An Attribute Based Encryption Scheme with Fine-Grained Attribute Revocation

Qiang Li
Institute of Information Engineering,
Chinese Academy of Sciences
Graduate University of Chinese Academy of Sciences
Beijing, China
Email: liqcas@gmail.com

Dengguo Feng
Institute of Software
Chinese Academy of Sciences
Beijing, China
Email: feng@is.iscas.ac.cn

Liwu Zhang
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
Email: zlw@is.iscas.ac.cn

***Abstract*—As a new public key primitive, attribute-based encryption (ABE) is envisioned to be a promising tool for implementing fine-grained access control. When applying ABE schemes to practical applications, revocation mechanism is very necessary for any ABE schemes involving many users. Revocation for ABE schemes is a challenge issue since each attribute is conceivably shared by multiple users. Revocation of any single user would affect others who share his attributes. In this paper, we propose a KP-ABE scheme with fine-grained attribute revocation under the direct revocation model. In our scheme, we can revoke one attribute of a user instead of all attributes issued to him and the user can complete decryption as long as the unrevoked attributes of the user satisfy the access structure. The revocation does not affect any other user's private key. Moreover, our scheme supports an important property for achieving the user accountability to prevent illegal key sharing among colluding users. We show how to construct such a KP-ABE scheme with fine-grained attribute revocation and prove its security under the q-BDHE assumption in the standard model.**

Keywords: attribute-based encryption; key-policy; revocation; accountability

I. Introduction

The concept of attribute-based encryption (ABE) which was proposed by Sahai and Waters[1] has attracted much attention in research in recent years. There are two complementary forms of ABE: key policy attribute based encryption(KP-ABE)[2] and ciphertext policy attribute based encryption(CP-ABE)[4]. For the first time, ABE enables public key based one-to-many encryption. However, two important problems must be considered when applying ABE schemes to practical applications. The first problem is revocation mechanism which is necessary for any ABE schemes involving many users. And the second problem is user accountability which prevents illegal key sharing among colluding users.

Supported by the National Natural Science Foundation of China under Grant No. 91118006; The National High-Tech Research and Development Plan of China under Grant Nos. 2011AA01A203, 2012AA01A403; The Opening Project of Key Lab of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security) under Grant No. C11604.

For the revocation scheme of ABE, Pirretti *et al.*[5] proposed the first key revocation scheme. Later, Boldyreva *et al.*[6] proposed a revocable ABE scheme extended from their revocable IBE. Both of the two schemes require the users to periodically go to the authority for key reissuing. Based on the previous work mentioned above, Attrapadung *et al.*[7] defined two revocation models explicitly: indirect revocation model which enforces revocation by the key authority who releases a key update material periodically; the other is direct revocation model which enforces revocation directly by the sender who specifies the revocation list while encrypting. Attrapadung *et al.*[8] also proposed a new cryptosystem called Broadcast ABE which supports direct identity revocation mechanism. We note that, in the ABE schemes mentioned above, the revocation of user identity would revoke the entire user access privilege and the revocation of attributes would affect others who share the same attributes.

For the user accountability of ABE, Hinek *et al.*[9] proposed the first scheme which resolved the key abuse problem of users. But in their scheme, another third party should be involved in each user's decryption which makes it impractical. Li *et al.*[10] use the technique of identity-based encryption with wildcards to achieve the accountability of users. Li *et al.*[11] and Yu *et al.*[12] also proposed ABE schemes which achieve the user accountability. In their schemes, user accountability can be achieved in black-box model by embedding additional user specific information into the attribute private key issued to that user.

978-1-4673-0921-9/12/$31.00 ©2012 IEEE

Our Contribution

In this paper, we propose a new KP-ABE scheme with fine-grained attribute revocation. In our scheme, we can revoke one attribute of a user instead of all attributes issued to him and the user can complete decryption as long as the unrevoked attributes of the user satisfy the access structure. The revocation does not affect any other user's private key. Besides that, by embedding additional user identity in the private key issued to the user, our scheme also achieves the user accountability to prevent illegal key sharing among colluding users. Finally, we prove the security of our KP-ABE scheme with fine-grained attribute revocation in the proposed model under the q-BDHE assumption in the standard model.

