Practical Identity Based Broadcast Encryption for Online Social Networks

Stijn Meul, Filipe Beato, Bart Preneel and Vincent Rijmen

***Abstract*—Nowadays Online Social Networks (OSNs) constitute an important and useful communication channel. At the same time, coarse-grained privacy preferences protect the shared information insufficiently. Cryptographic techniques can provide interesting mechanism to protect privacy of users in OSNs. However, this approach faces several issues, such as, OSN provider acceptance, user adoption, key management and usability. We suggest a practical solution that uses Identity Based Encryption (IBE) to simplify key management and enforce confidentiality of data in OSNs. Moreover, we devise an outsider anonymous broadcast IBE scheme to disseminate information among multiple users, even if they are not using the system. Finally, we demonstrate the viability and tolerable overhead of our solution via an open-source prototype.**

***Index Terms*—identity-based encryption (IBE), broadcast encryption (BE), distributed key generation (DKG), online social networks (OSNs).**

I. Introduction

ONLINE SOCIAL NETWORKS (OSNS) such as Facebook, Google+, and Twitter are increasingly being used 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, it is possible to infer a considerable amount of sensitive information from the shared and stored content. Currently, users are allowed to configure "privacy preferences" in order to limit and select which users or groups can access the shared content. These preferences are generally too coarse-grained and difficult to configure [1]. Another problem is that these preferences do not exclude the provider along with the dangers of data leaks [2] nor external governments [3].

Stijn Meul
June 6, 2014

A. Problem Statement

All these worrisome issues motivate the need for effective techniques to properly protect user's privacy in OSNs. Several solutions have been proposed and advocated to use cryptographic mechanisms in order to address the privacy issues, either by an add-on atop of existing OSNs [4], [5], [6], [7], or by complete new privacy-friendly architectures [8], mainly decentralized [9], [10]. 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 [11]. 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 all interest to maintain their users and a privacy-friendly image. Hence, it is important to protect user's sharing information and the recipient set as it can contain private and sensitive information to the user.

B. Main Idea

Identity Based Encryption (IBE) [12] 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. Nevertheless, IBE-based systems require a trusted central Private Key Generator (PKG) server to generate the private parameters for each user based on a master secret. Consequently, such an architecture only shifts the trusted party from the OSN to the PKG. However, this problem can be mitigated if the master secret is divided among multiple PKGs following a Distributed Key Generation (DKG) [13] protocol based on Verifiable Secret Sharing (VSS) [14]. 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 at least t shares.

Many OSN users are not only represented on a single OSN but on several, thus, can also hold multiple public keys. Moreover, the multi-PKG setting could be supported and maintained by several existing OSNs. In particular, collaboration between OSN providers that compete along is assumed to be a difficult task and orthogonal to their economical business model. Figure 1 depicts an overview example of the proposed model, where a user authenticates to t-PKGs of his choice using, e.g., a similar token as in open id protocols, to retrieve his private key. This action can be performed after the reception of encrypted content as a consequence of user curiosity . The PKG servers can also be represented by governmental entities from different continents or subsidised research institutions, with no incentives to collaborate nor overcome more powerful adversaries using legal measures [15] among at least t-PKGs.

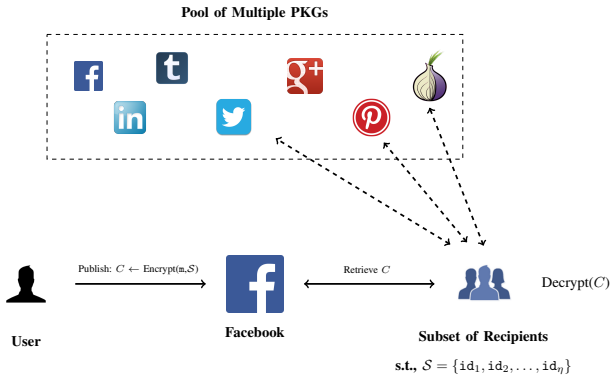

Fig. 1. Multiple (n, t)-PKG IBE for OSNs overview, for a message `m` published for the set $\mathcal{S}$ for $t = 3$.

