at most t-1 are compromised

multi-PKGs there is no need to have a single trusted party, assuming that ~~at least~~ t of the PKGs are ~~not~~ 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.

References

1. R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. *SIGCOMM Comput. Commun. Rev.*, 39(4):135–146, Aug. 2009.
2. 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*, Lecture Notes in Computer Science, page 10, San Diego,CA,USA, 2014. Springer-Verlag.
3. A. Barth, D. Boneh, and B. Waters. Privacy in encrypted content distribution using private broadcast encryption. In G. D. Crescenzo and A. D. Rubin, editors, *Financial Cryptography*, volume 4107 of *Lecture Notes in Computer Science*, pages 52–64. Springer, 2006.
4. F. Beato, M. Kohlweiss, and K. Wouters. Scramble! your social network data. In S. Fischer-Hübner and N. Hopper, editors, *PETS*, volume 6794 of *Lecture Notes in Computer Science*, pages 211–225. Springer, 2011.
5. D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *IACR Cryptology ePrint Archive*, 2001:90, 2001.
6. D. Boyd and N. Ellison. Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*, 13(1), 2008.
7. B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *FOCS*, pages 383–395, 1985.
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*, pages 285–299. IEEE, 2012.
9. L. A. Cutillo, R. Molva, and M. Önen. Safebook: A distributed privacy preserving online social network. In *WOWMOM*, pages 1–3, 2011.
10. J. Dwyer. Four nerds and a cry to arms against Facebook. May 11, 2010. http://nyti.ms/1hc60kv. Accessed: Dec 3, 2013.
11. N. Fazio and I. M. Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. *IACR Cryptology ePrint Archive*, 2012:129, 2012.
12. P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, SFCS '87, pages 427–438, Washington, DC, USA, 1987. IEEE Computer Society.
13. A. Fiat and M. Naor. Broadcast encryption. In D. R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.
14. M. Fischetti. Data theft: Hackers attack. *Scientific American*, 305(100), 2011.
15. S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Appl. Math.*, 156(16):3113–3121, Sept. 2008.
16. C. Gentry. Practical identity-based encryption without random oracles. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer Berlin Heidelberg, 2006.
17. S. Guha, K. Tang, and P. Francis. Noyb: privacy in online social networks. In *WOSN*, pages 49–54, New York, NY, USA, 2008. ACM.
18. A. Joux. A new index calculus algorithm with complexity l(1/4+o(1)) in very small characteristic. *IACR Cryptology ePrint Archive*, 2013:95, 2013.
19. 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 H. Y. Jeong, M. S. Obaidat, N. Y. Yen, and J. J. H. Park, editors, *CSA*, volume 279 of *LNEE*, pages 435–443. Springer Berlin Heidelberg, 2014.
20. A. Kate and I. Goldberg. Distributed key generation for the internet. In *ICDCS*, pages 119–128, 2009.
21. B. Libert, K. G. Paterson, and E. A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 206–224. Springer, 2012.
22. W. Luo, Q. Xie, and U. Hengartner. Facecloak: An architecture for user privacy on social networking sites. In *IEEE CSE*, pages 26–33, Washington, DC, USA, 2009. IEEE.
23. C. Matyszczyk. If your account is subpoenaed, Facebook sends police, well, everything. http://preview.tinyurl.com/facebook-subpoena, 2012.
24. 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*, CRYPTO '91, pages 129–140, London, UK, UK, 1992. Springer-Verlag.
25. W. Post. NSA slides explain the PRISM data-collection program. June 6, 2013 http://wapo.st/J2gkLY. Accessed Sept. 6, 2013.
26. J. Salowey, A. Choudhury, and D. McGrew. AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288 (Proposed Standard), August 2008.
27. M. Scott. Miracl–multiprecision integer and rational arithmetic c/c++ library. *Shamus Software Ltd, Dublin, Ireland, URL*, 2003.
28. A. Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, 1979.
29. A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.
30. J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011.

2. Choose cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^l$, $H_3 : \{0,1\}^l \rightarrow \{0,1\}^l$ and $H_4 : \{0,1\}^l \rightarrow \{0,1\}^l$, such that, H_1, H_2 can be modeled as random oracles.
3. Each PKG j generates $n-1$ shares s_{jv} of a Pedersen VSS scheme by executing `DKG.Setup`, and redistributes the $n-1$ shares with the other v PKGs.
4. Each PKG j publishes $P_{pub}^{(j)} = s_j P$, s.t., $s_j = \sum_{v=1}^{n} s_{jv}$.

The master secret key cannot be retrieved unless t out of n PKGs would collude. The following parameters are published publicly:

$$params = \{q, \mathbb{G}_1, \mathbb{G}_2, e, P, Q, H_1, H_2, H_3, H_4, t, n, P_{pub}^{(0)}, \ldots, P_{pub}^{(n)}\}$$

`KeyGen`($\{\mathtt{PKG}_0, \ldots, \mathtt{PKG}_t\}$, $\mathtt{id}_i$): On the input of a user $\mathtt{id}_i$ the sub set A of size t of PKG servers, generate a a valid private key for $\mathtt{id}_i$.
1. User with $\mathtt{id}_i$, authenticates to A or all PKGs and sends $\mathtt{id}_i$.
2. Each PKG computes $Q_{\mathtt{id}_i} = H_1(\mathtt{id}_i)$, and $Q_{priv,\mathtt{id}_i}^{(j)} = s_j Q_{\mathtt{id}_i}$, where s_j is the secret share from PKG j.
3. All PKGs return $Q_{priv,\mathtt{id}_i}^{(j)}$ to the corresponding user u_i over a secure channel.
4. Each user verifies for each $Q_{priv,\mathtt{id}_i}^{(j)}$ value whether,

$$e\left(Q_{priv,\mathtt{id}_i}^{(j)}, P\right) \stackrel{?}{=} e\left(Q_{\mathtt{id}_i}, P_{pub}^{(j)}\right)$$

