SCIENCE CHINA
Information Sciences

• LETTER •

March 2020, Vol. 63 139110:1–139110:3
https://doi.org/10.1007/s11432-018-9455-5

Space efficient revocable IBE for mobile devices in cloud computing

Baodong QIN[1,2*], Ximeng LIU[3], Zhuo WEI[4] & Dong ZHENG[1,5*]

[1] *National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an* 710121*, China;*
[2] *State Key Laboratory of Cryptology, P.O. Box* 5159*, Beijing* 100878*, China;*
[3] *School of Information Systems, Singapore Management University, Singapore* 178902*, Singapore;*
[4] *Huawei Singapore Research Center, Singapore* 117674*, Singapore;*
[5] *Westone Cryptologic Research Center, Beijing* 100070*, China*

Received 15 March 2018/Accepted 17 May 2018/Published online 11 February 2020

Citation Qin B D, Liu X M, Zhuo W, et al. Space efficient revocable IBE for mobile devices in cloud computing. Sci China Inf Sci, 2020, 63(3): 139110, https://doi.org/10.1007/s11432-018-9455-5

Dear editor,

Revocation capacity is one of the main properties for an identity-based encryption (IBE), as in practice users' private keys are possibly leaked or expired. However, existing revocable IBE schemes [1–3] usually lack of short keys. Recently, Lin et al. [4] proposed a method to design space efficient revocable IBE scheme from non-monotonic key-policy attribute-based encryption scheme. But, it requires too many pairings (linear to the number of revoked users) to decrypt an IBE ciphertext. In this study, we overcome this problem by adopting the technique of server-aided revocation, recently proposed by Qin et al. [5] in ESORICS 2015. The main contribution is a new server-aided revocable IBE scheme, which can largely shift decryption overhead from local users to an untrusted cloud computing server, and significantly reduce the complexities of PKG's (Private-Key Generator) key update information and server's long-term identity-based public information (also called long-term transformation keys) with the comparison of previous (server-aided) revocable IBE schemes.

The involved cloud server stores users' transformation keys and does ciphertext transformation for local users. The local user only needs two pairings to decrypt a transformed ciphertext. The PKG first distributes a long-term private key to each user via a secure communication channel, and a long-term (identity-based) transformation key to the server via a public communication channel. For each time period, the PKG publicly distributes a time-based key update information to the server, so that the server can help any non-revoked user to generate a short-term transformation key and use this key to translate his/her ciphertexts into efficiently decryptable ciphertexts. In other words, only the non-revoked user can efficiently recover messages from the transformed ciphertexts using his long-term private key. Moreover, the long-term private key supports delegation to other parties for a specified time period. If one period decryption key is compromised, it does not affect the security of other period decryption keys. So, our scheme is secure against the decryption key exposure attacks introduced by Seo and Emura [1]. The formal definitions of server-aided revocable IBE and its selective-ID security against chosen-plaintext attacks (sID-CPA security) are given in Appendix A.

Intuition of our scheme. Our construction benefits from the non-monotonic access structure of a key-policy attribute-based encryption (KP-ABE) scheme and the homomorphic key generation algorithm of the KP-ABE scheme that have already been used to construct standard revocable IBE scheme by Lin et al. [4]. In particular, we use Ostrovsky, Sahai and Waters non-monotonic KP-

[*] Corresponding author (email: qinbaodong@xupt.edu.cn, zhengdong_xupt@sina.com)

ABE scheme (shorted as OSW scheme) [6] as a basic tool. In our scheme, a master secret key msk is split into two parts such that $\mathrm{msk} = \alpha_1+\alpha_2$. For simplicity, we denote by $\mathsf{OSW}[\alpha_i]$ the key generation algorithm of the OSW scheme with master secret key α_i. For each user id, the long-term private key $\mathrm{sk}_{\mathrm{id}}$ is generated via $\mathsf{OSW}[\alpha_2]$ for predicate id. There is no long-term transformation key in this scheme. For each time period t, the PKG generates a key update information ku_t via $\mathsf{OSW}[\alpha_1]$ for predicate $t \wedge_{\omega\in \mathrm{rl}} \neg\omega$, where rl is the set of identities of revoked users. The key update is just the (time-based) short-term transformation key. Essentially, the short-term transformation key together with the user's long-term private key can be viewed as a decryption key generated via $\mathsf{OSW}[\alpha_1 + \alpha_2]$ for predicate $\mathrm{id} \wedge t \wedge_{\omega\in\mathrm{rl}} \neg\omega$. In our encryption algorithm, a message is encrypted under the attribute set $\{\mathrm{id}, t\}$. So, by the security of the underlying ABE scheme, the user can correctly recover the message if and only if his identity id is not in the revocation list rl at time period t. To support decryption key delegation, a user can convert his long-term private key for predicate id into a key for a tight predicate $\mathrm{id} \wedge t$.

