. For a stronger adversarial model these providers should operate under different jurisdictions to avoid coercion from the government to reveal their shares, for instance, Twitter (US), Spotify (Sweden/UK), Shazam (UK) or SoundCloud (Germany), Privalia (Spain). Nevertheless, an analysis of the security provided by a trans-jurisdictional distribution is beyond the scope of this paper.

3.1. Private sharing

We model our OSN private sharing scheme (OSN-PS) as a generalization of a BE scheme using IBE with multiple PKGs, aiming at sharing private information on current popular OSNs.

Definition 3 (OSN-PS)**.** For the universe of users $\mathcal{U} = \{\mathtt{id}_0, \dots, \mathtt{id}_N\}$ in a OSN, and a list of available $\Gamma = \{\mathtt{PKG}_0, \dots, \mathtt{PKG}_n\}$. Then, a OSN private sharing scheme (OSN-PS) Π is composed by four randomized algorithms: $\Pi \leftarrow$ {Setup, KeyGen, Publish, Retrieve}.

Π.Setup(λ, t, n): On the input of a security parameter λ, the threshold *t* and the number of PKGs *n*, generates the public parameters $params \leftarrow (mpk, msk)$ of the system.

Π.KeyGen($params, \Psi, \mathtt{id}_i$) Returns the private key $sk_{\mathtt{id}_i}$ for the user identity $\mathtt{id}_i$ according to the *params* and using a subset $\Psi \subseteq \Gamma$, s.t., $|\Psi| \geq t$.

Π.Publish($params, \mathtt{m}, \mathcal{S}$): Takes a subset $\mathcal{S}$, s.t., $\mathcal{S} \subset \mathcal{U}$ along with the secret message m and generates C.

Π.Retrieve($mpk, sk_{\mathtt{id}}, \mathtt{C}$): Reconstructs m from C using the private key $sk_{\mathtt{id}}$ if $\mathtt{id}_i \in \mathcal{S}$. Otherwise, return $\perp$.

3.2. Adversarial model

We consider an adversary to be any entity attempting to passively access the shared information m by monitoring the OSN, such as the communication sharing channel; with no motivational incentive to tamper with the content. This can be any curious user in the OSN, the OSN provider or even a government agency [3]. Such adversaries should not learn the content of the message and the identity of members in the recipient set $\mathcal{S}$, otherwise we consider the adversary breaks both confidentiality and recipient anonymity [21].

– Confidentiality. The confidentiality property holds if the OSN PS-scheme achieves ciphertext indistinguishability. In particular, if the adversary $\mathcal{A}$ does not win the following game between the Challenger Ch with a non-negligible probability. This is similar to the confidentiality modeled by Günther et al. [13]. The confidentiality property holds if the OSN PS-scheme achieves ciphertext indistinguishability. In particular, if the adversary $\mathcal{A}$ does not win the following game between the Challenger Ch with high probability. This is similar to the confidentiality modeled by Günther et al. [13].

Game 1 (OSN-PS Confidentiality)**.** Let $\Pi \leftarrow$ {Setup, KeyGen, Publish, Retrieve} be a OSN PS-scheme, $\mathcal{A}$ a probabilistic polynomial time (PPT) adversary, and Ch the challenger. We say that Π is (IND-CCA) secure if $\mathcal{A}$ wins the below game with Ch with negligible probability.

Init: Ch runs Setup(λ), and gives $\mathcal{A}$ the resulting *params*.

Setup: Ch generates keys for each potential recipient $i \in \mathcal{S}$, running $sk_i \leftarrow \mathtt{KeyGen}(\mathtt{params}, \mathtt{u_i})$, and sends each pk_i for $i \in \mathcal{S}$ to the $\mathcal{A}$.

Phase 1: The $\mathcal{A}$ adaptively performs queries to the Retrieve(C, *sk*) oracle.

Challenge: $\mathcal{A}$ sends to the $\texttt{Ch}$ two different messages $(\texttt{m}_0, \texttt{m}_1)$, s.t., $|\texttt{m}_0| = |\texttt{m}_1|$. $\texttt{Ch}$ picks a random bit $b \in \{0, 1\}$, runs $\texttt{C}' \leftarrow \texttt{Publish}(params, \mathcal{S}, \texttt{m}_b)$, and sends $\texttt{C}'$ to $\mathcal{A}$.
Phase 2: $\mathcal{A}$ adaptively issues additional decryption queries $\texttt{Retrieve}(\texttt{C}', sk)$, such that, $\texttt{C} \neq \texttt{C}'$.
Guess: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins if $b = b'$.

The $\mathcal{A}$ advantage to win the above game is defined as:

$$\mathsf{Adv}^{\mathsf{Ind}}_{\mathcal{A},\Pi} = |Pr[b = b'] - \frac{1}{2}|$$

– Recipient set anonymity. The high-level idea behind recipient set privacy is as follows. For any two recipient sets $\mathcal{S}_0$ and $\mathcal{S}_1$ an adversary $\mathcal{A}$ cannot distinguish between a ciphertext intended for the recipient set $\mathcal{S}_0$, and a ciphertext intended for the recipient set $\mathcal{S}_1$, given that $\mathcal{A}$ does not possess the secret key of any user in $\mathcal{S}_0 \cup \mathcal{S}_1$.

