SCIENCE CHINA
Information Sciences

• LETTER •

October 2020, Vol. 63 209302:1–209302:3
https://doi.org/10.1007/s11432-019-2793-y

Revocable and certificateless public auditing for cloud storage

Yinghui ZHANG[1,2,3*], Tiantian ZHANG[1,3], Shengmin XU[4], Guowen XU[5] & Dong ZHENG[1,2,3]

[1]*School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an* 710121*, China;*
[2]*Guangxi Cooperative Innovation Center of Cloud Computing and Big Data, Guilin University of Electronic Technology, Guilin* 541004*, China;*
[3]*National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an* 710121*, China;*
[4]*Secure Mobile Centre, School of Information Systems, Singapore Management University, Singapore* 178902*, Singapore;*
[5]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu* 611731*, China*

Received 9 November 2019/Revised 20 January 2020/Accepted 4 February 2020/Published online 15 July 2020

Citation Zhang Y H, Zhang T T, Xu S M, et al. Revocable and certificateless public auditing for cloud storage. Sci China Inf Sci, 2020, 63(10): 209302, https://doi.org/10.1007/s11432-019-2793-y

Dear editor,

Plenty of computing and storage resources in the cloud are provided for users with restricted computing and storage resources, which has attracted the attention of many researchers [1, 2]. A generic blockchain-based cloud data auditing scheme [3] is proposed, which is compatible with any blockchains including the bitcoin blockchain. In the data integrity checking scheme, certificateless signature (CLS) can be used to verify the identity of users. Besides, the key exchange is utilized in the key generation [2], which can eliminate the security channel to achieve system robustness. Considering the real situation, the users who join the cloud storage system may be revoked for some reasons. Therefore after a user is revoked, it is necessary to detect the validity of the tags of the revoked user and update the tags on manner. The group manager (GM) of scheme [4] updates the key of the non-revoked user with the number of revoked users. In addition, some improvements have been proposed to support tag updates. Li et al. [5] adopted an interactive method to update the tags of the revoked user, which aims at reducing the computing overhead of the user. However, there is a risk of signature forgery in their scheme because the secret key and the signature message are not bound in the signature process.

We put forward a certificateless public data integrity detection scheme that supports user revocation in cloud storage. Our scheme enables the key update of the group user and the tag update of the revoked user. Besides, the user's private key used for signing is bound with the signature message, which is secure against tag forgery attacks. The idea of key exchange is exploited to enhance the robustness of the system during the key generation process, which enables the GM to interact with the user through public channels. In addition, according to the number of revoked users, the GM updates the partial key for the non-revoked user who then generates a new full key by utilizing the updated partial key. Because the cloud service provider (CSP) adopts the non-revoked user's latest public key and the number of revoked users to inspect the identity of the user, it is ensured that the revoked user without the latest key cannot pass authentication. The proposed scheme can resist chosen-message attacks.

Model. The system model mainly consists of four entities: the group users, the GM, the CSP and the third-party auditor (TPA). During the

* Corresponding author (email: yhzhaang@163.com)

data integrity check, challenge information, which is randomly generated by the TPA, is sent to the CSP. The proof information of corresponding blocks challenged, which is generated by the CSP, is sent to the TPA. Finally, by using the parameters of open log files and pseudo-random functions, the TPA performs verification operations for the validity of the proof. The CSP is semi-honest and may cheat the TPA to think that the data stored on the CSP has integrity and it undamaged for additional benefits.

Our scheme. We specifically describe our scheme, including eleven algorithms. We define that each data block of a file is an element in Z_q^* and num is the number of users in a group.

• Setup. Let G_1 and G_2 be two multiplicative cyclic groups of prime order q. There exists a bilinear map $\hat{e} : G_1 \times G_1 \to G_2$. g is a generator of G_1. $H_1 : \{0,1\}^{ID_l} \times G_1 \times G_1 \times \{0,1\}^* \to Z_q^*$, $H_2 : G_1 \to Z_q^*$, $H_3 : \{0,1\}^* \to G_1$, $H_4 : Z_q^* \to \{0,1\}^*$ and $H_5 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to G_1$ are five secure hash functions. ID_l is defined as the bit length of user identity. $f : Z_q^* \times Z_q^* \to Z_q^*$ is a pseudo-random function. The GM randomly chooses $s \in Z_q^*$ and computes $P = g^s$. Let s be the master secret key and P be the corresponding public key. It keeps s private and publishes the system parameters $params = (q, g, G_1, G_2, \hat{e}, P, H_1, H_2, H_3, H_4, H_5, f)$.

• SecretValueGen. A user u_i with the identity ID_i, $1 \leqslant i \leqslant$ num, first selects $x_i \in Z_q^*$ and computes $X_i = g^{x_i}$. Then u_i sets $usk_i = x_i$ and $upk_i = X_i$ and sends upk_i to GM.

