SCIENCE CHINA
Information Sciences

• LETTER •

June 2021, Vol. 64 169301:1–169301:2
https://doi.org/10.1007/s11432-019-2856-8

Ciphertext-policy attribute-based proxy re-encryption via constrained PRFs

Zengpeng LI[1,2,3*], Vishal SHARMA[3], Chunguang MA[4*], Chunpeng GE[5,6] & Willy SUSILO[6]

[1]*College of Computer Science and Technology, Qingdao University, Qingdao* 266071*, China;*
[2]*Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang* 550025*, China;*
[3]*ISTD Pillar, Singapore University of Technology and Design* (*SUTD*)*, Singapore* 487372*, Singapore;*
[4]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao* 266590*, China;*
[5]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing* 211106*, China;*
[6]*School of Computing and Information Technology, University of Wollongong, Wollongong* NSW 2522*, Australia*

Received 13 November 2019/Revised 20 January 2020/Accepted 18 March 2020/Published online 15 July 2020

Citation Li Z P, Sharma V, Ma C G, et al. Ciphertext-policy attribute-based proxy re-encryption via constrained PRFs. Sci China Inf Sci, 2021, 64(6): 169301, https://doi.org/10.1007/s11432-019-2856-8

Dear editor,

To maintain the confidentiality of the sensitive data, users tend to encrypt their data under an associated access policy (or attributes) before outsourcing them to the cloud. In the traditional access control model via cryptographic approaches, e.g., ciphertext-policy attribute-based encryption (or CP-ABE), users are allowed to formulate access policies and let the encrypted data associated with access policies. Only authenticated users, whose own attributes matching their formulated access policies, have the privilege to recover the ciphertext. However, once the access policies are formulated, the authorized user cannot update it unless the user downloads and re-encrypts the file under the new access policy. Hence, to achieve dynamic access control, the model of ciphertext-policy attribute-based proxy re-encryption (or CP-AB-PRE) is adopted by integrating the proxy re-encryption (or PRE).

Motivation. However, the current CP-AB-PRE approaches [1–4] cannot provide an efficient fine-grained access control (or FGAC). To date, no work has taken any notice of the advantage of constrained pseudorandom function (CPRF) [5–7] to obtain dynamic access policy under the CP-AB-PRE model. Thus, we provide a solution using key-homomorphic CPRF for the CP-AB-PRE.

CP-AB-PRE from CPRFs construction. In order to achieve FGAC under the model of CP-AB-PRE efficiently, the key point is how to efficiently express the computable predicates p in the language to satisfy the requirements of key-homomorphic CPRF. As discussed in [8], the authors claimed that a circuit-enabled CPRF can evaluate any predicate p. Unfortunately, the prefix CPRF only supports to evaluate predicates that are satisfying inputs starting with a particular prefix. Thus, our a CP-AB-PRE for FGAC solution is based on the circuit-enabled key homomorphic CPRF. In our construction, two cryptographic primitives armed used in the CP-AB-PRE scheme, symmetric encryption $\mathsf{Sym} = (\mathsf{Enc}, \mathsf{Dec})$ with symmetric key κ_{rnd} and a key-homomorphic CPRF $\mathcal{F}$ for predicates p. Furthermore, the access policy is depending on linear secret sharing. Below, we first recap the instantiation of $\mathcal{F}$ in [8] before presenting our main construction. The instantiation is set up for $\kappa = n+\ell$, where n is the input length, ℓ is the circuit depth, and κ is the number of $\mathbb{G}$. Notably, parameters are generated by g_i and a sequence $(\mathbb{G}_1, \mathbb{G}_2, \dots, \mathbb{G}_\kappa)$ of groups $\mathbb{G}_i$ of prime order p as well as elements $D_{i,\beta}$ uniformly drawn from $\mathbb{G}_1$, for $i \in [n]$ and $\beta \in \{0,1\}$. Hence, the CPRF $\mathcal{F}$ on input $x = (x_1, x_2, \dots, x_n) \in \{0,1\}^n$ is defined as $F(\mathrm{pp}, k, x) := e(e(\{D_{i,x_i}\}_{i\in[n]})^k, g_\ell) = g_\kappa^{\kappa\cdot\Pi_{i\in[n]} d_{i,x_i}}$ with $d_{i,\beta}$ such that $D_{i,\beta} = g^{d_{i,\beta}}$. Additionally, we have that for all pp, k_1, k_2, x, then $F(\mathrm{pp}, k_1 + k_2, x) = F(\mathrm{pp}, k_1, x) \cdot F(\mathrm{pp}, k_2, x)$. The detailed construction is as follows.

