Comment on "Circuit Ciphertext-Policy Attribute-Based Hybrid Encryption With Verifiable Delegation in Cloud Computing"

Zhengjun Cao and Olivier Markowitch

Abstract—The scheme [1] is flawed because: (1) its circuit access structure is confusingly described; (2) the cloud server cannot complete the related computations; (3) some users can conspire to generate new decryption keys, without the help of the key generation authority.

Index Terms—Ciphertext-policy attribute-based encryption, verifiable delegation, multilinear map, hybrid encryption

1 Introduction

Recently, Xu *et al*. [1] have presented a circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation to the cloud server. The scheme combines a general circuit ciphertext-policy attribute-based encryption and the outsourcing of partial decryption. But we find it fails to keep the compatibility of the two paradigms. Its circuit access structure is falsely described. The cloud server cannot complete the related computations because the encryption exponent s_1 (or the factor $g_k^{s_1}$) is not accessible to the server.

2 Review of the Scheme

The scheme involves four entities: the data owner, the authority, the cloud server, and some target users.

Setup. Let $G_1, \ldots, G_k$ be groups of prime order p, with the each generator g_i. There exist a set of bilinear maps $\{e_{ij} : G_i \times G_j \to G_{i+j} \mid i, j \geqslant 1, i+j \leqslant k\}$ (write as e) satisfying: $\forall\, \mu, \nu \in Z_p$, $e(g_i^\mu, g_j^\nu) = g_{i+j}^{\mu\nu}$. Let H_1, H_2, H_3 be three hash functions. Pick $\alpha, a \in Z_p, h_1, \ldots, h_{2n} \in G_1$, and set the system public parameters as $g = g_1, g_2, \ldots, g_k,\ g_k^\alpha, y = g^a, H_1, H_2, H_3, h_1, \ldots, h_{2n}$. The master key is g^α.

Hybrid-Encryption. Given a message $M \in \{0,1\}^m$, the data owner picks $R \in \{0,1\}^m, s_1, s_2, s_3 \in Z_p$ and computes

$$\begin{aligned}
&C_M = M \oplus H_1(g_k^{\alpha s_1}),\ C'_M = g_{k-1}^{s_1},\ r_1 = H_2(g_k^{\alpha s_1}),\\
&C_R = R \oplus H_1(g_k^{\alpha s_2}),\ C'_R = g_{k-1}^{s_2},\ r_2 = H_2(g_k^{\alpha s_2}),\\
&\sigma_1 = \left(K_H K_{ID_o} H_3^{r_1}(ID_o \| C_M \| C_R)\right)^{s_3},\\
&\sigma_2 = \left(K_H K_{ID_o} H_3^{r_2}(ID_o \| C_M \| C_R)\right)^{s_3},\\
&\sigma_M = \{\sigma_1, g_k^{\alpha s_3}, g_{k-1}^{ts_3}, H_3^{s_3}(ID_o \| C_M \| C_R)\},\\
&\sigma_R = \{\sigma_2, g_k^{\alpha s_3}, g_{k-1}^{ts_3}, H_3^{s_3}(ID_o \| C_M \| C_R)\}.
\end{aligned}$$

For the access structure $f = (n, q, A, B, GateType)$, generate $\bar{f}$ such that negation gates appear only at the input wires. Pick $r_1, \ldots, r_{n+q-1} \in Z_p$, set $r_{n+q} = s_1$, and associate r_w to the wire w according to the below cases (page 124, [1]).

Input wire. For $w \in [1, n]$, $z_w \in Z_p$, compute

$$C_{w,1} = y^{r_w}(yh_w)^{-z_w}, \quad C_{w,2} = g^{z_w}.$$

Gate OR. For $j = depth(w)$, $a_w \in Z_p$, compute

$$C_{w,1} = g^{a_w},\ C_{w,2} = g_j^{a(r_w - a_w r_{A(w)})},\ C_{w,3} = g_j^{a(r_w - a_w r_{B(w)})}. \tag{1}$$

Gate AND. For $j = depth(w)$, $a_w, b_w \in Z_p$, compute

$$C_{w,1} = g^{a_w},\ \ C_{w,2} = g_j^{a(r_w - a_w r_{A(w)} - b_w r_{B(w)})}. \tag{2}$$

The full ciphertext CT contains $C_M, C'_M, C_R, C'_R, \sigma_M, \sigma_R$, and the ciphertexts of f and $\bar{f}$.

Key-Generation. For the user with the attribute index $x \in \{0,1\}^n$, the authority picks $t \in Z_p$ and creates the private key $\{K_H, L, K_1, \ldots, K_n\}$, where

$$K_H = g^\alpha y^t,\ L = g^t,\ K_i = \begin{cases} (yh_i)^t, & x_i = 1,\\ (yh_{n+i})^t, & x_i = 0. \end{cases} \tag{3}$$

For the data owner with the identity ID_o, the authority generates his private key $\{K_H, L, K_{ID_o}\}$, where

$$K_H = g^\alpha y^t,\ L = g^t,\ K_{ID_o} = H_3^t(ID_o). \tag{4}$$

Partial-Decryption. Given the transformation key $TK = \{L, K_1, \ldots, K_n\}$, the full ciphertext CT, and a user's attribute index x, the cloud server evaluates the circuit from the bottom up according to the below three cases.

