A BGN Type Outsourcing the Decryption of CP-ABE Ciphertexts

Li Zhenlin[1,2(✉)], Zhang Wei[2], Ding Yitao[2], and Bai Ping[2]

[1] Department of Electronic Technique, Engineering University of PAP, Xi'an, Shaanxi, China
lizhenlinl9921l09@l63.com
[2] Key Laboratory of Information Security, Engineering College of PAP, Xi'an, Shaanxi, China
zhaangweei@yeah.net, 15803396982@l63.com, 15502960211@l63.com

Abstract. Cloud computing security is the key bottleneck that restricts its development, and access control on the result of cloud computing is a hot spot of current research. Based on the somewhat homomorphic encryption BGN and combined with Green's scheme that proposed outsourcing the decryption of CP-ABE (Ciphertext-Policy Attribute-Based Encryption) ciphertexts, we constructed a BGN type outsourcing the decryption of CP-ABE ciphertexts. In our construction, partial decryption of ciphertexts is outsourced to the cloud, and only users whose attribute meets the access policy will get the correct decryption. And the scheme supports arbitrary homomorphic additions and one homomorphic multiplication on ciphertexts. Finally, we prove its semantic security under the subgroup decision assumption and compare it with other schemes.

1 Introduction

With the emerge of cloud computing [1], the development of information industry is moving in the fast lane. Cloud computing provides users with massive storage services and powerful computing services, which remarkably makes a contribution to economy [2–5]. However, security issues associated with cloud computing have become increasingly prominent [6]. Kaufman [7] pointed out that the security issue of cloud services was not only one of the biggest challenge of difficulties it faced, but also the problem that should be solved as soon as possible.

If the users save their sensitive data to the cloud server in plaintext, then because the cloud may copy even distort the information, but users do not know such unauthorized behavior of the cloud, which may cause immeasurable loss, the cloud will not be unconditional trusted. In order to prevent malicious leakage and illegal access to sensitive data, users can outsource their data in the encrypted form.

The traditional encryption and decryption model of cloud computing cannot achieve fine-grained access control on the results of cloud computing. In reality, we do not need everyone to gain the final results. In 1984 Shamir [8] proposed Identity-Based Encryption (IBE), in which a user's public key was generated by a unique identifier

L. Barolli et al. (eds.), *Advances in Internetworking, Data & Web Technologies*,
Lecture Notes on Data Engineering and Communications Technologies 6,
DOI 10.1007/978-3-319-59463-7_6

that was related to his/her identity, and the servers did not need query the user's public key certificate any more. Attribute-Based Encryption (ABE), proposed by Sahai and Waters [9], is seen as a promotion of IBE. In ABE system, the user's key and the ciphertexts are associated with attribute, and only when attribute meets the access policy, the user will get the correct decryption, which succeeds in fine-grained access control on the ciphertexts. Due to such good characteristics, ABE scheme has attracted great attention of cryptographers. A large number of relevant research on ABE have emerged in recent years [10–13], and it also has been widely applied to cloud computing security algorithm [14–16], which becomes an important tool for data protection in cloud computing.

In this paper, based on the classic somewhat homomorphic encryption scheme BGN [17], adopting the method of [13] in which we called it outsourcing the decryption of CP-ABE ciphertexts, we propose a BGN type outsourcing the decryption of CP-ABE ciphertexts. In our scheme, partial decryption of ciphertexts is outsourced to the cloud, which greatly reduces the computing overhead of users. The user's private key is associated with his/her attributes, and access control policy is embedded into the ciphertexts, and only the users whose attributes satisfy the access policy can decrypt the ciphertexts. Meanwhile, our scheme can operate on ciphertexts for arbitrary additions and one multiplication.

In Sect. 2, we give the preliminary knowledge of this paper. We present our construction of outsourcing and analyze the homomorphic properties of the scheme in Sect. 3. In Sects. 4 and 5, its security and performance analysis is described respectively. In the next chapter, we make a conclusion.