C. Contribution:

In this paper, we propose a novel practical solution that uses IBE with multiple untrusted PKGs atop of current OSNs. We highlight the fact that those multi-PKGs can be supported by several existing OSNs under the business competition assumption, and motivated by the possible attractive incentives towards more privacy concerned audience. Along with the multi-PKG IBE model we devise an IBE broadcast encryption protocol to support multiple recipients. Using a broadcast IBE-based mechanism we allow users to share content with multiple recipients, even if they are not using the system, and, thus, enforce confidentiality of the data while hiding the recipient set. Finally, we implemented our solution on top of the Scramble Firefox extension [5], and show that only a small overhead is required.

D. Roadmap:

The remainder of this paper is organized as follows. Section II gives a brief overview of the cryptographic background. Next, Section III presents the model followed by the description of the suggested solution in Section IV. Section V describes the implementation details, while Section VI reviews related work. Finally, Section VII summarizes and concludes the paper.

II. 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.

A. Identity Based Encryption

The concept of Identity Based Encryption (IBE) was introduced by Shamir [12], 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 propose the first practical IBE using bilinear pairings [16], later extended by Gentry [17].

A generic IBE scheme is composed of four randomized algorithms:

`IBE.Setup`:
: On the input of a security parameter λ, outputs a master secret s and the master public parameters $params$.

`IBE.Extract`:
: Takes the public parameters $params$, the master secret s, and an `id` and returns the private key $d_{\mathtt{id}}$.

`IBE.Encrypt`:
: Returns the encryption C of the message `m` on the input of the $params$, the `id`, and the arbitrary length message `m`.

`IBE.Decrypt`:
: Reconstruct `m`from C by using the secret $d_{\mathtt{id}}$.

The `IBE.Setup` and `IBE.Extract` algorithms are executed by a trusted Private Key Generator (PKG) server, whereas `IBE.Encrypt` and `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.

B. Anonymous Broadcast Encryption

Broadcast encryption (BE) was introduced by Fiat and Naor [18], 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 bound to the ciphertext and the number of recipients. To overcome this issue, the set $\mathcal{S}$ of recipients is generally known. Barth et al. [19] and Libert et al. [20] 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 [21] suggested the notion of outsider anonymous BE that represents a more relaxed notion of ANOBE.

A generic BE and ANOBE scheme consists of four randomized algorithms:

`BE.Setup`:
: On the input of a security parameter λ, generates the public parameters $params$ of the system.

`BE.KeyGen`:
: Returns the public and private key (pk, sk) for each user according to the $params$.

`BE.Encrypt`:
: Takes the set $\mathcal{S} = \{pk_i \ldots pk_{|\mathcal{S}|}\}$ along with the secret message `m` and generates C.

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

Note that the pk can be represented by the `id` value from the IBE scheme.

C. Distributed Key Generation

Distributed Key Generation (DKG) was introduced by Pedersen [13] to allow a group of entities to collaboratively setup a secret sharing environment over a public channel. Secret sharing was introduced by Shamir [22] and consists of dividing a secret $\mathtt{s}$ into n shares among n entities, such that, only a subset of size greater than or equal to a threshold t can reconstruct $\mathtt{s}$, where $t \geq n$. In practice, a random secret $\mathtt{s}$ is generated along with a polynomial $f(x)$ of degree $t-1$ such that $f(0) = \mathtt{s}$, 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 $\mathtt{s}$. Further, Chor et al. [14] suggested a Verifiable Secret Sharing (VSS) scheme to allow anyone to verify that the right shares are used. The scheme was extended by Feldman [23] and Pedersen [13].

For multiple parties to jointly generate a secret sharing $\mathtt{s}$, all entities are required to participate in a DKG scheme. Each entity i involved generates a different $\mathtt{s}_i$ and $f^i(x)$, and later on distributes and verifies the shares s_{ij}. Hence, a generic DKG does not require a trusted party, as the master secret is computed as the sum of all the polynomials and can only be retrieved by joining t shares. A generic DKG protocol consists of two phases:

`DKG.Setup`:
: Every entity i generates a random secret $\mathtt{s}_i$ 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`:
: 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 parties or more.