Game 2 (OSN-PS Recipient Set Anonymity)**.** A OSN-PS scheme $\Pi \leftarrow \{\texttt{Setup}, \texttt{KeyGen}, \texttt{Publish}, \texttt{Retrieve}\}$ is recipient anonymous (ANOPS) if a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ wins the following game with negligible probability:

Init: $\texttt{Ch}$ runs *Setup*(λ), and gives $\mathcal{A}$ the resulting *params*. $\mathcal{A}$ outputs $\mathcal{S}_0, \mathcal{S}_1 \in \mathcal{U}$, such that, $|\mathcal{S}_0| = |\mathcal{S}_1|$, and $(\mathcal{S}_0 \triangle \mathcal{S}_1) = \emptyset$.[1]
Setup: $\texttt{Ch}$ generates keys for each potential recipient i, running $sk_i \leftarrow \texttt{KeyGen}(\texttt{params}, \texttt{u}_i)$, and sends each pk_i for $i \in \mathcal{S}_0 \cap \mathcal{S}_1$ and sk_i for $i \in \mathcal{S}_0 \cup \mathcal{S}_1$ to the $\mathcal{A}$.
Phase 1: $\mathcal{A}$ adaptively issues decryption queries $q_1 = (i, \texttt{C})$, and $\texttt{Ch}$ returns $\texttt{Retrieve}(params, sk_i, \texttt{C})$.
Challenge: $\mathcal{A}$ gives the $\texttt{Ch}$ a message $\texttt{m}$. The $\texttt{Ch}$ picks a random bit $b \in \{0, 1\}$ and runs $\texttt{C}' \leftarrow \texttt{Publish}(params, \{u_i | u_i \in \mathcal{S}_b\}, \texttt{m})$, and sends $\texttt{C}'$ to $\mathcal{A}$.
Phase 2: $\mathcal{A}$ adaptively issues additional decryption queries $q_2 = (i, \texttt{C})$, such that $\texttt{C} \neq \texttt{C}'$.
Guess: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins if $b = b'$.

The advantage of $\mathcal{A}$ of winning the above game is defined as:

$$\mathsf{Adv}^{\mathsf{ANOPS}}_{\mathcal{A},\Pi} = \left| Pr[b = b'] - \frac{1}{2} \right|$$

In addition, we assume that such an adversary cannot compromise more than t identity servers (PKGs) or control the user-computing environment. Any malicious recipient who copy or forwards shared content is considered to break the social contract. Protection against traffic analysis or timing attacks is beyond the scope of this protocol.

3.3. Goals

We aim to protect OSN users privacy by ensuring confidentiality, data integrity and recipient anonymity [21]. In this way we allow users to enforce access control without having to rely on the privacy preferences offered by the OSN. At the same time, we aim at limited modifications to the OSN environment. In particular, we require as little effort as possible, and reduced prior knowledge from users in order to achieve a user-friendly scheme as defined by Balsa et al. [12]. In contrast to previous solutions, users are allowed to be in the recipient set by default as their public key is represented by the OSN identifier. Our main goals are summarized as follows:

- **Content privacy.** The content should be confidentiality protected from any unauthorized entities.
- **Recipients privacy.** The recipients of the messages should be hidden from any unauthorized entities.
- **Ease of key management.** The original OSN environment should not be altered since some OSN providers are probably not willing to support a more confidential architecture because it could hurt their business model.
- **Immediate deployment.** No additional changes to the OSN design and infrastructure of current OSNs.
- **Direct opt-In.** Registration to third-party key architectures or key exchange should be required to enable the system. In fact, users should be able to receive confidential messages upon registration to any OSN.

4. Practical outsider-anonymous private sharing scheme for OSNs

In this section, we describe our OSN outsider-anonymous private sharing scheme (oANOPS). The proposed solution is based on the IBE scheme from Boneh et al. [18] and a relaxed version of the broadcast scheme from Libert et al. [22]. The system relies on a DKG protocol as described by Pedersen [15] to bootstrap multiple PKGs. In addition, we converted the schemes from using Type 1 (i.e., $\mathbb{G}_1 = \mathbb{G}_2$) to Type 3 (i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$) pairings for efficiency [29] and because Type 1 pairings are no longer secure according Joux in [30].

The scheme allows users to publish any content while enforcing access rules by selecting the recipient set per content. Only authorized users can run the Retrieve and output access the content. We acknowledge that we do not support revocation, however, we assume that it is hard to protect content from malicious authorized recipients, who save, store, and broadcast the content.

4.1. Basic scheme

Let λ be the security parameter for a security level of l bits, and $\mathcal{S}$ the set of desired recipients u_i with corresponding $\texttt{id}_i$, such that $\mathcal{S} = \{u_1, .., u_\eta\}$ where $\eta = |\mathcal{S}|$. Let $\mathcal{G}$ be a generator that satisfies the Bilinear Diffie–Helman (BDH) assumption, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ the bilinear map such that $e(aP, bQ) = e(P, Q)^{ab}$ for $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_q$ as in [18] . In addition, let $\langle \texttt{C}, \texttt{T} \rangle \leftarrow \mathsf{E}_\mathsf{k}(M)$ be any secure authenticated symmetric encryption that takes as input the plaintext $M \in \{0, 1\}^*$ and a key $\mathcal{K} \in \{0, 1\}^*$, and generates ciphertext $\texttt{C} \in \{0, 1\}^*$ and authentication tag $\texttt{T} \in \{0, 1\}^\tau$ as output [31], such that, $\texttt{E} : M \times \mathcal{K} \rightarrow \{\texttt{C}, \texttt{T}\}$. Similarly, $\langle M, \texttt{T} \rangle \leftarrow \mathsf{D}_\mathsf{k}(\texttt{C})$ be the valid authenticated decryption that takes ciphertext $\texttt{C}$ as input and computes the plaintext M along with an authentication tag $\texttt{T}$. Thus, our oANOPS scheme Π for OSNs is composed by four randomized algorithms: $\Pi \leftarrow \{\texttt{Setup}, \texttt{KeyGen}, \texttt{Publish}, \texttt{Retrieve}\}$.