II. Preliminaries

A. Attribute Based Encryption with Fine-Grained Attribute Revocation

Denote *U* to be the universe of all the users. An attribute based encryption scheme that support fine-grained attribute revocation under direct revocation model consists of five algorithms **Setup, Encryption, KeyGen, Decryption, Trace**, we describe each of these five algorithm below:

Setup(λ) $\rightarrow$ (pk, msk)**:** This is a randomized algorithm that takes as input a security parameter λ. It outputs a public key pk and a master secret key msk;

Encryption($\omega, R_j, \mathcal{M}, pk$) $\rightarrow$ ct**:** This is a randomized algorithm that takes as input an attribute set ω, a revocation list $R_j \subseteq U$ of attribute $j \in \omega$, a message $\mathcal{M}$, and the public key pk. It outputs a ciphertext ct.

KeyGen($ID, \mathbb{A}, msk, pk$) $\rightarrow$ sk**:** This is a randomized algorithm that takes as input a user index $ID \in U$, an access structure $\mathbb{A}$, the master secret key msk, and the public key pk. It outputs a user private key sk.

Decryption(ct, ω, R_j, sk, pk)$\rightarrow$ $\mathcal{M}$**:** This algorithm takes as input a ciphertext ct that was encrypted under an attribute set ω with an attribute revocation list $R_j \subseteq U$ of attribute $j \in \omega$, the user private key sk for user $ID \in U$ with an access structure $\mathbb{A}$. Define the attribute set ω' for the user ID as: if $ID \in R_j$, let $\omega' = \omega - \{j\}$; otherwise, let $\omega' = \omega$. It outputs the message $\mathcal{M}$ if and only if the attribute set $\omega' \in \mathbb{A}$.

Trace(pk, sk) $\rightarrow$ ID**:** This algorithm is used to trace a decryption key to its original holder. It takes as input a decryption key sk and the public key pk. It outputs an identity associated with this decryption key.

B. Selective Security Model

The selective security notion for the above scheme is defined in the following game:

Init: The adversary declares an attribute set ω^* and an attribute revocation list $R_j^* \subseteq U$ of attribute $j \in \omega^*$ that it wishes to be challenged upon.

Setup: The challenger runs the Setup algorithm of ABE and gives the public key pk to the adversary.

Phase1: The adversary is allowed to issue queries for user private key sk of the user $ID \in U$ with the access structure $\mathbb{A}$, such that $\omega^{*'}$ (see the definition of *Decryption*) doesn't satisfy the access structure $\mathbb{A}$.

Challenge: The adversary submits two equal-length messages M_0, M_1. The challenger chooses a random bit $b \in \{0, 1\}$, and computes the challenge ciphertext ct^* under the attribute set ω^* with the attribute revocation list R_j^*.

Phase2: Phase1 is repeated.

Guess: The adversary outputs a guess b' of *b*.

The advantage of the adversary in this game is defined as Pr[$b' = b$]-1/2. We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase1 and Phase2.

Definition1. An ABE scheme defined above is secure in selective security model if all polynomial time adversaries have at most a negligible advantage in the above game.

C. Decision q-BDHE Assumption

Security of our scheme is based on the complexity assumption called the Decision q-BDHE (Bilinear Diffie-Hellman Exponent) assumption. It is stated as follows:

Let $\mathbb{G}_1$ be a bilinear group of prime order *p*, given a vector of 2*q*+1 elements:

$$(g, g^s, g^a, g^{a^2}, ..., g^{a^q}, g^{a^{q+2}}, ..., g^{a^{2q}}) \in \mathbb{G}_1^{2q+1}$$

We say that the Decision q-BDHE assumption holds in $\mathbb{G}_1$ if no polynomial-time algorithm has a non-negligible advantage to distinguish $e(g,g)^{sa^{q+1}}$ from a random element in $\mathbb{G}_2$.

III. Construction

A. Our Construction

Our construction is inspired by the identity-based revocation mechanism of Attrapadung *et al.*[3] and KP-ABE of Goyal *et al.*[2]. In our scheme, if one attribute of a user is revoked, the user can complete decryption as long as the unrevoked attributes of the user satisfy the access structure. Our scheme achieves the user accountability by embedding additional user identity in the private key and the identity can be detected by the trace algorithm. In our scheme, we use Linear Secret Sharing Schemes (LSSS) to represent access structure and the relevant background on LSSS can be found in [2].