• PartialKeyGen. After receiving upk_i, the GM randomly selects $r_i \in Z_q^*$ and computes $R_i = g^{r_i}$, $h_1 = H_1(ID_i||X_i||R_i||RN)$ and $k_i = r_i + sh_1 + H_2(X_i^s)$. Here RN is the number of user revocation. In general, the GM sets the initial value of RN to 0 and publishes it to the CSP and group users. Then the GM sends $psk_i = k_i$ and $ppk_i = R_i$ to the user u_i through public channels.

• FullUserKeyGen. After receiving psk_i and ppk_i, the user u_i firstly computes $d_i = k_i - H_2(P^{x_i})$ and $h_1 = H_1(ID_i||X_i||R_i||RN)$. then checks whether the equation $g^{di} = R_i \cdot P^{h_1}$ holds or not. The user u_i accepts psk_i and ppk_i if and only if the above equation holds. Otherwise, it refuses them and applies new psk_i and ppk_i again. Then the user u_i computes $D_i = H_3(ID_i)^{d_i}$. Finally the user u_i sets $sk_i = (x_i, D_i)$, $pk_i = (X_i, R_i)$.

• TagGen. The user u_i computes data-block tags of file M, where $M = \{m_1, m_2, \ldots, m_n\}$. Firstly the user u_i computes file identity $FID = H_4(M)$ and tag for each data block $\sigma_j = D_i^{m_j} \cdot H_5([FID]||j||n)^{x_i}$, $1 \leqslant j \leqslant n$. Let $\sigma = \{\sigma_j\}_{1 \leqslant j \leqslant n}$. The GM keeps a public log file and saves $H_3(ID_i)$, pk_i, h_1, n and FID in the public log file. Finally, the user u_i uploads (M, σ) to the CSP.

• Upload. After receiving the messages from the user u_i, the CSP first calculates $\overline{FID} = H_4(M)$ and $h_1 = H_1(ID_i||X_i||R_i||RN)$. Then it verifies the validity of tags and legitimacy of user identity by checking whether the equation $\hat{e}(\sigma_j, g) = \hat{e}(H_3(ID_i)^{m_j}, R_i \cdot P^{h_1}) \cdot \hat{e}(H_5(\overline{FID}||j||n), X_i)$ holds or not. The CSP considers that these tags are valid when the equation holds. Otherwise, it considers that these tags are generated by an illegal user or a revoked user.

• Challenge. The TPA first retrieves the public log file to gain $H_3(ID_i)$, pk_i, h_1, n and FID. Then it selects a c-element subset J, where $J \subseteq [1, n]$, and $k_1 \in Z_q^*$ as the seed for the pseudo-random function f. Then it sends challenge message $chal = \{c, k_1\}$ to the CSP.

• ProofGen. The CSP computes $v_j = f(k_1, j)$ for each $j \in J$. Then it computes $\lambda = \sum_{j \in J} v_j m_j$ and $T = \prod_{j \in J} \sigma_j^{v_j}$. Then it sets $proof = (\lambda, T)$ as the proof and returns it to the TPA.

• Verify. The TPA calculates $v_j = f(k_1, j)$ for each $j \in J$ receipt of the proof information from the CSP. Then $proof$ can be verified as below: $\hat{e}(T, g) = \hat{e}(H_3(ID_i)^{\lambda}, R_i \cdot P^{h_1}) \cdot \hat{e}(\prod_{j \in J} H_5(FID||j||n)^{v_j}, X_i)$.

• RevTagUpdate. u_r is defined as a revoked user and u_k is defined as a non-revoked user, where $1 \leqslant r, k \leqslant n$, $k \neq r$. We suppose that there is no collusion among u_r, u_k and the CSP. Firstly, the CSP randomly selects $\rho \in Z_q^*$, computes $P_{rev} = g^{\rho}$ and $\gamma_{rev} = \rho + H_2(X_k^{\rho})$ and sends (P_{rev}, γ_{rev}) to u_k. When receiving (P_{rev}, γ_{rev}), u_k calculates

$$K_1 = (\gamma_{rev} - H_2(P_{rev}^{x_k}))x_k, K_2 = D_k^{\frac{1}{x_k}}$$

and sends (K_1, K_2) to u_r. After receiving (K_1, K_2), u_r calculates

$$R_1 = \frac{K_1}{x_r}, R_2 = \frac{K_2^{x_r}}{D_r}$$

and sends (R_1, R_2) to the CSP. When receiving (R_1, R_2), the CSP queries all the data-block tags $(m_{j'}, \sigma_{j'})$ generated by user u_r and updates these data-block tags by calculating $\sigma_{j'}^* = (R_2^{m_{j'}} \cdot \sigma_{j'})^{\frac{R_1}{\rho}}$.

• UpdateKey. Firstly, the RN increases by one. According to the new RN, the GM runs PartialKeyGen algorithm to generate a new partial key for each non-revoked user. Finally, non-revoked users regenerate their full public-private keys by running FullUserKeyGen algorithm based on the new partial keys.