$\texttt{Setup}$ (λ, t, n): Outputs the public *params* of the system with respect to the security parameter λ, a list of available PKGs $\Gamma = \{\texttt{PKG}_0, \ldots, \texttt{PKG}_n\}$, such that $|\Gamma| = n$, for the threshold t.

1. On input of security parameter λ generate a prime q, two groups $\mathbb{G}_1, \mathbb{G}_2$ of order q satisfying the BDH assumption, and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Choose random generators $P \in \mathbb{G}_1$, and $Q \in \mathbb{G}_2$.
2. Choose the hash functions $\texttt{H}_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$, $\texttt{H}_2 : \mathbb{G}_T \rightarrow \{0, 1\}^l$, $\texttt{H}_3 : \{0, 1\}^l \times \{0, 1\}^l \rightarrow \mathbb{Z}_q^*$, and $\texttt{H}_4 : \{0, 1\}^l \rightarrow \{0, 1\}^l$, modeled as random oracles.
3. Each $\texttt{PKG}_j \in \Gamma$ generates $n - 1$ shares σ_{jv} of a Feldman VSS scheme by executing Pedersen DKG, and redistributing the $n - 1$ shares σ_{jv} with the other v PKGs.
4. $\texttt{PKG}_j$ publishes $P^{(j)}_{pub} = s_j P$, s.t., $s_j = \sum_{v=1}^{n} \sigma_{jv}$.
5. Select a semantically secure authenticated $\langle \texttt{C} \parallel \texttt{T} \rangle \leftarrow \texttt{E}(\cdot), \texttt{D}(\cdot)$, so that $\texttt{C}$ represents the encrypted output and $\texttt{T}$ the authenticity tag.

The master secret key $msk = \sum_{j \in \Psi} b_j s_j$ for $b_j = \prod_{z \in \Psi} \frac{z}{z-j}$ cannot be retrieved unless a subset $\Psi \subseteq \Gamma$ is of size at least t, s.t.,

[1] $\mathcal{S}_0 \triangle \mathcal{S}_1$ represents the symmetric difference, such that: $\mathcal{S}_0 \triangle \mathcal{S}_1 = \{x : (x \in \mathcal{S}_0) \oplus (x \in \mathcal{S}_1)\}$

$|\Psi| \geq t$. The following parameters are published publicly:

$$params = \{p, q, \mathbb{G}_1, \mathbb{G}_2, e, P, Q, \mathtt{H}_1, \mathtt{H}_2, \mathtt{H}_3, \mathtt{H}_4, t, n, P_{pub}^{(0)}, \ldots, P_{pub}^{(n)}\}$$

$\mathtt{KeyGen}(\Psi = \{\mathtt{PKG}_0, \ldots, \mathtt{PKG}_t\}, \mathtt{id}_i)$: On input of a user $\mathtt{id}_i$ the subset Ψ of size t of PKG servers, generates a valid private key for $\mathtt{id}_i$.

1. User with identifier $\mathtt{id}_i$, authenticates to a subset Ψ, s.t., $|\Psi| \geq t$, or all PKGs and sends $\mathtt{id}_i$.
2. Each $\mathtt{PKG}_j \in \Psi$ determines the respective secret share s_j by computing $Q_{\mathtt{id}_i} = \mathtt{H}_1(\mathtt{id}_i)$, and $Q_{priv,\mathtt{id}_i}^{(j)} = s_j Q_{\mathtt{id}_i}$.
3. The user $\mathtt{id}_i$ computes the shared public parameter P_{pub} using the Lagrange coefficients b_j as follows:
$$P_{pub} = \sum_{j \in \Psi} b_j P_{pub}^{(j)} \quad \text{for} \quad b_j = \prod_{z \in \Psi} \frac{z}{z - j}$$
4. All PKGs in Ψ return $Q_{priv,\mathtt{id}_i}^{(j)}$ to the corresponding user $\mathtt{id}_i$ over a secure channel.
5. Each user verifies for each $Q_{priv,\mathtt{id}_i}^{(j)}$ value whether,
$$e\left(Q_{priv,\mathtt{id}_i}^{(j)}, P\right) \stackrel{?}{=} e\left(Q_{\mathtt{id}_i}, P_{pub}^{(j)}\right)$$
Finally, the user with $\mathtt{id}_i$ calculates the associated private key $sk_{\mathtt{id}_i}$ using the Lagrange coefficients b_j as follows:
$$sk_{\mathtt{id}_i} = \sum_{j \in \Psi} b_j Q_{priv,\mathtt{id}_i}^{(j)}$$

In this way, no user nor PKG learns the master key *msk* of the system. In fact, an adversary is required to corrupt at least t or more parties to reconstruct *msk*. This algorithm combines DKG.Reconstruct, IBE.Extract and BE.KeyGen algorithms.

$\mathtt{Publish}(params, \mathcal{S}, \mathtt{m})$: Takes the message $\mathtt{m}$, the subset $\mathcal{S}$ of size η and the public parameters *params*, output a broadcast message $\mathtt{C}$.