We first introduce the concept of zero inner-product and see how the technique is used to construct our scheme. A private key for an identity ID is defined by setting a vector $\mathbf{X} = (x_1, ..., x_n)^\top$ such that $x_i = ID^{i-1}$. To encrypt with a revoked user set $S = \{ID_1, ..., ID_q\}$, one defines $\mathbf{Y} = (y_1, ..., y_n)^\top$ as the coefficient vector of $Ps[Z]$ from

$$Ps[Z] = \sum_{i=1}^{q+1} y_i Z^{i-1} = \prod_{ID_j \in S} (Z - ID_j) \qquad (1)$$

where, if $q+1 < n$, the coordinates $y_{q+2}, ..., y_n$ are set to 0. By doing so, we note that $Ps[ID] =< \mathbf{X}, \mathbf{Y} >$ evaluates to 0 iff $ID \in S$.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order p, and g is a generator of $\mathbb{G}_1$. In addition, define e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ to be a bilinear map. We assume that the sender uses at most m attributes when encryption. The parameter n specifies the maximal size of revoked user set every ciphertext has.

Setup(n): The algorithm randomly chooses $\alpha, \overrightarrow{\alpha} = (\alpha_1, \alpha_2, ...\alpha_n) \in \mathbb{Z}_p$ and sets $\overrightarrow{H} = (h_1, ..., h_n)^\top = (g^{\alpha_1}, ..., g^{\alpha_n})$. It then randomly chooses $\{t_{0,i}, t_{1,i} \in \mathbb{G}_1\}_{i=0,1,...,m}$, and defines two functions $T_0(x), T_1(x) : \mathbb{Z}_p \to \mathbb{G}_1$ by $T_0(x) = \prod_{i=0}^m t_{0,i}^{(x^i)}, T_1(x) = \prod_{i=0}^m t_{1,i}^{(x^i)}$. The master key is $\mathbf{msk} = \{\alpha, \alpha_1, \{t_{0,i}, t_{1,i}\}_{i=0,1,...,m}\}$, while the public key is $\mathbf{pk} = \{g, e(g,g)^\alpha, \overrightarrow{H} = (h_1, ..., h_n)^\top\}$.

Encryption($pk, \omega, R_j, \mathcal{M}$): To encrypt a message $\mathcal{M}$ under a set of attribute ω, with a user revocation list R_j(where $|R_j| < n$) of attribute $j \in \omega$, the algorithm first defines $\mathbf{Y} = (y_1, ..., y_n)^\top$ as the coefficient vector of $Ps[Z]$ from equation(1), it then chooses a random value $s \in \mathbb{Z}_p$. The ciphertext is published as $\mathbf{ct} = (C, C_1, C_{2,0}, C_{2,1}, C_3)$ where

$$C = \mathcal{M}e(g,g)^{\alpha s}, C_1 = g^s, C_{2,0} = \{C_{2,0}^x = T_0(x)^s\}_{x \in \omega},$$

$$C_{2,1} = \{C_{2,1}^x = T_1(x)^s\}_{x \in \omega - \{j\}}, C_3 = (h_1^{y_1} ... h_n^{y_n})^s.$$

KeyGen($ID, (M, \rho), pk, msk$): To generate a secret key for user ID under an LSSS access structure (M, ρ), the algorithm first defines a vector $\overrightarrow{X} = (x_1, ..., x_n)^\top$ such that $x_i = ID^{i-1}$ for $i = 1$ to n. Let M be an $l \times k$ matrix. It then chooses random $r, \{z_{i,0}\}_{i \in [2,...,k]}, \{z_{i,1}\}_{i \in [2,...,k]} \in \mathbb{Z}_p$ and defines two vectors $\overrightarrow{v_0} = (\alpha + r\alpha_1, z_{2,0}, ..., z_{k,0})^\top, \overrightarrow{v_1} = (\alpha, z_{2,1}, ..., z_{k,1})^\top$. For $i = 1$ to l, it calculates $\lambda_{i,0} = \mathbf{M}_i \cdot \overrightarrow{v_0}, \lambda_{i,1} = \mathbf{M}_i \cdot \overrightarrow{v_1}$, where $\mathbf{M}_i$ is the vector corresponding to the ith row of M. The algorithm then randomly chooses $\{r_{i,0}\}_{i \in [1,...,l]}, \{r_{i,1}\}_{i \in [1,...,l]} \in \mathbb{Z}_p$ and outputs the private key as $\mathbf{sk} = (D_{1,0}, D_{1,1}, D_{2,0}, D_{2,1}, D_3, K)$ where