2 Preliminares

2.1 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map with the following properties:

1. Bilinearity: for all $u, k \in \mathbb{G}$ and $a, b \in Z_p$, then $e\left(u^a, k^b\right) = e(u, k)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.
3. Computable: the bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ can be computed in polynomial time.

2.2 Access Structures

Definition 1 (Access Structure [18]). Let $\{P_1, P_2, \cdots, P_n\}$ be a set of participants and let $P = 2^{\{P_1, P_2, \cdots, P_n\}}$. And access structure Γ is a non-empty subset of $\{P_1, P_2, \cdots, P_n\}$. We define its monotone property as follows: If $A \in \Gamma$ and $A \subseteq B$, then $B \in \Gamma$. We call the sets in Γ the authorized sets, otherwise the unauthorized sets.

2.3 Linear Secret Sharing Schemes

Definition 2 (Linear Secret-Sharing Schemes (LSSS) [18]**).** A secret-sharing scheme Π over a set of participants P is called linear (over Z_p) if

1. The shares of the participants form a vector over Z_p.
2. There exists a $l \times n$ matrix $\boldsymbol{M}$ that is called the share-generating matrix for Π. We define a function ρ that maps every row of the share-generating matrix to a related participant, i.e., for $i = 1, 2, \cdots l$, the value $\rho(i)$ is the participant which is associated with row i. And we build a column vector $\boldsymbol{v} = (s, y_2, \cdots, y_n)$, in which $y_2, \cdots, y_n \in Z_p$ are chosen randomly, and $s \in Z_p$ is just the secret to be shared, then $\boldsymbol{Mv}$ is the vector of l shares of the secret s according to Π. The share $(\boldsymbol{Mv})_i$ belongs to participant $\rho(i)$.

Definition 3 (Linear Reconstruction [18]**).** Each linear secret sharing-scheme has the linear reconstruction property: Suppose that Π is an LSSS for the access structure Γ. Let $S \in \Gamma$ be an authorized set, and let $I \subseteq \{1, 2, \ldots, l\}$ and $I = \{i : \rho(i) \in S\}$. Then, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, there must exist constants $\{w_i \in Z_p\}_{i \in I}$ such that $\sum_{i \in I} w_i \boldsymbol{M}_i \boldsymbol{v} = s$.

2.4 BGN Scheme

The BGN [17] is a classic somewhat homomorphic encryption that is proposed by Boneh, Goh and Nissim, and BGN scheme supports arbitrary homomorphic additions and one homomorphic multiplication. As all know, BGN is the first somewhat homomorphic encryption after the concept of homomorphic encryption was proposed in [19], and in 2010 Gentry [20] implemented BGN on lattice. The scheme is described as follows:

KeyGen(τ): Given a security parameter $\tau \in Z^+$, run $\mathcal{G}(\tau)$ to obtain a tuple $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$. Let $n = q_1 q_2$. Pick two generators $k, u \xleftarrow{R} \mathbb{G}$ randomly and set $h = u^{q_2}$. Then h is a random generator of the subgroup of $\mathbb{G}$ of order q_1. The public key is $PK = (n, \mathbb{G}, \mathbb{G}_1, e, k, h)$ and the private key is $SK = q_1$.

Encrypt(PK, M): The message space is described as $m \in \{0, 1, \cdots, T\}$ with $T < q_2$. We use public key PK to encrypt a message m, pick a random $r \xleftarrow{R} \{0, 1, \cdots n-1\}$ and compute $C = k^m h^r \in$ G. Output C as the ciphertext.

Decrypt(SK, C): We use the private key $SK = q_1$ to decrypt a ciphertext C, observe that $C^{q_1} = (k^m h^r)^{q_1} = (k^{q_1})^m$. To obtain m, we compute the discrete log of C^{q_1} base k^{q_1}. Since $0 \le m \le T$ this takes expected time $O(\sqrt{T})$ using Pollard's lambda [21] method.