1. Generate a random symmetric session key $\mathsf{k} \leftarrow \{0, 1\}^l$.
2. Choose a random value $\rho \in \{0, 1\}^l$ and compute r as a hash of concatenated values $r = \mathtt{H}_3(\rho, \mathsf{k})$
3. For each recipient $\mathtt{id}_i \in \mathcal{S}$, compute the ciphertext, running the IBE.Encrypt algorithm, as follows.
$$w_i = \rho \oplus \mathtt{H}_2(g_{\mathtt{id}_i}^r) \quad \text{where } g_{\mathtt{id}_i} = e\left(Q_{\mathtt{id}_i}, P_{pub}\right) \in \mathbb{G}_T$$
4. Let W be a random permuted concatenation of w_i, $v \leftarrow k \oplus \mathtt{H}_4(\rho)$, and $U \leftarrow rP$, then the authenticated data c_1 is computed as,
$$c_1 \quad = \quad \{U \parallel v \parallel W\} \text{ s.t. } W = \{w_1 \parallel w_2 \parallel \ldots \parallel w_{|\mathcal{S}|}\}$$
5. Apply authenticated symmetric encryption on $\mathtt{M}$, the concatenation of the intended recipient set $\mathcal{S}$ and the plaintext message $\mathtt{m}$, such that $\mathtt{M} = (\mathtt{m} \parallel \mathcal{S})$.
$$\langle c_2, \mathtt{T} \rangle \leftarrow \mathsf{E}_\mathsf{k}(\mathtt{M})$$
6. Publish the result $\mathtt{C} = \{c_1 \parallel c_2 \parallel \mathtt{T}\}$ on the OSN.

$\mathtt{Retrieve}(params, sk_{\mathtt{id}}, \mathtt{C})$: on input of the broadcast message $\mathtt{C}$ and the private key $sk_{\mathtt{id}}$ of user $\mathtt{id}_i$, reconstruct the plaintext message $\mathtt{m}$. This algorithm comprises the {IBE,BE}.Decrypt algorithms. Therefore, the user retrieves $\mathtt{C}$ from the OSN, and for each $w_i \in W$ performs the following:

1. Compute $w_i \oplus \mathtt{H}_2(e(sk_{\mathtt{id}}, U)) = \rho$ for $sk_{\mathtt{id}}$, and $v \oplus \mathtt{H}_4(\rho) = \mathsf{k}$
2. Set $r = \mathtt{H}_3(\rho, \mathsf{k})$. Verify $U \stackrel{?}{=} rP$. If the check fails, try next w_i, and return to 1.
3. Retrieve $\langle \mathtt{M}, \mathtt{T}' \rangle \leftarrow \mathsf{D}_\mathsf{k}(c_2)$
4. Verify whether $\mathtt{T}' \stackrel{?}{=} \mathtt{T} \in \mathtt{C}$, and return $\mathtt{m}$. Otherwise return $\perp$.

Correctness. The OSN oANOPS scheme is correct if for every member $\mathtt{id}_i \in \mathcal{S}$, s.t., $sk_{\mathtt{id}_i} \leftarrow \mathtt{KeyGen}(\{\mathtt{PKG}_0, \ldots, \mathtt{PKG}_t\}, \mathtt{id}_i)$, then $\mathtt{m} = \mathtt{Retrieve}(params, sk_{\mathtt{id}_i}, \mathtt{Publish}(params, \mathcal{S}, \mathtt{m}))$.

1. Let $w_i = \rho \oplus \mathtt{H}_2(g_i^r)$, where $g_i^r = e(Q_{\mathtt{id}}, P_{pub})^r \in \mathbb{G}_T$, $P_{pub} = \sum_{j \in \Psi} b_j P_{pub}^{(j)}$, $Q_{priv,\mathtt{id}_i}^{(j)} = s_i Q_{\mathtt{id}_i}$, and $sk_{\mathtt{id}} = \sum_{j \in \Psi} (b_j s_i Q_{\mathtt{id}_i})$. Then:
$$\begin{aligned} w_i \oplus \mathtt{H}_2(e(sk_{\mathtt{id}}, U)) &= \rho \oplus \mathtt{H}_2(g_i^r) \oplus \mathtt{H}_2(e(sk_{\mathtt{id}}, rP)) \\ &= \rho \oplus \mathtt{H}_2(e(Q_{\mathtt{id}_i}, P_{pub})^r) \oplus \mathtt{H}_2(e(sk_{\mathtt{id}}, rP)) \\ &= \rho \end{aligned}$$
2. Let $v \oplus \mathtt{H}_4(\rho) = \mathsf{k} \oplus \mathtt{H}_4(\rho) \oplus \mathtt{H}_4(\rho) = \mathsf{k}$.
3. Retrieve $\mathtt{M}/\perp, \mathtt{T}' \leftarrow \mathsf{D}_\mathsf{k}(c_1)$.

4.2. Replying and placing comments

It is common on OSNs for users to post replies and comments to the previously shared content $\mathtt{m}$. As users in the recipient set $\mathcal{S}$ are able to reconstruct the symmetric session key k, it is possible to encrypt the new comment with k. As in security using the same key is not advisable, a hash chain can be used, for instance, the first reply would be $H(k)$, then $H(H(k))$. In this way, a conversation among users can be build and new users can be added at the middle of the conversation just by receiving the respective hash value of the joint point without learning previous shared information. This is possible due to the one-way secure hash functions property, as it is infeasible to any adversary to reverse the hash and obtain a previous node of the chain.

4.3. Evaluation

We now evaluate the proposed OSN oANOPS scheme in terms of key management, security, anonymity, and complexity. In light, we show that the proposed scheme avoids key escrow, ensures confidentiality of the shared information $\mathtt{m}$, and provides recipient set anonymity towards non players and the PKGs. Note that, using IBE allows any user in the OSN to be part of the recipient set $\mathcal{S}$ before registering in the system. In addition, users can reuse (a hash of) the same symmetric key k during the comments and discussion phase.