III. Model

We consider that a user u to be a member of an OSN, such as Facebook, Twitter or Google+. Such u is connected with other users in the same OSN by a friendship relation with who shares information [24]. Inherently, u aims to interact and share information $\mathtt{m}$ with other users. Each user holds a public and private key pair which is given by an IBE identity server (composed of multiple PKG servers), such that the public key is represented by the `id` of the user in the OSN. Note that each user can be registered into multiple OSNs and hold different public keys and `ids`. We assume that the authentication between the user and the identity servers is done using a token similarly to open id, and performed under an authenticated channel, e.g., TLS.

a) Threat Model:: We consider an adversary to be any entity attempting to passively access the shared information $\mathtt{m}$ by monitoring the sharing channel but 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 an adversary should not learn the content nor the identity of the members in the recipient set $\mathcal{S}$, otherwise we consider that the adversary breaks confidentiality and the outsider recipient anonymity of the protocol as defined in [21]. In addition, we assume that such an adversary cannot compromise more than t identity servers. Furthermore, we stress that such adversary cannot control the user computing environment. Also, it is hard to protect against a malicious recipient who copies or forwards shared content. In this case, we say that such recipient breaks the social contract. We stress that we offer no protection against traffic analysis or timing attacks.

b) Goals:: We aim to protect OSN users' privacy by ensuring confidentiality, data integrity and outsider 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 prior knowledge from users in order to achieve a user-friendly scheme as defined by Balsa et al. [11]. In contrast to previous solutions, users are allowed to be in the recipient set by default.

IV. Practical IBE for OSNs

In this section, we describe our system. The proposed solution is based on the IBE scheme from Boneh et al. [16] and a relaxed version of the broadcast scheme from Libert et al. [20]. Further, the system relies on a DKG protocol as described by Pedersen [13] 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 [25] and because Type 1 pairings are no longer secure according Joux in [26].

A. 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 $\mathtt{id}_i$, 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 $e\left(aP, bQ\right) = e\left(P, Q\right)^{ab}$ for $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_q$ as in [16] . In addition, let $\{C, T\} \leftarrow \mathtt{E}_k(M)$ be any secure authenticated symmetric encryption that takes as input the plaintext M and generates ciphertext C and authentication tag T as output [27]. Similarly, $\{M, T\} \leftarrow \mathtt{D}_k(C)$ be the valid authenticated decryption that takes ciphertext C as input and computes the plaintext M along with an authentication tag T. Our scheme for OSNs is composed by five randomized algorithms: `Setup`, `KeyGen`, `Publish`, and `Retrieve`.

`Setup`(λ, t, n):
: Outputs the public $params$ of the system with respect to the security parameter λ, the number of PKGs n and the threshold t.

1) On input of security parameter λ generate a prime q, two groups G_1, G_2 of order q, and

an admissible bilinear map $e : G_1 \times G_2 \to G_T$. Choose random generators $P \in G_1$ and $Q \in G_2$.

2) Choose cryptographic hash functions $H_1 : \{0,1\}^* \to G_1$, $H_2 : G_T \to \{0,1\}^l$ and $H_3 : \{0,1\}^l \to \{0,1\}^l$ such that, H_1, H_2 can be modelled as random oracles.
3) Each PKG j generates $n-1$ shares σ_{jv} of a Pedersen VSS scheme by executing `DKG.Setup`, and redistributing the $n-1$ shares σ_{jv} with the other v PKGs.
4) Each PKG j publishes $P_{pub}^{(j)} = s_j P$, s.t., $s_j = \sum_{v=1}^{n} \sigma_{jv}$.