Homomorphic properties: The BGN scheme is clearly additively homomorphic:

$$C = C_1 C_2 h^r = k^{m_1} h^{r_1} \cdot k^{m_2} h^{r_2} \cdot h^r = k^{m_1 + m_2} h^{r_1 + r_2 + r} \in \mathbb{G}$$

Multiplicatively homomorphic: Let $k_1 = e(k, k)$ and $h_1 = e(k, h)$, then k_1 is of order n and h_1 is of order q_1. There is some (unknown) $\beta \in Z$ such that $h = k^{\beta q_2}$. We have:

$$\begin{aligned} C &= e(C_1, C_2)h_1^r \\ &= e(k^{m_1}h^{r_1}, k^{m_2}h^{r_2})h_1^r \\ &= k_1^{m_1 m_2}h_1^{m_1 r_2 + m_2 r_1 + \beta q_2 r_1 r_2 + r} \\ &= k_1^{m_1 m_2}h_1^{\tilde{r}} \in \mathbb{G}_1 \end{aligned}$$

Where $\tilde{r} = m_1 r_2 + m_2 r_1 + \beta q_2 r_1 r_2 + r$ is distributed uniformly in z_n. The new ciphertext $C \in \mathbb{G}_1$, because there is no efficient algorithm to make $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}$, the scheme can operate on ciphertexts for only one multiplication.

2.5 Outsourcing the Decryption of ABE Ciphertexts Model

Outsourcing the decryption of attribute-based encryption ciphertexts is proposed by Green, Hohenberger and Waters [13]. The difference from traditional attribute-based encryption is that a transformation algorithm is added in the new scheme, in which partial decryption is outsourced to the cloud, and clients only compute on a little data feedback by the cloud. This method that is called the outsourcing makes full use of the powerful computing ability of the cloud, which greatly improves the decryption efficiency of clients. The traditional attribute-based encryption model and outsourcing the decryption of attribute-based encryption ciphertexts model are shown in Figs. 1 and 2 respectively:

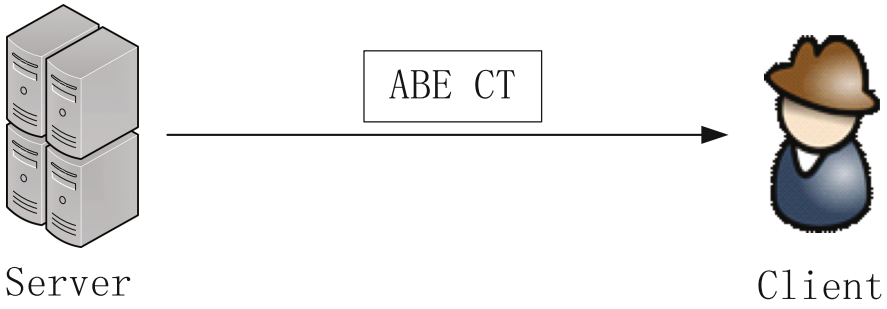

Fig. 1. Traditional ABE model

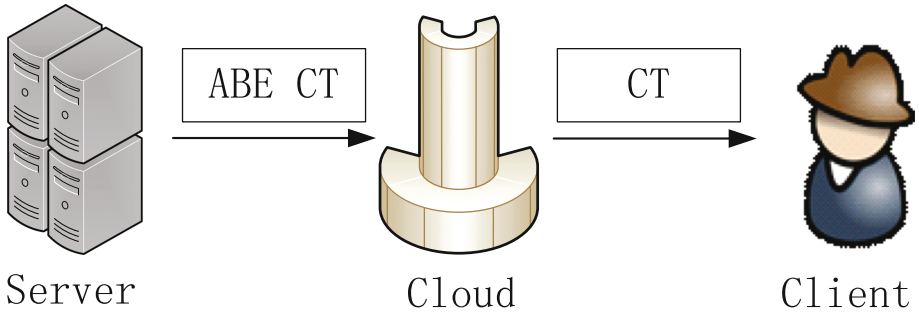

Fig. 2. Outsourcing the decryption of ABE ciphertexts model

In the traditional attribute-based encryption model, which is shown in Fig. 1, the clients must download all ABE ciphertexts to decrypt. Obviously the overhead of storage and computation is too much expensive. In order to solve such shortcomings, the outsourcing model shown in Fig. 2 is designed. The decryption of ABE ciphertexts will be outsourced to cloud which sends partial ciphertexts back, and the clients only need download a small amount of data and compute some simple operations, the storage and computation overhead of the procedure has remarkable reduction.

3 Our Construction

In this part, we construct a BGN type outsourcing the decryption of CP-ABE ciphertexts. Combining the BGN scheme with the idea of outsourcing decryption of attribute-based ciphertexts, we present our construction that can realize access control on the results of cloud outsourcing. Our scheme consists of the following five algorithms:

Setup(λ, U): The setup algorithm takes as input a security parameter λ and a universe description $U = \{0,1\}^*$. It runs $\mathcal{G}(\lambda)$ to obtain a tuple $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$, $\mathbb{G}, \mathbb{G}_1$ are two groups of order $n = q_1 q_2$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be a bilinear map. It picks two generators $k, u \xleftarrow{R} \mathbb{G}$ randomly and set $h = u^{q_2}$, then h is a random generator of the subgroup of $\mathbb{G}$ of order q_1. It then chooses two group $\mathbb{G}', \mathbb{G}'_{\mathrm{T}}$ of prime order p and a hash function F that maps $\{0,1\}^*$ to $\mathbb{G}'$ and hash function H that maps $\mathbb{G}'_{\mathrm{T}}$ to $(0,1)$. Let g be a generator of $\mathbb{G}'$ and $e' : \mathbb{G}' \times \mathbb{G}' \to \mathbb{G}'_{\mathrm{T}}$ be a bilinear map. What's more, it chooses exponents $\alpha, a \in Z_p$ randomly. The algorithm sets $\mathrm{MSK} = (g^{\alpha}, PK)$ as the master secret key. And the public parameters is $\mathrm{PK} = (n, g, k, h, e, e'$ $(g,g)^{\alpha}, g^{a}, F, H, \mathbb{G}, \mathbb{G}_1)$.

Encrypt(PK, m, ($\boldsymbol{M}$, ρ)): The encryption algorithm takes as input the public parameters PK and a message m to encrypt. In addition, it takes as input an LSSS access structure ($\boldsymbol{M}$, ρ). The function ρ associates rows of $\boldsymbol{M}$ to attributes. Let $\boldsymbol{M}$ be an $l \times n$ matrix. The algorithm first chooses a random vector $\boldsymbol{v} = (s, y_2, \cdots, y_n) \in Z_p^n$, and s is the secret to be shared. For $i = 1, 2, \cdots, l$, it computes $\lambda_i = \boldsymbol{M}_i \cdot \mathbf{v}$, in which $\boldsymbol{M}_i$ is the vector corresponding to the i th row of $\boldsymbol{M}$. In addition, the algorithm chooses random $R, r_1, \cdots, r_l \in Z_p$. Output the ciphertext $CT =$

$$
\begin{aligned}
&c = k^{mH(e'(g,g)^{\alpha s})} h^R, C' = g^s \\
&\left(C_1 = g^{a\lambda_1} \cdot F(\rho(1))^{-r_1}, D_1 = g^{r_1}\right) \\
&\cdots\cdots \\
&\left(C_l = g^{a\lambda_l} \cdot F(\rho(l))^{-r_l}, D_l = g^{r_l}\right)
\end{aligned}
$$

KeyGen(MSK, S): The keygen algorithm chooses $t' \in Z_p$ randomly, then it takes as input MSK and an attribute set S to obtain $SK'\left(PK, K' = g^{\alpha} g^{at'}, L' = g^{t'}, \left\{K'_x = F(x)^{t'}\right\}_{x \in S}\right)$. It chooses a random value $z \in Z_p$. Let $t = t'/z$, it then published the transformation key TK as:

$$PK, K = K'^{1/z} = g^{\alpha/z} g^{at}, L = L'^{1/z} = g^t, \{K_x\}_{x\in S} = \left\{K_x'^{1/z}\right\}_{x\in S}$$

and the private key is $SK = (q_1, z, \mathrm{TK})$.