Complexity. In terms of efficiency, users are required to decrypt w_i on average $|\mathcal{S}|/2$ times before obtaining the symmetric key k. The size complexity is linearly bounded to the size of the recipient set $\mathcal{S}$, i.e., $\mathcal{O}(\mathcal{S})$. In contrast, the complexity of key storage is minimal, requiring only the need to store the private keys, as the public keys of the users are represented by their public $\mathtt{id}$s, and the session key is encrypted with the content.

Security analysis. As the OSN ANO-PS scheme consists of secure underlying key privacy IBE, and authentication encryption schemes, the semantic security follows directly.

Theorem 1. *If the OSN oANO-PS-IBE scheme is correct, the DKG protocol is secure such that no more than t-PKGs gets compromised, the IBE scheme is CCA-secure and CCA-key private, and the* $\mathsf{E}(\cdot)$ *is a secure authenticated encryption scheme. Then a PS-IBE scheme is CCA outsider recipient private.*

Proof sketch: The confidentiality, integrity, and outsider recipient anonymity hold as a consequence of the security of the underlying authenticated encryption scheme. In particular, the session key can only be obtained if the recipient holds the corresponding secret key $sk_{\mathtt{id}}$, assuming the IBE-scheme is also semantically secure, i.e., IND-CCA.

Regarding recipient privacy, according to Theorem 1 a OSN oANO-PS-scheme is recipient privacy if the underlying constructions fulfill

Please cite this article as: F. Beato et al., Practical identity-based private sharing for online social networks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.07.009

certain requirements. As shown by Boneh and Waters [18], the underlying IBE is semantically secure under an adaptive adversary. As demonstrated by Paterson and Srinivasan [32] an IBE scheme is CCA-key private, and PKG anonymous if its also IND-CCA secure. Hence, if the chosen authentication encryption scheme is semantically secure, e.g., AES-GCM, then we show that our scheme is recipient private. As the OSN oANO-PS scheme also shares $\mathcal{S}$ along with the message we conclude that the scheme is outsider-anonymous. However, as the ciphertext size increases linearly with the size of $\mathcal{S}$, a powerful adversary may infer the cardinality of the set. We also note that we aim at an outsider-anonymous recipient privacy so that it does not guarantee privacy against users in $\mathcal{S}$ authorized to decrypt the data, as modeled by Günther et al. [13].

A user is able to detect malicious behavior of any PKG from the public commitments of the Feldman VSS [25]. It is also required that at least t from n PKGs do not get compromised. In case the OSN providers would maintain the PKG infrastructure, one could rely on the assumption that direct business competitors do not collude nor get legally coerced. Furthermore, the authentication and identity verification to the different servers can be done via, for instance, an open id token. This token could be generated as a proof of identity by any of the OSN providers. In addition, according to Gennaro et al. [26] Pedersen DKG is vulnerable to rushing adversaries that wish to learn extra information about the keys, this is however mitigated by increasing the security parameter [27].

Key management. In contrast to the other versions of PS-schemes, the IBE version requires very little to any effort for key distribution, while the public key (`id`) verification is bound to the OSN identity, along with authentication to the different PKGs. The DKG approach solves the key escrow issues that come with generic Identity-Based solutions. In contrast to classic public key infrastructure, if a public key is revoked, the user would no longer be able to use that identifier for encryption, e.g., Facebook ID. Therefore, to support revocation an expiration date is concatenated to the identifier [18], requiring an extra periodic key update process. Similarly to the PS-BE scheme, the access control rights are selected per content, thereby allowing group revocation to be represented by removal of the revoked user id. Similarly to PS-BE version, revocation is just applied to future content, providing no forward security.

4.4. Possible extensions

Now we discuss some possible improvements and extensions to improve the efficiency, protect the cardinality of the recipient set privacy, and offer the extra property of undetectability.

Efficiency. Barth et al. [21] and Libert et al. [22] propose using a tag based system to hint users where their symmetric key can be found, and improve the efficiency of the `Retrieve` phase. However, as a design choice we deliberately decided to not implement such property in the scheme as it introduces a linear dependency from extra public parameters to the users, i.e., there are extra public parameters that need to be shared and verified, and extra $N * |Tag|$-size to the ciphertext. In addition, to reduce ciphertext size several broadcast encryption scheme, such as Fazio et al. [23], make use of binary tree construction. Such solutions, however, require a fixed size universe of users.

Recipient set cardinality privacy. Although an adversary from outside $\mathcal{S}$ is not able to learn the identity of the recipients in the set $\mathcal{S}$, it learns the cardinality of $\mathcal{S}$. A possible solution is to use dummies, i.e., extra random w_i values. By padding W with random values $w'_i \leftarrow^R \{0, 1, \}^l$ increases the recipient privacy at the cost of ciphertext size and complexity during the `Retrieve` phase.

Undetectability. Although the confidentiality of $\mathtt{m}$ is guaranteed, an adversary is able to detect that secret information is being shared. In particular, when the OSN is the main communication channel other curious friends and the provider are able to detect, and later blocked by the latter. Recently, Beato et al. [33] modeled the property of undetectability in OSNs and provided a general solution to achieve it. To do so, they allow users to post a social indistinguishable text *st*, storing the encrypted $\mathtt{m}$ in an additional storage server. The *st* is then used as an index on a mapping servers used to retrieve the location of the storage server and subsequently $\mathtt{m}$.