The master secret key $msk = \sum_{j \in \Lambda} b_j s_j$ for $b_j = \prod_{z \in \Lambda} \frac{z}{z-j}$ cannot be retrieved unless Λ is a subset of size t different PKG servers. The following parameters are published publicly:

$$params = \{q, G_1, G_2, e, P, Q, H_1, H_2, H_3, t, n, P_{pub}^{(0)}, \ldots, P_{pub}^{(n)}\}$$

`KeyGen(`$\{\texttt{PKG}_0, \ldots, \texttt{PKG}_t\}, \texttt{id}_i$`)`:

On input of a user $\texttt{id}_i$ the subset Λ of size t of PKG servers, generates a valid private key for $\texttt{id}_i$.

1) User with identifier $\texttt{id}_i$, authenticates to Λ or all PKGs and sends $\texttt{id}_i$.
2) Each PKG computes $Q_{\texttt{id}_i} = H_1(\texttt{id}_i)$, and $Q_{priv,\texttt{id}_i}^{(j)} = s_j Q_{\texttt{id}_i}$, where s_j is the secret share from PKG j.
3) The user $\texttt{id}_i$ computes the shared public parameter P using the Lagrange coefficients b_j as follows:
$$P = \sum_{j \in \Lambda} b_j P_{pub}^{(j)} \quad \text{for} \quad b_j = \prod_{z \in \Lambda} \frac{z}{z-j}$$
4) All PKGs in Λ return $Q_{priv,\texttt{id}_i}^{(j)}$ to the corresponding user $\texttt{id}_i$ over a secure channel.
5) Each user verifies for each $Q_{priv,\texttt{id}_i}^{(j)}$ value whether,
$$e\left(Q_{priv,\texttt{id}_i}^{(j)}, P\right) \stackrel{?}{=} e\left(Q_{\texttt{id}_i}, P_{pub}^{(j)}\right)$$
Next, $\texttt{id}_i$ calculates the private key $s_{\texttt{id}_i}$ using the Lagrange coefficients b_j as follows:
$$s_{\texttt{id}_i} = \sum_{j \in \Lambda} b_j Q_{priv,\texttt{id}_i}^{(j)} \quad \text{for} \quad b_j = \prod_{z \in \Lambda} \frac{z}{z-j}$$

In this way, no user or PKG learns the master key msk of the system. This algorithm combines `DKG.Reconstruct`, `IBE.Extract` and `BE.KeyGen` algorithms.

`Publish(`$params, \mathcal{S}, m$`)`:

Takes the message m, the subset $\mathcal{S}$ of size η and the public parameters $params$, output a broadcast message $\mathcal{B}$.

1) Generate a random symmetric session key $k \leftarrow \{0,1\}^l$.
2) Choose a random value $\rho \in \{0,1\}^l$ and compute r as a hash of concatenated values $r = H_3(\{\rho \parallel k\})$
3) For each recipient $\texttt{id}_i \in \mathcal{S}$, compute the ciphertext, running the `IBE.Encrypt` algorithm, as follows.
$$\begin{aligned} w_i = \quad & \rho \oplus H_2\left(g_{\texttt{id}_i}^r\right) \\ & \text{where} \quad g_{\texttt{id}_i} = e\left(Q_{\texttt{id}_i}, P_{pub}\right) \in G_T \end{aligned}$$
4) Let w be a randomised concatenation, then the authenticated data $\mathcal{A}$ is computed as
$$\begin{aligned} \mathcal{A} &= \{\eta \parallel rP \parallel k \oplus H_3(\rho) \parallel w_1 \parallel \ldots \parallel w_\eta\} \\ &= \{\eta \parallel U \parallel v \parallel w\} \\ &\quad \text{for} \quad w = \{w_1 \parallel w_2 \parallel \ldots \parallel w_\eta\} \end{aligned}$$
And $\mathcal{M}$ a concatenation of the intended recipient set $\mathcal{S}$ and the plaintext message m, such that $\mathcal{M} = \{m \parallel \mathcal{S}\}$. (`BE.Encrypt`)
5) Apply authenticated symmetric encryption
$$\langle c, t \rangle \leftarrow \mathtt{E}_k(\mathcal{M}, \mathcal{A})$$
6) The following message is then published in the OSN
$$\mathcal{B} = \{\mathcal{A} \parallel t \parallel c\}$$