Input wire. For $w \in [1, n]$, $f_w(x) = x_w$ or $\bar{x}_w$. If $f_w(x) = 1$, compute

$$\begin{aligned}
E_w &= e(K_w, C_{w,2}) \cdot e(L, C_{w,1})\\
&= e(y^t h_w^t, g^{z_w}) \cdot e(g^t, g^{ar_w} y^{-z_w} h_w^{-z_w}) = g_2^{ar_w t}.
\end{aligned}$$

Gate OR. Let $j = depth(w)$. If $f_{A(w)}(x) = 1$, compute

$$\begin{aligned}
E_w &= e(E_{A(w)}, C_{w,1}) \cdot e(C_{w,2}, L)\\
&= e\left(g_j^{ar_{A(w)}t}, g^{a_w}\right) \cdot e\left(g_j^{a(r_w - a_w r_{A(w)})}, g^t\right) = g_{j+1}^{ar_w t}.
\end{aligned} \tag{5}$$

Gate AND. Let $j = depth(w)$. If $f_{A(w)}(x) = 1$, compute

$$\begin{aligned}
E_w &= e(E_{A(w)}, C_{w,1}) \cdot e(E_{B(w)}, C_{w,2}) \cdot e(C_{w,3}, L)\\
&= e(g_j^{ar_{A(w)}t}, g^{a_w}) \cdot e(g_j^{ar_{B(w)}t}, g^{b_w})\\
&\quad \cdot e\left(g_j^{a(r_w - a_w r_{A(w)} - b_w r_{B(w)})}, g^t\right) = g_{j+1}^{ar_w t}.
\end{aligned} \tag{6}$$

If $f(x) = f_{n+q} = 1$, compute

$$C''_M = (g_k)^{as_1 t}. \tag{7}$$

Otherwise, if $\bar{f}(x) = 1$, compute $C''_R = (g_k)^{as_2 t}$. Output the partially ciphertext $CT' = (\sigma_M, C_M, C_R, C'_M, C''_M)$ if $f(x) = 1$, and $(\sigma_R, C_M, C_R, C'_R, C''_R)$ if $f(x) = 0$.

Decryption. Given CT', σ and ID_o, a target user performs as follows. If $f(x) = 1$, compute

$$\chi_M = e(C'_M, K)/C''_M, \quad r_1 = H_2(\chi_M), \tag{8}$$

check $e(\sigma_1, g_{k-1}) = e(H_{3,k-1}^{s_3}(ID_o \| C_M \| C_R), g^{r_1}) \cdot g_k^{\alpha s_3} \cdot e(yH_3(ID_o),\ g_{k-1}^{ts_3})$. Then compute $M = H_1(\chi_M) \oplus C_M$.

If $f(x) = 0$, compute $\chi_R = e(C'_P, K)/C''_R, r_2 = H_2(\chi_R)$, and check the signature. Then compute $R = H_1(\chi_R) \oplus C_R$.

- *Zhengjun Cao is with the Department of Mathematics, Shanghai University Shangda Road 99, Shanghai 200444, China, also with the State key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Xitucheng Road 10, Beijing, China. E-mail: caozhj@shu.edu.cn.*
- *Olivier Markowitch is with Computer Sciences Department, Université Libre de BruxellesBoulevard du Triomphe - CP 2121050 Bruxelles, Belgium. E-mail: olivier.markowitch@ulb.ac.be.*

Manuscript received 16 Apr. 2020; revised 26 July 2020; accepted 31 Aug. 2020. Date of publication 4 Sept. 2020; date of current version 17 Sept. 2020.
(Corresponding author: Zhengjun Cao.)
Recommended for acceptance by S. Chen.
Digital Object Identifier no. 10.1109/TPDS.2020.3021683

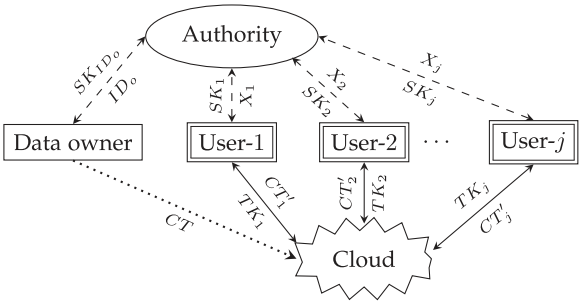

Fig. 1. The proposed system model.

3 The Flaws

The proposed system model can be depicted as Fig. 1. Though the proposed scheme is interesting, we find there are some flaws.