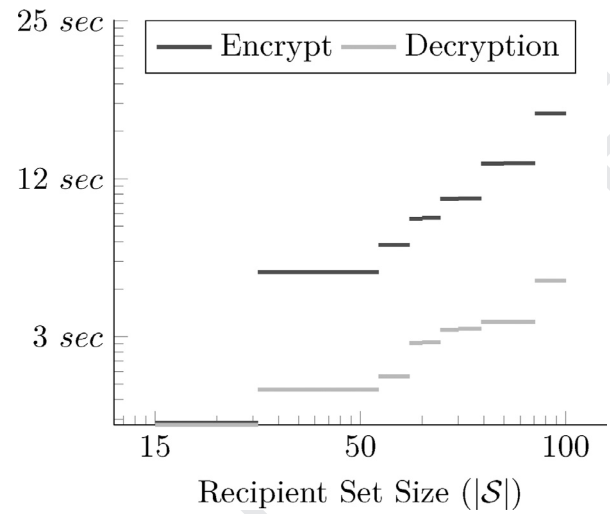

Fig. 2. The average execution time (in log scale) of the OSN ANOPS scheme for varying sizes of the recipient set.

5. Practicalities

To demonstrate the viability of our solution, we implemented a proof-of-concept prototype of the distributed identity based broadcast encryption scheme for OSNs.[2] In this section, we discuss the implementation details and the performance results of the cryptographic blocks.

5.1. Implementation

For the client component we modified the cryptographic library from Scramble [6] as it is an available open source privacy preserving project. In addition, Scramble is implemented as a Firefox Extension compatible with Firefox 14+, but as it is written in simple Javascript it could easily be ported to other browsers, e.g., Chrome. We implemented the multiple PKG servers in PHP. The bilinear pairing and cryptographic blocks for the PKG and the client component are implemented using the multi-precision MIRACL library [34]. To overcome the limitation of accessing binary code from a browser extension implementation, a local client-server socket implementation was used to perform the cryptographic requests to the developed scheme using the MIRACL library. For the DKG library we used the available implementation from Kate and Goldberg [35,36] to generate the collective master secret key for the $(n, t)-$PKG servers. AES-GCM [31] was used for the authenticated encryption. The Facebook username was used, i.e., $\mathtt{id} = facebook.com/user.name$, was used as the public key.

5.2. Performance

Experiments were conducted on a Intel Core 2.4 GHz i5 processor with 8Gb of 1600 MHz DDR3L onboard memory. Fig. 2 illustrates the execution times for the scheme proposed in Section 4 for $\lambda = 256$bits. Each recipient has to decrypt W_i an average of $N/2$ times to retrieve the secret and subsequently decrypt the secret message $\mathtt{m}$. Note that

[2] Source of our implementations is available upon request.

the efficiency comes at the cost of the recipient anonymity $\mathcal{S}$, as for hiding the $\mathcal{S}$ it is required to produce more IBE.Encrypt calls, while more efficient broadcast encryption schemes require constant time decryption and overhead [37].

We also analyzed the execution times of the IBE scheme, as it represents the most costly part of the scheme. Furthermore, our solution uses the random oracle assumption to improve the efficiency when compared with the standard model, i.e., Gentry [19]. Nevertheless, we believe that our solution presents a tolerable cost to average users with 100 friends and a usual group size of 15 [38].

6. Related work

The increased popularity of Online Social Networks (OSNs) and the amount of disseminated information prompted several privacy concerns. Guha et al. [7] proposed NOYB a solution that replaces the personal details of users by fake information. Later, FaceCloak [8] and Scramble [6] make use of cryptographic mechanisms to enforce privacy to the published information. Further, Persona [5] and EaSiER [39] suggest an attribute based encryption scheme for social networks. Günther et al. [13] suggested a private profile management cryptographic model serving as a building block for privacy in social interactions alongside with formal security definitions on confidentiality and unlinkability. In addition, two different profile management schemes are prosed based on symmetric cryptography and Gentry and Waters broadcast encryption scheme [19], achieving both confidentiality and unlinkability. However, their construction requires users to hold full power and manage their profile data as in decentralized networks minimizing the communication and storage overhead, whereas recipient anonymity is not addressed. However, all the aforementioned solutions suffer from a complex key management infrastructure while do not protecting the identities of the recipients.

Other solutions take a more drastic approach by proposing novel, privacy-friendly architectures meant to replace existing platforms [9–11]. Besides the privacy protection offered, these solutions face a reduced user willingness to adopt to a new platform.

Recently, Jung et al. [40] proposed a key management scheme based on dynamical IBE for decentralized OSNs. Their scheme, however, presents several problems. Foremost contains a single point of failure as a trusted party should generate the secret keys for a given id. This proposal still requires an additional public key that needs to be certified and shared among other users for the broadcasting, thus, not solving the key management issue.

In general all previous schemes require public parameters that should be shared and verified by users. By employing an Identity-based scheme we allow users to motivate their friends to use the solution, as registered users can already encrypt messages to unregistered friends.

Different cryptographic solutions have been proposed addressing other specific privacy problems in OSNs. For protecting content privacy on the friend search and common friend finder scenarios, De Cristofaro et al. [41] introduced private contact discovery protocol. The protocol enables two users of a OSN to learn their common contacts without learning any of the other friends. Later, Nagy et al. [42] extended [41] to the finding friends problem, using private set intersection techniques. The protocol allows users to privately generate and share their list of friends such that other friends can compare and find common friends in the honest-but-curious model.