If the check fails, report that PKG as malicious and request another PKG. Next, u_i calculates the private key $d_{\mathtt{id}_i}$ using the Lagrange coefficients a_j as follows:

$$d_{\mathtt{id}_i} = \sum_{j \in A} a_j Q_{priv,\mathtt{id}_i}^{(j)} \quad \text{for} \quad a_j = \prod_{z \in A} \frac{z}{z-j}$$

In this way, no user learns the master key $\mathtt{s}$ of the system. This algorithm combines `DKG.Reconstruct`, `IBE.Extract` and `BE.KeyGen` algorithms.

`Publish`($params, \mathcal{S}$, `m`): Take the message `m`, the subset $\mathcal{S}$ of size η and the public $params$, output an encrypted message $\mathcal{B}$.
1. Generate a random symmetric session key $k \leftarrow \{0,1\}^l$.
2. Choose a random value $\sigma \in \{0,1\}^l$ and set $r = H_3(\sigma, k)$
3. For each recipient $u_i \in \mathcal{S}$, compute the ciphertext, running the `IBE.Encrypt` algorithm, as follows.

$$W_i = \sigma \oplus H_2(r g_{\mathtt{id}_i}) \quad \text{where} \quad g_{\mathtt{id}_i} = e(Q_{\mathtt{id}_i}, P_{pub}) \in \mathbb{G}_T$$

4. Let A be a randomized concatenation as follows

$$\begin{aligned} A &= \{\eta \,\|\, rP \,\|\, k \oplus H_4(\sigma) \,\|\, W_1 \,\|\, W_2 \,\|\, \ldots \,\|\, W_\eta\} \\ &= \{\eta \,\|\, U \,\|\, V \,\|\, W\} \quad \text{for} \quad W = \{W_1 \,\|\, W_2 \,\|\, \ldots \,\|\, W_\eta\} \end{aligned}$$

And M a concatenation of the intended recipient set $\mathcal{S}$ and the plaintext message `m`, such that $M = \{\mathtt{m} \,\|\, \mathcal{S}\}$. (`BE.Encrypt`)

5. Apply authenticated symmetric encryption

$$\{C, T\} \leftarrow \mathrm{E}_k(M, A)$$

6. The following message is then published in the OSN

$$\mathcal{B} = \{A \,\|\, T \,\|\, C\}$$

`Retrieve`($params, d_{\mathtt{id}_i}, \mathcal{B}$): on the input of the encrypted message and the private key, reconstruct the secret message `m`. This algorithm comprises the `{IBE,BE}.Decrypt` algorithms.

1. Compute $W_i \oplus H_2(e(d_{\mathtt{id}_i}, U)) = \sigma$ for $d_{\mathtt{id}_i}$, and $V \oplus H_4\{\sigma\} = k$
2. Set $r = H_3(\sigma, k)$.
3. Retrieve $\{M, T'\} \leftarrow \mathrm{D}_k(C, A)$
4. Verify whether $T' \stackrel{?}{=} T \in \mathcal{B}$, and return `m`. Otherwise return $\perp$.

4.2 Evaluation

Our solution achieves confidentiality, integrity and outsider recipient anonymity as in [3, 5, 11]. It 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. ~~While~~ 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. [3] and Libert et al. [21] 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. Where the current scheme 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 as the recipient set $\mathcal{S}$.

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 [5].

~~While~~ for the multi-PKG setting, a user is able to detect malicious behavior from the public commitments of the Pedersen VSS [24]. 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. Furthermore, authentication and identity verification to the different servers can be done via, for instance, an open id token.

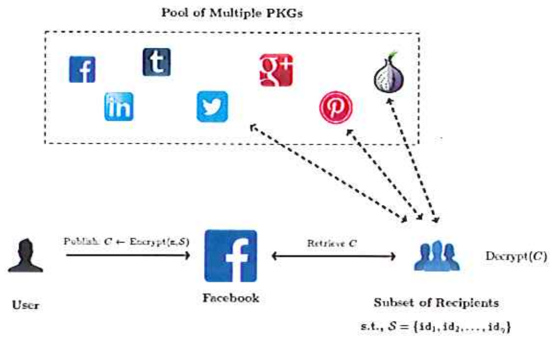

Fig. 1. Multiple (n, t)-PKG IBE for OSNs overview, for a message $\mathtt{m}$ published for the set $\mathcal{S}$ for $t = 3$.

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 [29], 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 [5], later extended by Gentry [16].

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 $\mathtt{m}$ on the input of the *params*, the `id`, and the arbitrary length message $\mathtt{m}$.

`IBE.Decrypt`: Reconstruct $\mathtt{m}$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.

2.2 Anonymous Broadcast Encryption

Broadcast encryption (BE) was introduced by Fiat and Naor [13], 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. [3] and Libert et al. [21] 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 e Perera [11] suggested the notion of outsider anonymous 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 $\mathcal{S} = \{pk_i \dots pk_{|\mathcal{S}|}\}$ along with the secret message $\mathtt{m}$and generates C.

`BE.Decrypt`: Reconstructs $\mathtt{m}$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.

2.3 Distributed Key Generation

Distributed Key Generation (DKG) was introduced by Pedersen [24] to allow a group of entities to collaboratively setup a secret sharing environment over a public channel. Secret sharing was introduced by Shamir [28] 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