◇ *There are some obvious typos*. For example, the expressions $f_{A(w)(x)} = f_{A(w)(x)} = 1$, $f(x) = f_{n+q} = 1$, $E_{A(w)}, E_{B(w)}$ (the 3rd, 5th, 7th, 10th lines, the right column, page 125, [1]), where f_{n+q}, $E_{A(w)}, E_{B(w)}$ are undefined. The notation $H^{s_3}_{3,k-1}(ID_o \| C_M \| C_R)$ (the 6th, 8th lines, the left column, page 124) should be corrected as $H^{s_3}_3(ID_o \| C_M \| C_R)$.

◇ *The circuit access structure is confusingly described.* It confuses the consistency of Eqs. (1) and (5), and that of Eqs. (2) and (6). Literally, they could be corrected as

$$C_{w,1} = g^{a_w}, \quad C_{w,2} = g_j^{a(r_w - a_w r_{A(w)})} \qquad (1')$$

$$C_{w,1} = g^{a_w},\ C_{w,2} = g^{b_w},\ C_{w,3} = g_j^{a(r_w - a_w r_{A(w)} - b_w r_{B(w)})} \qquad (2')$$

◇ *The cloud server cannot complete the related computation because the encryption exponent s_1 (or the factor $g_k^{s_1}$) is not accessible to the server.* Actually, $C''_M = (g_k)^{as_1 t}$ (see Eq. (7)), the exponents a, t are chosen by the authority, and s_1 is chosen by the data owner. Hence, the server can only access

$$\begin{gathered} g, g_2, \ldots, g_k, y = g^a (\text{from the system's parameters}) \\ g^t, g^{at} h_i^t \text{ or } g^{at} h_{n+i}^t (\text{from the transformation key } TK) \\ C'_M = g_{k-1}^{s_1} (\text{from the full ciphertext } CT), \end{gathered}$$

and obtain $e(g^a, g^t) = g_2^{at}, e(g_2^{at}, g_{k-1}^{s_1}) = g_{k+1}^{ats_1} \neq C''_M$. Likewise, the server cannot finish the computation of C''_R.

If set $C'_M = g_{k-2}^{s_1}$, then $e(g_2^{at}, g_{k-2}^{s_1}) = g_k^{ats_1} = C''_M$. That means the server can properly do the computation. But it still contradicts Eq. (8). In fact, χ_M should be equal to $g_k^{\alpha s_1}$ (see the Hybrid-Encryption phase). Therefore,

$$g_k^{\alpha s_1} = e(g_{k-2}^{s_1}, K)/g_k^{ats_1} \Rightarrow K = g_2^{\alpha + at} = g_2^{at} \cdot g_2^{\alpha}.$$

The scheme has not specified the parameter K. Even worse, the factor g_2^{α} is not accessible to the user. He cannot finish the computation.

Note that if g_2^{α} is directly set as a public parameter, then it becomes unnecessary for the user to ask the cloud server to perform the partial decryption, because the user can recover the plaintext by $M = H_1(e(g_2^{\alpha}, C'_M)) \oplus C_M$, which is independent of C''_M in the partial ciphertext CT'.

◇ *Some users can conspire to generate new decryption keys, without the help of the key generation authority.* As we know, in an attribute-based encryption there are many target receivers whose private keys are matching the specified access structure in a ciphertext. Since all users are conferred the same K_H, L (see Eq. (2)), any two users with the attribute indexes $\tilde{x} = \tilde{x}_1 \cdots \tilde{x}_n \in \{0,1\}^n, \overline{x} = \overline{x}_1 \cdots \overline{x}_n \in \{0,1\}^n$, can collaborate to generate a new private key corresponding to the attribute index $X = X_1 \cdots X_n$, where $X_i = \max\ \{\tilde{x}_i, \overline{x}_i\},\ \ i = 1, \ldots, n$. The new key $\{K_H = g^{\alpha} y^t,\ L = g^t,\ \hat{K}_1,\ \ldots, \hat{K}_n\}$ satisfies more attributes than the original two. Therefore, it can match more access structures. The flaw is due to its simple private key generation mechanism (each subkey is just conferred on each attribute, which seems insufficient for the scenario of outsourcing computing).

◇ *The cloud server and the data owner can conspire to retrieve the user's private key.* Note that the user's private key is just $\{K_H, L, K_1, \cdots, K_n\}$, and the data owner's key is $\{K_H, L, K_{ID_o}\}$. Since the so-called transformation key $TK = \{L,\ K_1, \cdots, K_n\}$ should be submitted to the server for the partial decryption, the server and the data owner can collaborate to recover the user's key. The flaw is also due to its simple private key generation mechanism.

4 Conclusion

We show that the Xu *et al.*'s scheme is flawed. We want to stress that the private key generations for users in an attribute based encryption should be considered carefully.

Acknowledgments

The authors were grateful to the reviewers and the editor for their suggestions.They would like to thank the Open Foundation of State key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (SKLNST-2018-1-15).

References

[1] J. Xu *et al.*, "Circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 119–129, Jan. 2016.