$$D_{1,0} = \{D_{1,0}^i = g^{\lambda_{i,0}} T_0(\rho(i))^{r_{i,0}}\}_{i \in [1,...,l]},$$

$$D_{2,0} = \{D_{2,0}^i = g^{r_{i,0}}\}_{i \in [1,...,l]},$$

$$D_{1,1} = \{D_{1,1}^i = g^{\lambda_{i,1}} T_1(\rho(i))^{r_{i,1}}\}_{i \in [1,...,l]},$$

$$D_{2,1} = \{D_{2,1}^i = g^{r_{i,1}}\}_{i \in [1,...,l]},$$

$$D_3 = g^r, K = \{K_i = (h_1^{-\frac{x_i}{x_1}} \cdot h_i)^r\}_{i \in [2,...,n]}.$$

Indeed, we can also write $K_{\mathbf{X}} = (K_2, ..., K_n) = g^{r \cdot M_{\mathbf{X}}^\top \overrightarrow{\alpha}}$, where the matrix $M_{\mathbf{X}} \in (\mathbb{Z}_p)^{n \times (n-1)}$ is defined by $M_{\mathbf{X}} = \begin{pmatrix} -\frac{x_2}{x_1} & -\frac{x_3}{x_1} & \cdots & -\frac{x_n}{x_1} \\ & I_{n-1} & & \end{pmatrix}$

Decryption(pk, ID, sk, ct): For a user with a secret key $\mathbf{sk}_{\{ID,(M,\rho)\}}$, ciphertext $\mathbf{ct}$ with an attribute set ω and a user revocation list R_j, if $ID \in R_j$, we let $\omega' = \omega - \{j\}$, if not, we let $\omega' = \omega$. The user could decrypt successfully if and only if the attribute set ω' satisfies the access structure $(\mathbf{M}, \rho)$. The decryption algorithm proceeds as follows:

1) if $ID \notin R_j$: The algorithm first defines $\mathbf{X}$ from ID and $\mathbf{Y}$ from R_j as usual. Then it computes

elements K:

$$K = \prod_{i=2}^{n} K_i^{y_i} = (h_1^{-<\mathbf{X},\mathbf{Y}>/x_1} \prod_{i=1}^{n} h_i^{y_i})^r.$$

so that when$< \mathbf{X}, \mathbf{Y} > \neq 0 (ID \notin R_j)$, the following computation can be done:

$$\tau = \left(\frac{e(K, C_1)}{e(C_3, D_3)} \right)^{-\frac{x_1}{<\mathbf{X},\mathbf{Y}>}} = e(g, g)^{rs\alpha_1}.$$

Let $I = \{i : \rho(i) \in \omega'\}$. The algorithm can calculate corresponding sets of reconstruction constants $\{\mu_i \in \mathbb{Z}_p\}_{i \in I}$, such that $\sum_{i \in I} \mu_i \lambda_{i,0} = \alpha + r\alpha_1$. It then computes:

$$\phi = \prod_{i \in I} \left(\frac{e(C_1, D_{1,0}^i)}{e(C_{2,0}^{\rho(i)}, D_{2,0}^i)} \right)^{\mu_i} = e(g, g)^{s\alpha + sr\alpha_1}.$$

The algorithm can then compute $e(g, g)^{s\alpha} = \phi / \tau$, and divide out this value from C and obtain the message $\mathcal{M}$.

2) if $ID \in R_j$: Let $I = \{i : \rho(i) \in \omega'\}$. The algorithm can calculate corresponding sets of reconstruction constants $\{\mu_i \in \mathbb{Z}_p\}_{i \in I}$, such that $\sum_{i \in I} \mu_i \lambda_{i,1} = \alpha$. It then computes:

$$\prod_{i \in I} \left(\frac{e(C_1, D_{1,1}^i)}{e(C_{2,1}^{\rho(i)}, D_{2,1}^i)} \right)^{\mu_i} = e(g, g)^{s\alpha}.$$

The algorithm can then divide out this value from C and obtain the message $\mathcal{M}$.