`Retrieve(`$params, s_{\texttt{id}_i}, \mathcal{B}$`)`:

on input of the broadcast message $\mathcal{B}$ and the private key $s_{\texttt{id}_i}$ of user $\texttt{id}_i$, reconstruct the plaintext message m. This algorithm comprises the $\{\texttt{IBE},\texttt{BE}\}\texttt{.Decrypt}$ algorithms. For each $i \in \{\}$

1) Compute $w_i \oplus H_2(e(s_{\texttt{id}_i}, U)) = \rho$ for $s_{\texttt{id}_i}$, and $v \oplus H_3\{\rho\} = k$
2) Set $r = H_3(\rho, k)$. Verify $U \stackrel{?}{=} rP$. If the check fails, try next W_i and return to 1.
3) Retrieve $\langle \mathcal{M}, t' \rangle \leftarrow \mathtt{D}_k(c, \mathcal{A})$
4) Verify whether $t' \stackrel{?}{=} t \in \mathcal{B}$, and return m. Otherwise return $\perp$.

B. Evaluation

Our solution achieves confidentiality, integrity and outsider recipient anonymity as in [19], [16], [21],because the session key can only be obtain if the recipient holds the corresponding secret key $d_{\texttt{id}_i}$ and as a consequence of the authenticated encryption. Our solution can also be used in any OSN that assigns unique public `ids`, such as usernames. As the public keys are represented as strings users are not required to upload keys to an additional third party server. The DKG approach solves the key escrow issues that come with IBE solutions.

In terms of efficiency, users are required to decrypt W_i on average $O(n/2)$ before obtaining the symmetric key k. Both Barth et al. [19] and Libert et al. [20] propose using a tag based system to hint users where their symmetric key can be found. 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. 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. If the users opt not to reuse k they can still encrypt a fresh session key to all recipients in $\mathcal{S}$ using k.

In contrast to classic public key infrastructure, if a public key in IBE 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 [16].

For the multi-PKG setting, a user is able to detect malicious behavior from the public commitments of the Pedersen VSS [13]. It is also required that at least t from n PKGs do not get compromised, thus, the higher threshold t the higher the level of security. 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 is generated as a proof of identity by any of the OSN providers.

V. 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.[1] In this section, we discuss the implementation details and the performance results of the cryptographic blocks.

c) Implementation:: For the client component we modified the cryptographic library from Scramble [5] 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 [28]. 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 protocol we implemented a primitive version of Pedersen's DKG protocol [29] to generate the collective master secret key for the $(n, t)-$PKG servers. Adaption to the asynchronous setting could be done with the available implementation from Kate and Goldberg [30], [31]. AES-GCM [27] was used for the authenticated encryption. For the public key the Facebook username was used, i.e., $\texttt{id} = facebook.com/user.name$.

d) Performance:: Experiments were conducted on a Intel Core 2.4 GHz i5 processor with 8 Gb of 1600 MHz DDR3L onboard memory. Table II illustrates the execution times for the scheme proposed in Section IV 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 `m`. Note that efficiency comes at the cost of the recipient anonymity $\mathcal{S}$, as to hide $\mathcal{S}$ it is required to produce more `IBE.Encrypt` calls.

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

	Execution Time (ms)	
Number of Recipients	`Publish`	`Retrieve`
1	284.5	275.4
10	2564.5	460.9
15	3799.6	560.6
50	12300.5	1237.8
100	25867.7	2260.2

TABLE I
PERFORMANCE OF THE PUBLISH AND RETRIEVE STAGES IN FUNCTION OF THE NUMBER OF INTENDED RECIPIENTS.