Transform(TK, CT): The transformation algorithm takes as input a transformation key $\mathrm{TK} = \left(\mathrm{PK}, \mathrm{K}, \mathrm{L}, \{K_x\}_{x\in S}\right)$ for a set S and a ciphertext $\mathrm{CT} = (c, C', C_1, \cdots, C_l)$ for access structure $(\boldsymbol{M}, \rho)$. If S does not satisfy the access structure, it outputs $\perp$. Suppose that S satisfies the access structure and let $I \subset \{1, 2, \cdots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, let $\left\{\omega_i \in Z_p\right\}_{i\in I}$ be a set of constants such that if $\{\lambda_i\}$ are valid shares of any secret s according to $\boldsymbol{M}$, then $\sum_{i\in I} \omega_i \lambda_i = s$. The transformation algorithm calculates:

$$\begin{aligned} Q &= e'(C', K) / \left(e'\left(\prod_{i\in I} C_i^{w_i}, L\right) \cdot \prod_{i\in I} e'\left(D_i^{w_i}, K_{\rho(i)}\right)\right) \\ &= e'(g,g)^{s\alpha/z} e'(g,g)^{sat} / \left(\left(\prod_{i\in I} e'(g,g)^{ta\lambda_i w_i}\right)\right) \\ &= e'(g,g)^{s\alpha/z} \end{aligned}$$

It outputs the partially decrypted ciphertext $\mathrm{CT}' = (c, Q)$.

Decrypt(SK, CT): The decryption algorithm takes as input a private key $SK = (q_1, z, \mathrm{TK})$ and a ciphertext CT. If the ciphertext is not partially decrypted, then the algorithm first executes transformation algorithm. If the output is $\perp$, then this algorithm outputs $\perp$ as well. Otherwise, it uses (z, Q) to obtain $e'(g,g)^{s\alpha} = Q^z$, then decrypts c using the partial private key q_1, observe that $c^{q_1} = \left(k^{mH(e'(g,g)^{\alpha s})} h^R\right)^{q_1} = \left(k^{H(e'(g,g)^{\alpha s})q_1}\right)^m$, and using Pollard's lambda method, we compute the discrete log of C_{q_1} base $k^{H(e'(g,g)^{\alpha s})q_1}$ to cover m.

Our outsourcing construction is based on the BGN scheme, so it satisfies the properties of arbitrary additions and one multiplication.

1. Additively Homomorphic: For two ciphertexts $c_1 = k^{m_1 H(e'(g,g)^{\alpha s})} h^{R_1} \in \mathbb{G}$ and $c_2 = k^{m_2 H(e'(g,g)^{\alpha s})} h^{R_2} \in \mathbb{G}$, we have:

$$\begin{aligned} c' &= c_1 c_2 h^R \\ &= \left(k^{m_1 H(e'(g,g)^{\alpha s})} h^{R_1}\right) \cdot \left(k^{m_2 H(e'(g,g)^{\alpha s})} h^{R_2}\right) h^R \\ &= k^{(m_1+m_2) H(e'(g,g)^{\alpha s})} h^{R_1+R_2+R} \in \mathbb{G} \end{aligned}$$

The legal decryptor whose attribute meets the access policy can gain the value of $e'(g,g)^{s\alpha}$, then he will decrypt the ciphertexts through decryption algorithm.