The construction. Let $\mathbb{G}$ be a bilinear group of prime order p, and let g be a random generator of $\mathbb{G}$. We denote by $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ the bilinear map from $\mathbb{G}$ to some finite group $\mathbb{G}_T$.

- $\mathsf{Setup}(1^\kappa) \to (\mathrm{sp}, \mathrm{msk}, \mathrm{rl}, \mathrm{st})$: The setup algorithm chooses random terms α_1, α_2, $\beta \xleftarrow{\$} \mathbb{Z}_p$, and sets $\alpha = \alpha_1+\alpha_2 \pmod p$, $g_1 = g^\alpha$ and $g_2 = g^\beta$. In addition, it chooses two random polynomials $h(x)$ and $q(x)$ of degree 2 subject to the constraint that $q(0) = \beta$. It also defines two publicly computable functions $T, V : \mathbb{Z}_p \to \mathbb{G}$, where $T(x)$ maps $x \in \mathbb{Z}_p$ to $g_2^{x^2} \cdot g^{h(x)}$, and $V(x)$ maps $x \in \mathbb{Z}_p$ to $g^{q(x)}$. The revocation list rl is initialized as an empty set $\emptyset$. It outputs $\mathrm{rl} = \emptyset$ as well as $\mathrm{sp} = (g, g_1, g_2 = g^{q(0)}, g^{q(1)}, g^{q(2)}, g^{h(0)}, g^{h(1)}, g^{h(2)}, T(x), V(x))$ and $\mathrm{msk} = (\alpha_1, \alpha_2)$.
- $\mathsf{PubKG}(\mathrm{msk}, \mathrm{id}) \to \mathrm{pk}_{\mathrm{id}}$: The long-term transformation key generation algorithm directly returns $\mathrm{pk}_{\mathrm{id}} = \emptyset$, i.e., for each user, no predistributed long-term transformation key is required.
- $\mathsf{TranKU}(\mathrm{msk}, t, \mathrm{rl}, \mathrm{st}) \to \mathrm{ku}_t$: The transformation key update algorithm outputs a transformation key update, which is computed for a special access structure, namely, $t \wedge_{\omega\in\mathrm{rl}} \neg\omega$ and sends to the server. For time period $t \in \mathbb{Z}_p^*$, it chooses random value $r_t \in \mathbb{Z}_p$ and for each revoked user $\omega \in \mathrm{rl}$, it chooses random values $\lambda_\omega, r_\omega \in \mathbb{Z}_p$. Next, it computes $E_t = (E_t^{(1)}, E_t^{(2)}) = (g_2^{\alpha_1-\sum_{\omega\in\mathrm{rl}}\lambda_\omega} T(t)^{r_t}, g^{r_t})$ and $E_{\neg\omega} = (E_\omega^{(3)}, E_\omega^{(4)}, E_\omega^{(5)}) = (g_2^{\lambda_\omega+r_\omega}, V(\omega)^{r_\omega}, g^{r_\omega})$. Finally, the algorithm outputs a transformation key update $\mathrm{ku}_t = (E_t, \{E_{\neg\omega}\}_{\omega\in\mathrm{rl}})$.
- $\mathsf{TranKG}(\mathrm{id}, t, \mathrm{pk}_{\mathrm{id}}, \mathrm{ku}_t) \to \mathrm{tk}_{\mathrm{id},t}$: The short-term transformation key generation algorithm sets $\mathrm{tk}_{\mathrm{id},t} := \mathrm{ku}_t$ and returns the (time-based) short-term transformation key $\mathrm{tk}_{\mathrm{id},t}$. In other words, the key update ku_t is just the short-term transformation key for all users at time period t.