	Execution Time (ms)	
IBE Stage	Boneh and Franklin	Gentry
`IBE.Setup`	368.10	424.49
`IBE.Extract`	13.84	37.46
`IBE.Encrypt`	271.90	1136.65
`IBE.Decrypt`	252.82	911.32

TABLE II
COMPARISON OF EXECUTION TIME FOR DIFFERENT IBE SCHEMES.

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; we show in Table **??** that there is a significant computational difference between a similar scheme in the standard model, i.e., Gentry [17]. Nevertheless, we believe that our solution presents a tolerable cost to average users with 100 friends and a usual group size of 15 [32].

VI. Related Work

The increased popularity of Online Social Networks (OSNs) and the amount of disseminated information prompted several privacy concerns. Guha et al. [6] proposed NOYB a solution that replaces the personal details of users by fake information. Later, FaceCloak [7] and Scramble [5] make use of cryptographic mechanisms to enforce privacy to the published information. Further, Persona suggest an attribute based encryption scheme for social networks. However, all the aforementioned solutions suffer from a complex key management infrastructure.

Other solutions take a more drastic approach by proposing novel, privacy-friendly architectures meant to replace existing platforms [8], [10], [9]. Besides the privacy protection offered, these solutions face a reduced user wilingness to adopt to a new platform.

Recently, Jung et al. [33] 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. In addition, by using 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.

VII. 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 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. Furthermore, the multiple PKG infrastructure can be maintained by several OSN providers, motivated by the attractive OSN privacy-friendly label, incentives towards more privacy concerned users, and considering the business model. Hence, 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. There are some important open challenges that call for further research. We endeavor to obtain a full open source project that supports different browsers. 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 to increase user privacy and security, as well as adoption of the scheme.

Acknowledgment

This paper is the result of a one-year masters thesis. Therefore I would like to thank my promotors prof. dr. ir. Bart Preneel and prof. dr. ir. Vincent Rijmen and the KU Leuven to make this research possible. I would also like to thank my daily supervisor Filipe Beato for helping me with the reviewing, writing and implementation.

References

[1] J. Bonneau and S. Preibusch, "The privacy jungle: On the market for data protection in social networks," *Economics of Information Security and Privacy*, pp. 121–167, 2010.

[2] M. Fischetti, "Data theft: Hackers attack," *Scientific American*, vol. 305, no. 100, 2011.

[3] W. Post, "NSA slides explain the PRISM data-collection program," June 6, 2013 http://wapo.st/J2gkLY. Accessed Sept. 6, 2013.

[4] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 135–146, Aug. 2009. [Online]. Available: http://doi.acm.org/10.1145/1594977.1592585

[5] F. Beato, M. Kohlweiss, and K. Wouters, "Scramble! your social network data," in *PETS*, ser. Lecture Notes in Computer Science, S. Fischer-Hübner and N. Hopper, Eds., vol. 6794. Springer, 2011, pp. 211–225.

[6] S. Guha, K. Tang, and P. Francis, "Noyb: privacy in online social networks," in *WOSN*. New York, NY, USA: ACM, 2008, pp. 49–54.

[7] W. Luo, Q. Xie, and U. Hengartner, "Facecloak: An architecture for user privacy on social networking sites," in *IEEE CSE*. Washington, DC, USA: IEEE, 2009, pp. 26–33.

[8] E. D. Cristofaro, C. Soriente, G. Tsudik, and A. Williams, "Hummingbird: Privacy at the time of twitter," in *IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 285–299.

[9] L. A. Cutillo, R. Molva, and M. Önen, "Safebook: A distributed privacy preserving online social network," in *WOWMOM*, 2011, pp. 1–3.

[10] J. Dwyer, "Four nerds and a cry to arms against Facebook," May 11, 2010. http://nyti.ms/1hc60kv. Accessed: Dec 3, 2013.

[11] E. Balsa, L. Brandimarte, A. Acquisti, C. Diaz, and S. F. G´urses, "Spiny CACTOS: OSN users attitudes and perceptions towards cryptographic access control tools," in *Workshop on Usable Security*, ser. Lecture Notes in Computer Science. San Diego,CA,USA: Springer-Verlag, 2014, p. 10.