7. Conclusion

Identity Based Encryption (IBE) solutions provide desirable properties to construct mechanisms to deliver privacy in OSNs. The minimal additional architectural support and the increased ease of key management represent a major motivation to implement IBE in OSNs. We developed an Identity-Based outsider anonymous private sharing (oANOPS) scheme that protects user shored content in OSNs. With such scheme, we show that using secret sharing and multi-PKGs there is no need to have a single trusted party, assuming that at least $t-1$ of the PKGs are compromised, as well as users key exchange and verification. Furthermore, assuming the competing business models of OSNs, the multiple PKG infrastructure can be maintained by several OSN providers. This can be motivated by an additional and attractive privacy-friendly label, thus creating more incentives towards privacy concerned users. In addition, users are provided with the option to use multiple identities, that they can use interchangeably among OSNs, e.g., use Twitter id as a public key in Facebook. In contrast to previous solutions, it is possible to share content with users not holding private keys to their identity as the valid public key is directly represented by their id in the OSN. This forces curious users to register if they wish to view the protected content shared with them. Lastly, we have extended Scramble and demonstrated that such extension presents a tolerable overhead to end-users.

Future work. There are some important open challenges that call for further research. We endeavor to obtain a full open source project that supports different browsers for a larger user adoption. Items like a more detailed security discussion and efficiency improvement are also important and required. In addition, for the authentication and proof of identity we foresee several open challenges during key generation and update.

Acknowledgments

We gratefully acknowledge Roel Peeters, Kimmo Halunen, Ruan de Clerq, and the anonymous reviewers for their insightful comments and suggestions. This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007), by the Flemish iMinds projects, and by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. The author Filipe Beato is supported by the FCT doctoral grant SFRH/BD/70311/2010.

References

[1] J. Bonneau, S. Preibusch, The privacy jungle: On the market for data protection in social networks, Econ. Inf. Secur. Privacy (2010) 121–167.
[2] M. Fischetti, Data theft: hackers attack, Sci. Am. 305 (100) (2011).
[3] W. Post, NSA slides explain the PRISM data-collection program, (June 6, 2013 http://wapo.st/J2gkLY. accessed 6.09.13).
[4] D. Lewis, icloud data breach: Hacking and celebrity photos, (Sept. 2, 2014 http://onforb.es/1Cmngvl, accessed 6.10.14).
[5] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, D. Starin, Persona: an online social network with user-defined privacy, SIGCOMM Comput. Commun. Rev. 39 (4) (2009) 135–146, doi:10.1145/1594977.1592585.
[6] F. Beato, M. Kohlweiss, K. Wouters, Scramble! your social network data, in: S. Fischer-Hübner, N. Hopper (Eds.), PETS, Lecture Notes in Computer Science, vol. 6794, Springer, 2011, pp. 211–225.
[7] S. Guha, K. Tang, P. Francis, Noyb: privacy in online social networks, in: Proceedings of the WOSN, ACM, New York, NY, USA, 2008, pp. 49–54.
[8] W. Luo, Q. Xie, U. Hengartner, Facecloak: an architecture for user privacy on social networking sites, in: Proceedings of the IEEE CSE, IEEE, Washington, DC, USA, 2009, pp. 26–33.
[9] E.D. Cristofaro, C. Soriente, G. Tsudik, A. Williams, Hummingbird: privacy at the time of twitter, in: Proceedings of the IEEE Symposium on Security and Privacy, IEEE, 2012, pp. 285–299.
[10] L.A. Cutillo, R. Molva, M. Önen, Safebook: a distributed privacy preserving online social network, in: Proceedings of the WOWMOM, 2011, pp. 1–3.
[11] J. Dwyer, Four nerds and a cry to arms against Facebook, (May 11, 2010. http://nyti.ms/1hc60kv. accessed: 3.12.13).
[12] E. Balsa, L. Brandimarte, A. Acquisti, C. Diaz, S.F. Gürses, Spiny CACTOS: OSN users attitudes and perceptions towards cryptographic access control tools, in: Proceedings of the Workshop on Usable Security, in: Lecture Notes in Computer Science, Springer-Verlag, San Diego,CA,USA, 2014, p. 10.
[13] F. Günther, M. Manulis, T. Strufe, Cryptographic treatment of private user profiles, in: G. Danezis, S. Dietrich, K. Sako (Eds.), Proceedings of the RLCPS – FC 2011 Workshops, LNCS, vol. 7126, Springer, 2011, pp. 40–54.
[14] A. Shamir, Identity-based cryptosystems and signature schemes, in: G.R. Blakley, D. Chaum (Eds.), Proceedings of the CRYPTO, Lecture Notes in Computer Science, vol. 196, Springer, 1984, pp. 47–53.

Please cite this article as: F. Beato et al., Practical identity-based private sharing for online social networks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.07.009