- $\mathsf{PrivKG}(\mathrm{msk}, \mathrm{id})$: The long-term private key generation algorithm outputs a long-term private key for user id. It chooses a random value $s_{\mathrm{id}} \in \mathbb{Z}_p$, and outputs $\mathrm{sk}_{\mathrm{id}} = (D_{\mathrm{id}}^{(1)}, D_{\mathrm{id}}^{(2)}) = (g_2^{\alpha_2} T(\mathrm{id})^{s_{\mathrm{id}}}, g^{s_{\mathrm{id}}})$.
- $\mathsf{DecKG}(\mathrm{sk}_{\mathrm{id}}, t)$: This algorithm generates a short-term decryption key for time period t. Given a long-term private key $\mathrm{sk}_{\mathrm{id}} = (D_{\mathrm{id}}^{(1)}, D_{\mathrm{id}}^{(2)})$, it chooses random terms $r_t, s'_{\mathrm{id}} \in \mathbb{Z}_p$, and outputs $\mathrm{dk}_{\mathrm{id},t} = (D_{\mathrm{id}}^{(1)} \cdot T(\mathrm{id})^{s'_{\mathrm{id}}} T(t)^{r_t}, D_{\mathrm{id}}^{(2)} \cdot g^{s'_{\mathrm{id}}}, g^{r_t})$.
- $\mathsf{Encrypt}(\mathrm{mpk}, \mathrm{id}, t, M)$: To encrypt a message $M \in \mathbb{G}_T$ under identity id and time period t, it chooses a random value $s \in \mathbb{Z}_p$ and outputs the following ciphertext $C_{\mathrm{id},t} = (C^{(1)}, C^{(2)}, \{C_x^{(3)}, C_x^{(4)}\}_{x\in\{\mathrm{id},t\}}) = (Me(g_1, g_2)^s, g^s, \{T(x)^s, V(x)^s\}_{x\in\{\mathrm{id},t\}})$.
- $\mathsf{Transform}(\mathrm{tk}_{\mathrm{id},t}, C_{\mathrm{id},t})$: This algorithm is run by the server. It computes a partially decrypted ciphertext $C'_{\mathrm{id},t}$ as follows. For each revoked user $\omega \in \mathrm{rl}$, we consider the set $S_\omega = \{\mathrm{id}, t, \omega\}$ which has three distinct values. Using the points in S_ω as an interpolation, it computes Lagrangian coefficients $\{\sigma_x\}_{x\in S_\omega}$ so that $\sum_{x\in S_\omega} \sigma_x \cdot q(x) = q(0) = \beta$. Then, the algorithm computes $H_t = e(C^{(2)}, E_t^{(1)})/e(C_t^{(3)}, E_t^{(2)})$. For every $\omega \in \mathrm{rl}$, it computes $H_{\neg\omega} = \frac{e(C^{(2)}, E_\omega^{(3)})}{e(\prod_{x\in\{\mathrm{id},t\}} (C_x^{(4)})^{\sigma_x}, E_\omega^{(5)}) \cdot e(C^{(2)}, E_\omega^{(4)})^{\sigma_\omega}}$. Now, the algorithm computes $H = H_t \cdot \prod_{\omega\in\mathrm{rl}} H_{\neg\omega}$, sets $C'^{(1)} = C^{(1)}/H$ and outputs $C'_{\mathrm{id},t} = (C'^{(1)}, C^{(2)}, \{C_x^{(3)}, C_x^{(4)}\}_{x\in\{\mathrm{id},t\}})$.
- $\mathsf{Decrypt}(\mathrm{dk}_{\mathrm{id},t}, C'_{\mathrm{id},t})$: This algorithm recovers the message M from the transformed ciphertext $C'_{\mathrm{id},t}$, using the decryption key $\mathrm{dk}_{\mathrm{id},t}$. It computes

$$H' = \frac{e(C^{(2)}, D_{\mathrm{id},t}^{(1)})}{e(C_{\mathrm{id}}^{(3)}, D_{\mathrm{id},t}^{(2)}) \cdot e(C_t^{(3)}, D_{\mathrm{id},t}^{(3)})}.$$

Finally, the message is obtained by computing $M = C'^{(1)}/H'$.

- $\mathsf{Revoke}(\omega, \mathrm{rl})$: To revoke identity ω, the revocation algorithm adds ω into rl and outputs an updated revocation list.