• $\mathsf{msk} \leftarrow \mathsf{CPABPRE.Setup}(1^\lambda)$. The algorithm first takes as input the security parameter λ and obtains a key $\mathsf{msk}_{\mathrm{cprf}} \in \mathcal{K}$ for $\mathcal{F}$ by invoking $\mathsf{msk}_{\mathrm{cprf}} \leftarrow \mathsf{CPRF.KeyGen}(1^\lambda)$. Then outputs $\mathsf{msk} := \mathsf{msk}_{\mathrm{cprf}}$ as used by $\mathcal{F}$.

• $\mathsf{sk}_{\mathbb{A}} \leftarrow \mathsf{CPABPRE.KeyGen}(\mathsf{msk}, \mathbb{A})$. The key generation algorithm inputs the msk along with the attribute set $\mathbb{A}$, and computes $\mathsf{sk}_{\mathrm{cprf}} = \mathsf{CPRF.Const}^+(\mathsf{msk}_{\mathrm{cprf}}, \mathbb{A})$ by taking as input the random key $\mathsf{msk}_{\mathrm{cprf}}$ for constrained PRF $\mathcal{F}$ and an attribute set $\mathbb{A}$. Then the algorithm generates a master secret key $\mathsf{sk}_{\mathbb{A}} := \mathsf{sk}_{\mathrm{cprf}}$.

• $\mathsf{ct}_\Gamma \leftarrow \mathsf{CPABPRE.Enc}(\mathsf{msk}, m, \Gamma)$. In order to encrypt the plaintext, the encryption algorithm first takes as input a msk, a message m, and the access policy Γ. Next, the algorithm encodes the Γ into the attributes x_Γ for constrained PRF. The details encoding techniques are inspired by the linear secret sharing [9], which implies that any ac-

* Corresponding author (email: lizengpeng@hrbeu.edu.cn, machunguang@hrbeu.edu,cn)

cess structures can be encoded into $\{0,1\}$-LSSS. After that, the algorithm picks a random randomness $r_{\rm cprf}$ for CPRF $\mathcal{F}$ (i.e., $\mathsf{CPRF.Eval}(\cdot)$), a random key $\kappa_{\rm rnd}$ for the symmetric encryption $\mathsf{Sym.Enc}$, and calculates

$$\begin{aligned} c_1 &= \kappa_{\rm rnd} \otimes \mathsf{CPRF.Eval}(\underline{\mathsf{msk}_{\rm cprf}}, [x_\Gamma, r_{\rm cprf}]), \\ c_2 &= \mathsf{Sym.Enc}(\kappa_{\rm rnd}, m) \end{aligned}$$

for the corresponding $[x_\Gamma, r_{\rm cprf}]$. Finally, the algorithm outputs the ciphertext $\mathsf{ct}_\Gamma = ([x_\Gamma, r_{\rm cprf}], c_1, c_2)$.