Security analysis. Our scheme takes advantage of the idea of key generation in certificateless signatures [2]. Three types of adversaries are taken

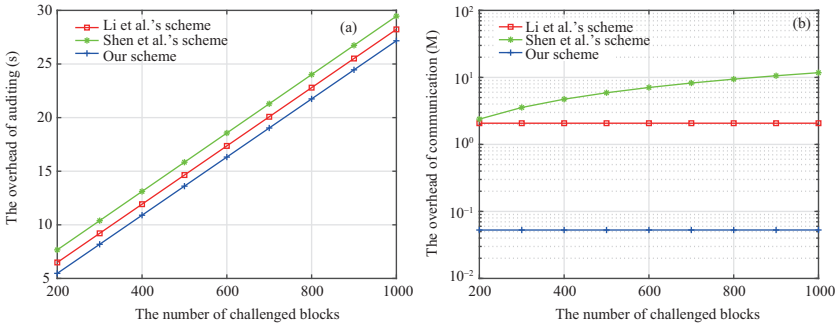

Figure 1 (Color online) (a) Computation cost and (b) communication cost.

into account [5], denoted as $\mathcal{A}_{\mathrm{I}}$, $\mathcal{A}_{\mathrm{II}}$, $\mathcal{A}_{\mathrm{III}}$, respectively. They have different capabilities. A public key replacement attack can be performed by $\mathcal{A}_{\mathrm{I}}$ even without accessing the master key. $\mathcal{A}_{\mathrm{II}}$ can not mount public key replacement attacks but can get the master secret key. $\mathcal{A}_{\mathrm{III}}$ cheats the verifier by forging data integrity proof. For the three adversaries mentioned above, the security of our scheme is described as follows. For $\mathcal{A}_{\mathrm{I}}$ and $\mathcal{A}_{\mathrm{II}}$, the proposed scheme is existentially unforgeable against chosen-message attacks under the CDH assumption. $\mathcal{A}_{\mathrm{III}}$ has an advantage ε to forge data integrity proof under DL, where ε is negligible. Suppose that a file is separated into n blocks, where m blocks are tampered with. The probability of tampered data block being detected is at least $1-(\frac{n-m}{n})^c$, where c is the number of blocks challenged.

Performance analysis. By contrasting our scheme with Li et al's scheme [5] and Shen et al's scheme [1], we analyze the performance of our scheme. Our experiment was carried out under the Windows platform and used a Java programming language with Java Pairing Based Cryptography. So as to facilitate comparison, we adopt the same parameters as scheme [1], that is, $|q|$ and $|p|$ is set to 160 and 512 bits respectively.

The efficiency of public auditing can be shown in Figure 1 (a). 460 blocks should be challenged for 1,000,000 blocks to achieve the detection rate of 99%. In the case, when the number of blocks being challenged grows from 200 to 1000, the time it takes to audit data integrity with our scheme is from 5.4727 to 27.1671 s. In our scheme, the computation cost during the auditing phase grows linearly as the number of blocks to be challenged increases. And our scheme takes less time than schemes [5] and [1].

In the process of public integrity checking, the communication cost is shown in Figure 1 (b). The size of a block index is set to 160 bits. The communication overhead does not change with the number of blocks to be challenged in our scheme and scheme [5]. In scheme [1] the communication cost grows as the number of blocks to be challenged increases. Obviously, our scheme is more effective than schemes [5] and [1].

Conclusion. In this study, we come up with a revocable certificateless public auditing for cloud storage, which supports the key update of the group user and the tag update of the revoked user. The idea of the key exchange is introduced to generate the partial key of the group user. Besides, we solve the problem of tag forgery in Li et al.'s scheme and enhance the robustness of the system. Security analysis and efficiency analysis indicate that our scheme is secure and efficient.

Acknowledgements This work was supported by the National Key R&D Program of China (Grant No. 2017YFB0802000), the National Natural Science Foundation of China (Grant Nos. 61671377, 61772418, 61602378, 61802303), the Key Research and Development Program of Shaanxi (Grant No. 2019KW-053), Guangxi Cooperative Innovation Center of Cloud Computing and Big Data (Grant No. YD1903), and Sichuan Science and Technology Program (Grant No. 2017GZDZX0002). Yinghui ZHANG is supported by New Star Team of Xi'an University of Posts and Telecommunications.

References

1 Shen W, Qin J, Yu J, et al. Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage. IEEE Trans Inf Forensic Secur, 2018, 14: 331–346
2 Zhang Y, Deng R H, Zheng D, et al. Efficient and Robust Certificateless Signature for Data Crowdsensing in Cloud-Assisted Industrial IoT. IEEE Trans Ind Inf, 2019, 15: 5099–5108
3 Zhang Y, Deng R, Liu X, et al. Outsourcing service fair payment based on blockchain and its applications in cloud computing. IEEE Trans Serv Comput, 2018, doi: 10.1109/TSC.2018.2864191
4 Zhang Y, Yu J, Hao R, et al. Enabling efficient user revocation in identity-based cloud storage auditing for shared big data. IEEE Trans Depend Secure Comput, 2018, doi: 10.1109/TDSC.2018.2829880
5 Li J, Yan H, Zhang Y. Certificateless public integrity checking of group shared data on cloud storage. IEEE Trans Serv Comput, 2018, doi: 10.1109/TSC.2018.2789893