[12] A. Shamir, "Identity-based cryptosystems and signature schemes," in *CRYPTO*, ser. Lecture Notes in Computer Science, G. R. Blakley and D. Chaum, Eds., vol. 196. Springer, 1984, pp. 47–53.

[13] 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*, ser. CRYPTO '91. London, UK, UK: Springer-Verlag, 1992, pp. 129–140. [Online]. Available: http://dl.acm.org/citation.cfm?id=646756.705507

[14] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract)," in *FOCS*, 1985, pp. 383–395.

[15] C. Matyszczyk, "If your account is subpoenaed, Facebook sends police, well, everything," http://preview.tinyurl.com/facebook-subpoena, 2012.

[16] D. Boneh and M. K. Franklin, "Identity based encryption from the Weil pairing," *IACR Cryptology ePrint Archive*, vol. 2001, p. 90, 2001.

[17] C. Gentry, "Practical identity-based encryption without random oracles," in *Advances in Cryptology - EUROCRYPT 2006*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed. Springer Berlin Heidelberg, 2006, vol. 4004, pp. 445–464.

[18] A. Fiat and M. Naor, "Broadcast encryption," in *CRYPTO*, ser. Lecture Notes in Computer Science, D. R. Stinson, Ed., vol. 773. Springer, 1993, pp. 480–491.

[19] A. Barth, D. Boneh, and B. Waters, "Privacy in encrypted content distribution using private broadcast encryption," in *Financial Cryptography*, ser. Lecture Notes in Computer Science, G. D. Crescenzo and A. D. Rubin, Eds., vol. 4107. Springer, 2006, pp. 52–64.

[20] B. Libert, K. G. Paterson, and E. A. Quaglia, "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model," in *Public Key Cryptography*, ser. Lecture Notes in Computer Science, M. Fischlin, J. Buchmann, and M. Manulis, Eds., vol. 7293. Springer, 2012, pp. 206–224.

[21] N. Fazio and I. M. Perera, "Outsider-anonymous broadcast encryption with sublinear ciphertexts," *IACR Cryptology ePrint Archive*, vol. 2012, p. 129, 2012.

[22] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[23] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, ser. SFCS '87. Washington, DC, USA: IEEE Computer Society, 1987, pp. 427–438. [Online]. Available: http://dx.doi.org/10.1109/SFCS.1987.4

[24] D. Boyd and N. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, 2008.

[25] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Appl. Math.*, vol. 156, no. 16, pp. 3113–3121, Sep. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.dam.2007.12.010

[26] A. Joux, "A new index calculus algorithm with complexity l(1/4+o(1)) in very small characteristic," *IACR Cryptology ePrint Archive*, vol. 2013, p. 95, 2013.

[27] J. Salowey, A. Choudhury, and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS," RFC 5288 (Proposed Standard), Internet Engineering Task Force, August 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5288.txt

[28] M. Scott, "Miracl–multiprecision integer and rational arithmetic c/c++ library," *Shamus Software Ltd, Dublin, Ireland, URL*, 2003.

[29] T. P. Pedersen, "A threshold cryptosystem without a trusted party (extended abstract)," in *EUROCRYPT*, 1991, pp. 522–526.

[30] A. Kate and I. Goldberg, "Distributed key generation for the internet," in *ICDCS*, 2009, pp. 119–128.

[31] A. Huang, "Distributed Key Generator," https://crysp.uwaterloo.ca/software/DKG/, 2012.
[32] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, "The anatomy of the facebook social graph," *CoRR*, vol. abs/1111.4503, 2011.
[33] Y. Jung, Y. Nam, J. Kim, W. Jeon, H. Lee, and D. Won, "Key management scheme using dynamic identity-based broadcast encryption for social network services," in *CSA*, ser. LNEE, H. Y. Jeong, M. S. Obaidat, N. Y. Yen, and J. J. J. H. Park, Eds., vol. 279. Springer Berlin Heidelberg, 2014, pp. 435–443.