- $\mathsf{rk}_{\mathbb{A},\Gamma} \leftarrow \mathsf{CPABPRE.ReKeyGen}(\mathsf{msk}_{(i)}, \mathsf{msk}_{(j)})$. In order to generate the re-encryption key, the re-key generation algorithm computes and outputs for $j = i+1$, $\mathsf{rk}_{\mathbb{A},\Gamma} = \mathsf{msk}_{(j)} \circ \mathsf{msk}_{(i)}^{-1} = \mathsf{msk}_{\rm cprf}^{(j)} \circ (\mathsf{msk}_{\rm cprf}^{(i)})^{-1}$.
- $\mathsf{rect}_\Gamma \leftarrow \mathsf{CPABPRE.ReEnc}(\mathsf{rk}_{\mathbb{A},\Gamma}, \mathsf{ct}_\Gamma = ([x_\Gamma, r_{\rm cprf}], c_1, c_2))$. In order to obtain the re-encrypted ciphertext, the re-encryption algorithm takes as input the $\mathsf{rk}_{\mathbb{A},\Gamma}$ and $\mathsf{ct}_\Gamma = ([x_\Gamma, r_{\rm cprf}], c_1, c_2)$, then returns $\mathsf{rect}_\Gamma = \big([x_\Gamma, r_{\rm cprf}], \Delta \bigotimes c_1, c_2\big)$, where $\Delta = \mathsf{CPRF.Eval}(\mathsf{rk}_{\mathbb{A},\Gamma}, [x_\Gamma, r_{\rm cprf}])$.
- $m \leftarrow \mathsf{CPABPRE.Dec}(\mathsf{sk}_{\mathbb{A}}^{(j)}, \mathsf{rect}_\Gamma)$. The output side decryption algorithm first takes a secret key $\mathsf{sk}_{\mathbb{A}}^{(j)}$ as input for a set of $\mathbb{A}$ of attributes, a ciphertext ct_Γ which contains an access policy Γ, then checks whether $\Gamma(\mathbb{A}) = 1$. If so, to recovery the message m, the algorithm decrypts the ciphertext by computing

$$\begin{aligned} \Delta \otimes c_1 &= \mathsf{CPRF.Eval}(\mathsf{rk}_{\mathbb{A},\Gamma}, [x_\Gamma, r_{\rm cprf}]) \otimes c_1 \\ &= \mathsf{CPRF.Eval}(\mathsf{rk}_{\mathbb{A},\Gamma}, [x_\Gamma, r_{\rm cprf}]) \otimes \kappa_{\rm rnd} \\ &\quad \otimes \mathsf{CPRF.Eval}(\underline{\mathsf{msk}_{\rm cprf}^{(i)}}, [x, r_{\rm cprf}]) \\ &= \mathsf{CPRF.Eval}(\underline{\mathsf{msk}_{\rm cprf}^{(j)}}, [x_\Gamma, r_{\rm cprf}]) \otimes \kappa_{\rm rnd}, \end{aligned}$$

then obtains $\kappa_{\rm rnd}$ as the following:

$$\big(\mathsf{CPRF.Eval}(\underline{\mathsf{msk}_{\rm cprf}^{(j)}}, [x_\Gamma, r_{\rm cprf}])\big)^{-1} \otimes \underline{(\Delta \otimes c_1)},$$

and returns $m \leftarrow \mathsf{Sym.Dec}(\kappa_{\rm rnd}, c_2)$; otherwise $\perp$ is returned.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant No. 61802214), National Natural Science Foundation of Shandong Province, China (Grant No. ZR2019BF009), Applied Fundamental Research Project of Qingdao (Grant No. 19-6-2-6-cg), and Foundation of Guizhou Provincial Key Laboratory of Public Big Data (Grant No. 2019BDKFJJ007).

References

1 Li Z P, Ma C G, Wang D. Achieving multi-hop PRE via branching program. IEEE Trans Cloud Comput, 2020, 8: 45–58

2 Li Z P, Ma C G, Wang D. Towards multi-hop homomorphic identity-based proxy re-encryption via branching program. IEEE Access, 2017, 5: 16214–16228

3 Li Z P, Ma C G, Wang D, et al. Toward proxy re-encryption from learning with errors in the exponent. In: Proceedings of Trust Security and Privacy in Computing and Communications, 2017. 683–690

4 Ge C P, Susilo W, Fang L, et al. A CCA-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system. Des Codes Cryptogr, 2018, 86: 2587–2603

5 Boneh D, Waters B. Constrained pseudorandom functions and their applications. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 2013. 280–300

6 Kiayias A, Papadopoulos S, Triandopoulos N, et al. Delegatable pseudorandom functions and applications. In: Proceedings of Computer and Communications Security, 2013. 669–684

7 Boyle E, Goldwasser S, Ivan I. Functional signatures and pseudorandom functions. In: Proceedings of Public Key Cryptography, 2014. 501–519

8 Banerjee A, Fuchsbauer G, Peikert C, et al. Key-homomorphic constrained pseudorandom functions. In: Proceedings of Theory of Cryptography Conference, 2015. 31–60

9 Boneh D, Gennaro R, Goldfeder S, et al. Threshold cryptosystems from threshold fully homomorphic encryption. In: Proceedings of International Cryptology Conference, 2018. 565–596