2. Multiplicatively Homomorphic: Let $k_1 = e(k,k)$ and $h_1 = e(k,h)$, then k_1 is of order n and h_1 is of order q_1. There is some (unknown) $\beta \in Z$ such that $h = k^{\beta q_2}$. We have:

$$\begin{aligned} c' &= e(c_1, c_2) h_1^R \\ &= e\left(k^{m_1 H(e'(g,g)^{\alpha s})} h^{R_1}, k^{m_2 H(e'(g,g)^{\alpha s})} h^{R_2}\right) h_1^R \\ &= k_1^{m_1 m_2 H(e'(g,g)^{\alpha s})^2} h_1^{R + (R_1 m_2 + R_2 m_1) H(e'(g,g)^{\alpha s}) + \beta q_2 R_1 R_2} \in \mathbb{G}_1 \end{aligned}$$

In the same way, the legal users can work out $m_1 m_2$. Since there is no efficient algorithm to make $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}$, so the scheme can operate on ciphertexts for only one multiplication.

4 Security

4.1 The Subgroup Decision Problem

We define an algorithm $\mathcal{G}$ such that given a parameter $\tau \in Z^+$, it outputs a tuple $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$ in which $\mathbb{G}, \mathbb{G}_1$ are groups of order $n = q_1 q_2$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is a bilinear map. On input τ, the algorithm $\mathcal{G}$ will work as follows:

1. Generate randomly two τ-bit primes q_1, q_2 and set $n = q_1 q_2 \in Z$.
2. Generate a bilinear group $\mathbb{G}$ of order n as defined above. And let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be the bilinear map.
3. Output $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$.

Obviously the group action in $\mathbb{G}, \mathbb{G}_1$ and the bilinear map are computable in polynomial time in τ. Let $\tau \in Z^+$ and let $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$ be a tuple produced by $\mathcal{G}(\tau)$ where $n = q_1 q_2$. Consider the following problem: given $(n, \mathbb{G}, \mathbb{G}_1, e)$ and an element $x \in \mathbb{G}$, output '1' if the order of x is q_1 and output '0' otherwise; i.e., decide if an element x is in a subgroup of $\mathbb{G}$, without knowing the factorization of n. We call it the subgroup decision problem and define the advantage of $\mathcal{A}$ in solving the subgroup decision problem $SD\text{-}Adv_{\mathcal{A}}(\tau)$ as:

$$SD\text{-}Adv_{\mathcal{A}}(\tau) = \left| \begin{array}{l} \Pr\left[\mathcal{A}(n, \mathbb{G}, \mathbb{G}_1, e, x) = 1 : (q_1, q_2, \mathbb{G}, \mathbb{G}_1, e) \leftarrow \mathcal{G}(\tau), \right. \\ \qquad\qquad \left. n = q_1 q_2, x \leftarrow \mathbb{G}\right] \\ -\Pr\left[\mathcal{A}(n, \mathbb{G}, \mathbb{G}_1, e, x^{q_2}) = 1 : (q_1, q_2, \mathbb{G}, \mathbb{G}_1, e) \leftarrow \mathcal{G}(\tau), \right. \\ \qquad\qquad \left. n = q_1 q_2, x \leftarrow \mathbb{G}\right] \end{array} \right|$$

Definition 4 We say that $\mathcal{G}$ satisfies the subgroup decision assumption if $SD\text{-}Adv_{\mathcal{A}}(\tau)$ is a negligible function in τ for any polynomial time algorithm $\mathcal{A}$.

Theorem 1 Our scheme is semantically secure assuming $\mathcal{G}$ satisfies the subgroup decision assumption.