[15] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, in: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, in: CRYPTO '91, Springer-Verlag, London, UK, UK, 1992, pp. 129–140.
[16] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract), in: Proceedings of the FOCS, 1985, pp. 383–395.
[17] C. Matyszczyk, If your account is subpoenaed, Facebook sends police, well, everything, 2012, (http://preview.tinyurl.com/facebook-subpoena).
[18] D. Boneh, M.K. Franklin, Identity based encryption from the Weil pairing, IACR Cryptol. ePrint Arch. 2001 (2001) 90.
[19] C. Gentry, Practical identity-based encryption without random oracles, in: S. Vaudenay (Ed.), Proceedings of the Advances in Cryptology - EUROCRYPT 2006, Lecture Notes in Computer Science, vol. 4004, Springer Berlin Heidelberg, 2006, pp. 445–464.
[20] A. Fiat, M. Naor, Broadcast encryption, in: D.R. Stinson (Ed.), Proceedings of the CRYPTO, Lecture Notes in Computer Science, 773, Springer, 1993, pp. 480–491.
[21] A. Barth, D. Boneh, B. Waters, Privacy in encrypted content distribution using private broadcast encryption, in: G.D. Crescenzo, A.D. Rubin (Eds.), Proceedings of the Financial Cryptography, Lecture Notes in Computer Science, vol. 4107, Springer, 2006, pp. 52–64.
[22] B. Libert, K.G. Paterson, E.A. Quaglia, Anonymous broadcast encryption: adaptive security and efficient constructions in the standard model, in: M. Fischlin, J. Buchmann, M. Manulis (Eds.), Proceedings of the Public Key Cryptography, Lecture Notes in Computer Science, vol. 7293, Springer, 2012, pp. 206–224.
[23] N. Fazio, I.M. Perera, Outsider-anonymous broadcast encryption with sublinear ciphertexts, IACR Cryptol. ePrint Arch. 2012 (2012) 129.
[24] A. Shamir, How to Share a Secret, Commun. ACM 22 (11) (1979) 612–613. http://doi.acm.org/10.1145/359168.359176.
[25] P. Feldman, A practical scheme for non-interactive verifiable secret sharing, in: Proceedings of the 28th Annual Symposium on Foundations of Computer Science, in: SFCS '87, IEEE Computer Society, Washington, DC, USA, 1987, pp. 427–438, doi:10.1109/SFCS.1987.4.
[26] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, Secure distributed key generation for discrete-log based cryptosystems, J. Cryptol. 20 (1) (2007) 51–83, doi:10.1007/s00145-006-0347-3.
[27] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, Secure applications of pedersen's distributed key generation protocol, in: M. Joye (Ed.), Proceedings of the Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, vol. 2612, Springer, 2003, pp. 373–390.
[28] D. Boyd, N. Ellison, Social network sites: definition, history, and scholarship, J. Comput. Mediat. Commun. 13 (1) (2008).
[29] S.D. Galbraith, K.G. Paterson, N.P. Smart, Pairings for cryptographers, Discrete Appl. Math. 156 (16) (2008) 3113–3121, doi:10.1016/j.dam.2007.12.010.
[30] A. Joux, A new index calculus algorithm with complexity l(1/4+o(1)) in very small characteristic, IACR Cryptol. ePrint Arch. 2013 (2013) 95.
[31] J. Salowey, A. Choudhury, D. McGrew, AES Galois Counter Mode (GCM) Cipher Suites for TLS, 2008, (RFC 5288 (Proposed Standard)).
[32] K.G. Paterson, S. Srinivasan, Security and anonymity of identity-based encryption with multiple trusted authorities, in: S.D. Galbraith, K.G. Paterson (Eds.), Proceedings of the Pairing 2008, LNCS, vol. 5209, Springer, 2008, pp. 354–375.
[33] F. Beato, E.D. Cristofaro, K.B. Rasmussen, Undetectable communication: the online social networks case, in: Proceedings of the 2014 Twelfth Annual International Conference on Privacy, Security and Trust, Toronto, ON, Canada, July 23-24, 2014, pp. 19–26.
[34] M. Scott, Miracl–multiprecision integer and rational arithmetic c/c++ library, Shamus Software Ltd, Dublin, Ireland, URL (2003).
[35] A. Kate, I. Goldberg, Distributed key generation for the internet, in: Proceedings of the ICDCS, 2009, pp. 119–128.
[36] A. Huang, Distributed Key Generator, 2012, (https://crysp.uwaterloo.ca/software/DKG/).
[37] D. Boneh, A. Sahai, B. Waters, Fully collusion resistant traitor tracing with short ciphertexts and private keys, in: S. Vaudenay (Ed.), Proceedings of the EUROCRYPT 2006, LNCS, vol. 4004, Springer, 2006, pp. 573–592.
[38] J. Ugander, B. Karrer, L. Backstrom, C. Marlow, The anatomy of the facebook social graph, Clin. Orthop. Relat. Res. abs/1111.4503 (2011). Q3
[39] S. Jahid, P. Mittal, N. Borisov, ACM ASIACCS 2011, in: B.S.N. Cheung, L.C.K. Hui, R.S. Sandhu, D.S. Wong (Eds.), EASiER: encryption-based access control in social networks with efficient revocation, ACM, 2011, pp. 411–415.
[40] Y. Jung, Y. Nam, J. Kim, W. Jeon, H. Lee, D. Won, Key management scheme using dynamic identity-based broadcast encryption for social network services, in: H.Y. Jeong, M. S. Obaidat, N.Y. Yen, J.J.J.H. Park (Eds.), Proceedings of the CSA, LNEE, vol. 279, Springer Berlin Heidelberg, 2014, pp. 435–443.
[41] E.D. Cristofaro, M. Manulis, B. Poettering, Private discovery of common social contacts, Int. J. Inf. Sec. 12 (1) (2013) 49–65, doi:10.1007/s10207-012-0183-4.
[42] M. Nagy, E.D. Cristofaro, A. Dmitrienko, N. Asokan, A. Sadeghi, Do I know you?: efficient and privacy-preserving common friend-finder protocols and applications, in: C.N.P. Jr. (Ed.), Proceedings of the ACSAC 2013, ACM, 2013, pp. 159–168.

Please cite this article as: F. Beato et al., Practical identity-based private sharing for online social networks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.07.009