Theorem 1. If an adversary can break our scheme with advantage ϵ in the sID-CPA security

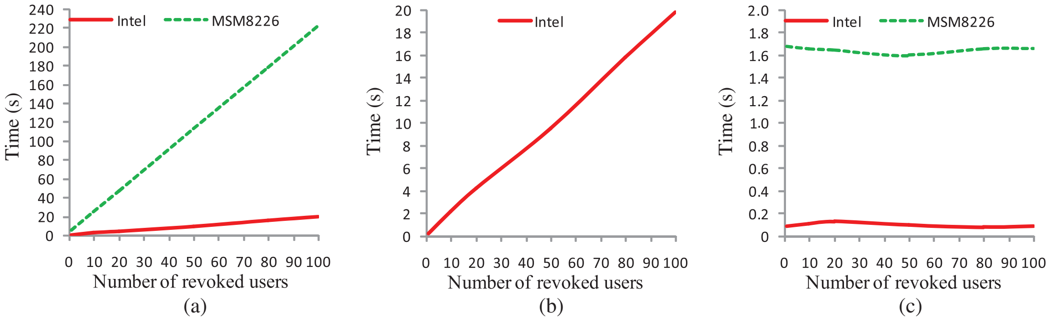

Figure 1 (Color online) Performance of our server-aided revocable IBE scheme. (a) Fulldecryption time; (b) transform time; (c) final decryption time.

model, a simulator can be constructed to break the security of the OSW KP-ABE scheme with advantage $\epsilon/2$.

The proof of Theorem 1 is presented in Appendix B.

Experimental evaluation. To evaluate the performance of our main scheme, we implement it in software based on the jPBC library [7] and using a 512-bit Type-A pairing. The pairing is constructed on the curve $y^2 = x^3 + x$ over the filed $\mathbb{F}_q$ for some 512-bit prime q and the order of the group $\mathbb{G}$ is some 160-bit prime q. We use a laptop to simulate the server environment and use a mobile phone to simulate the user environment. The hardware platforms are as follows: a 2.5 GHz Intel Core i5 CPU with 6 GB RAM running 64-bit Windows 10, and a 1.2 GHz MSM8226-based mobile phone with 2 GB of RAM running Android 4.3 OS.

As expected, the full decryption time of space efficient revocable IBE schemes [4] is linear in the number of revoked users, while the local decryption time of server-aided revocation is almost constant and independent of the number of revoked users, as shown in Figure 1. For example, for time period with 100 number of revoked users, fully decrypting a ciphertext needs more than 220 s (nearly 4 min), while the final decryption requires less than 1.8 s on the MSM8226 processor. Even for time period with few number of revoked users, the required decryption time seems to be unacceptable on computing resource constrained devices.

Conclusion. This study proposed a space efficient server-aided revocable IBE scheme that has faster decryption than that of Lin et al.'s non-server-aided space efficient revocable IBE scheme. In addition, the scheme removes the requirement of periodical communication between the private key generator and users during key updates, and supports a new security property, namely decryption key exposure resistance.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61502400, 61702105), Foundation of Sichuan Educational Committee (Grant No. 16ZB0140), and National Joint Fund of China (Grant No. U1636209).

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Seo J H, Emura K. Revocable identity-based encryption revisited: security model and construction. In: Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, 2013. 216–234

2 Boldyreva A, Goyal V, Kumar V. Identity-based encryption with efficient revocation. In: Proceedings of the 2008 ACM Conference on Computer and Communications Security, Alexandria, 2008. 417–426

3 Watanabe Y, Emura K, Seo J H. New revocable IBE in prime-order groups: adaptively secure, decryption key exposure resistant, and with short public parameters. In: Proceedings of The Cryptographers' Track at the RSA Conference, San Francisco, 2017. 432–449

4 Lin H, Cao Z, Fang Y, et al. How to design space efficient revocable IBE from non-monotonic ABE. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, Hong Kong, 2011. 381–385

5 Qin B D, Deng R H, Li Y J, et al. Server-aided revocable identity-based encryption. In: Proceedings of the 20th European Symposium on Research in Computer Security, Vienna, 2015. 286–304

6 Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 2007 ACM Conference on Computer and Communications Security, Alexandria, 2007. 195–203

7 de Caro A, Iovino V. jpbc: Java pairing based cryptography. In: Proceedings of the 16th IEEE Symposium on Computers and Communications, Kerkyra, 2011. 850–855