4.2 Proof

Suppose that a polynomial time algorithm $\mathcal{B}$ breaks the semantic security of the system with advantage $\varepsilon(\tau)$. That's to say, there will exist an algorithm $\mathcal{A}$ that breaks the subgroup decision assumption with the same advantage. Detailed proof procedure is as follows:

1. Algorithm $\mathcal{A}$ chooses a generator $g \in \mathbb{G}$ randomly, sends the public key $(n, \mathbb{G}, \mathbb{G}_1, e, g, x)$ to algorithm $\mathcal{B}$.
2. Algorithm $\mathcal{B}$ outputs two messages $m_0, m_1 \in \{0, 1, \cdots T\}$ to algorithm $\mathcal{A}$, and algorithm $\mathcal{A}$ responds with the ciphertext $C = g^{m_b} x^r \in \mathbb{G}$ for a random $b \xleftarrow{R} \{0, 1\}$ and random $r \xleftarrow{R} \{0, 1, \cdots, n-1\}$ to algorithm $\mathcal{B}$.
3. Algorithm $\mathcal{B}$ outputs $b' \in \{0, 1\}$ for b as its guess. If $b = b'$ algorithm $\mathcal{A}$ outputs 1 (i.e., x is uniformly distributed in a subgroup of $\mathbb{G}$); otherwise $\mathcal{A}$ outputs 0 (i.e., x is uniformly distributed in $\mathbb{G}$).

It is apparent that when x is uniformly distributed in $\mathbb{G}$, the challenge ciphertext C is uniform in $\mathbb{G}$ and is independent of b. Thus, in this case $\Pr|b = b'| = 1/2$. But then, when x is uniformly distributed in q_1-subgroup of $\mathbb{G}$, the public key and challenge C given to $\mathcal{B}$ are as in a real semantic security game. In this case, it is obvious that $\Pr|b = b'| > 1/2 + \varepsilon(\tau)$ by the definition of $\mathcal{B}$. It now follows that $\mathcal{A}$ satisfies $SD\text{-}Adv_{\mathcal{A}}(\tau) > \varepsilon(\tau)$ and hence $\mathcal{A}$ breaks the subgroup decision assumption with advantage $\varepsilon(\tau)$ as required.

Therefore, we prove semantic security of the scheme under the subgroup decision assumption. What's more, it's explicit that the leakage of the attribute does not affect the security of the system. Because even if an attacker got the attribute and the random parameter z, i.e., he could gain the value of $e'(g,g)^{s\alpha}$, however he would fail in computing $c^{q_1} = \left(k^{mH(e'(g,g)^{\alpha s})} h^R\right)^{q_1} = \left(k^{H(e'(g,g)^{\alpha s})q_1}\right)^m$ to cover m without q_1. On the other hand, if the attacker got nothing but q_1, his attribute did not meet the access policy, i.e., he could not work out $e'(g,g)^{s\alpha}$, he cannot decrypt the ciphertexts as well. To sum up, only the legitimate users can cover m in our scheme.

5 Performance Analysis

Green *et al.* [13] presented the idea of outsourcing the decryption of attribute-based encryption ciphertexts, and Boneh *et al.* [17] proposed a classic somewhat homomorphic encryption. In this section, we compared our scheme with the literature [13, 17] in the following aspects: whether to support homomorphic operation, the effect of attributes leak on security, the size of ciphertext and the decryption ops. The results are shown in Tables 1 and 2.

Table 1. Comparison with Green scheme

Scheme	Homomorphic	Effect of attributes leak on security
Green [13]	No	Deadly
Ours	Yes	Hardly

From Table 1, it is distinct that compared with [13], ours do support homomorphic operation on ciphertexts outsourced to the cloud. Moreover, the security is not directly determined by attributes, which means that the malicious users cannot carry out collusion attacks, our system security is based on the subgroup decision assumption.

Table 2. Comparison with BGN scheme

Scheme	Access control	Ciphertext size	Decryption Ops
BGN [17]	No	$O(T)$	$O(\sqrt{T})$
Ours	Yes	$O(T)$	$O(\sqrt{T}) + O_P$

From Table 2, O_P stands for the time to compute hash function H, and compared with BGN scheme, although the decryption overhead increases, the ciphertext length is just the same. On the other hand, the BGN scheme fails in providing fine-grained access control, however ours achieves restricting who can get the results of homomorphic encryption through employing ABE.

6 Summary

In this article, we bring the thought of outsourcing the decryption of ABE ciphertexts into BGN scheme, and propose our BGN type outsourcing the decryption of CP-ABE ciphertexts, which is suitable for the cloud environment. By using the method of attribute-based encryption, we can solve the problem of access control on cloud computing results, and the users' computation overhead in decryption reduces remarkably, because the process of outsourcing improves users' decrypting efficiency. Further work is to explore the combination of outsourcing the decryption of ABE ciphertexts with the full homomorphic encryption, and to construct a more efficient and practical outsourcing scheme for the full homomorphic encryption based on the cloud.

References

1. Hand, E.: Head in the clouds. Nature **449**(7165), 963 (2007)
2. Alamareen, A., Al-Jarrah, O., Aljarrah, I.A.: Image mosaicing using binary edge detection algorithm in a cloud-computing environment. Int. J. Inf. Technol. Web. Eng. **11**(3), 1–14 (2016)
3. Almiani, M., Razaque, A., Al-Dmour, A.: Privacy preserving framework to support mobile government services. Int. J. Inf. Technol. Web. Eng. **11**(3), 65–78 (2016)
4. Dam, H.K., Ghose, A., Qasim, M.: An agent-mediated platform for business processes. Int. J. Inf. Technol. Web. Eng. **10**(2), 43–61 (2015)
5. Mezghani, K., Ayadi, F.: Factors explaining IS managers attitudes toward cloud computing adoption. Int. J. Technol. Human Interact. **12**(1), 1–20 (2016)
6. Khan, N., Al-Yasiri, A.: Cloud security threats and techniques to strengthen cloud computing adoption framework. Int. J. Inf. Technol. Web. Eng. **11**(3), 50–64 (2016)

7. Kaufman, L.M.: Data security in the world of cloud computing. IEEE Secur. Priv. **7**(4), 61–64 (2009)
8. Shamir A.: Identity-based cryptosystems and signature schemes. In: Workshop on the Theory and Application of Cryptographic Techniques, pp. 47–53. Springer, Heidelberg (1984)
9. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 457–473. Springer, Heidelberg (2005)
10. Pirretti, M., Traynor, P., McDaniel, P., et al.: Secure attribute-based systems. J. Comput. Secur. **18**(5), 799–837 (2010)
11. Ning, J., Dong, X., Cao, Z., et al.: White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. IEEE Trans. Inf. Forensics Secur. **10**(6), 1 (2015)
12. Zhang, K., Ma, J., Liu, J., et al.: Adaptively secure multi-authority attribute-based encryption with verifiable outsourced decryption. Sci. China Inf. Sci. (2016)
13. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of ABE ciphertexts. In: USENIX Security Symposium (2011)
14. Wan, Z., Liu, J., Deng, R.H.: HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Trans. Inf. Forensics Secur. **7**(2), 743–754 (2012)
15. Yang, K., Jia, X., Ren, K., et al.: DAC-MACS: effective data access control for multi-authority cloud storage systems. IEEE Trans. Inf. Forensics Secur. **8**(11), 1790–1801 (2013)
16. Wang, S., Zhou, J., Liu, J., et al.: An efficient file hierarchy attribute-based encryption scheme in cloud computing. IEEE Trans. Inf. Forensics Secur. **11**(6), 1 (2016)
17. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Theory of Cryptography Conference, pp. 325–341. Springer, Heidelberg (2005)
18. Beimel A.: Secure schemes for secret sharing and key distribution. Int. J. Pure Appl. Math. (1996)
19. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Found. Secure Comput. **4**(11), 169–180 (1978)
20. Gentry, C., Halevi, S., Vaikuntanathan, V.: A simple BGN-type cryptosystem from LWE. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 506–522. Springer, Heidelberg (2010)
21. Menezes, A.J., Oorschot, P.V., Vanstone, S.A.: Handbook of Applied Cryptography, pp. 425–488. CRC Press, Boca Raton (1999)