Trace(pk, sk): Let $\mathbf{sk} = (D_{1,0}, D_{1,1}, D_{2,0}, D_{2,1}, D_3, K)$ be a valid decryption key, the algorithm computes:

$$\phi = \frac{e(g, K_2)}{e(h_2, D_3)} = e(g, h_1)^{-r \cdot ID}.$$

The algorithm can also compute:

$$\varphi = e(h_1, D_3)^{-1} = e(h_1, g^r)^{-1} = e(g, h_1)^{-r}.$$

Let $U = \{ID_1, ID_2, ..., ID_k\}$ to be the universe of all the users, then the algorithm tests whether $\varphi^{ID_i} = \phi$ for $i = 1$ to k, when the equation is satisfied, the algorithm returns the ID associated with this decryption key. The party who runs this algorithm may be the authority of the system or some other parties who knew the universe of all the users in the system.

B. Security

Theorem1. If an adversary can break our scheme with advantage ε in the selective security model, then a simulator can be constructed to solve the Decision q-BDHE problem.

proof. The proof of Theorem 1 is given in the appendix.

IV. Conclusion

In this paper, we present a new KP-ABE scheme with fine-grained attribute revocation. In our scheme, the sender can revoke one attribute of a user instead of the identity without affecting any other user's private key. By embedding additional user identity in the private key issued to the user, our scheme also achieves the user accountability to prevent illegal key sharing among colluding users.

References

[1] Sahai A, Waters B.: Fuzzy identity-based encryption. In: Cramer R, ed. Advances in Cryptology - EUROCRYPT 2005. Berlin: Springer-Verlag, 2005.457–473 (2005)

[2] Shamir A.: Goyal V, Pandey O, Sahai A, Waters B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on Computer and communications security. New York: ACM, 2006.89–98 (2006)

[3] Attapadung, N., Libert, B.: Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In: Catalano, D., et al. (eds.) PKC 2011. LNCS, vol.6571 Springer, Heidelberg (2011)

[4] Bethencourt J, Sahai A, Waters B.: Ciphertext-policy attribute-based encryption. In: Proceedings of the 2007 IEEE Symposium on Security and Privacy. Washington DC: IEEE Computer Society, 2007.321–334 (2007)

[5] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters.: Secure Atrribute-Based Systems. In Proc. of CCS'06, New York, NY, USA. (2006)

[6] Boldyreva A, Goyal V, Kumar V.: Identity-based encryption with efficient revocation. In: Proceedings of the 15th ACM conference on Computer and communications security. New York: ACM, 2008.417–426 (2008)

[7] Attrapadung N, Imai H.: Attribute-based encryption supporting direct/indirect revocation modes. In: Parker MG, ed. Cryptography and Coding. Berlin: Springer-Verlag, 2009.278–300 (2009)

[8] Attrapadung N, Imai H.: Conjunctive broadcast and attribute-based encryption. In: Shacham H, Waters B, eds. Pairing-Based Cryptography - Pairing 2009. Berlin: Springer-Verlag, 2009.248–265 (2009)

[9] M. Jason Hinek, Shaoquan Jiang, Reihaneh Safavi-Naini, and Siamak Fayyaz Shahandashti.: Attribute-Based Encryption with Key Cloning Protection. Available at http://eprint.iacr.org/2008/478.

[10] Li, J., Ren, K., Kim, K.: A2be: Accountable attribute-based encryption for abuse free access control. Cryptology ePrint Archive, Report 2009/118, http://eprint.iacr.org/ (2009)

[11] J. Li, K. Ren, B. Zhu, and Z. Wan.: Privacy-Aware Attribute-Based Encryption with User Accountability. In ISC 2009, volume 5735 of LNCS, 2009.347–362 (2009)

[12] Yu SC, Ren K, Lou WJ, Li J.: Defending against key abuse attacks in KP-ABE enabled broadcast systems. In: Proc. of the Security and Privacy in Communication Networks. Berlin, Heidelberg: Springer-Verlag, 2009.311–329 (2009)

Appendix

A. Proof of Theorem 1

Proof. Suppose there exists a polynomial-time adversary $\mathcal{A}$, that can attack our scheme in the

selective security model with advantage ε. We can build a simulator $\mathcal{B}$ that can solve the Decision q-BDHE problem with advantage $\frac{\epsilon}{2}$. The simulation proceeds as follows:

The challenger sets:

$$\overrightarrow{Y} = (g, g^s, g_1 = g^a, g_2 = g^{a^2}, ..., g_q = g^{a^q}, g_{q+2} = g^{a^{q+2}}, ..., g_{2q} = g^{a^{2q}}).$$

Then the challenger flips a fair binary coin μ: If $\mu = 0$, the challenger set $Z = e(g_1, g_q)^s$; If $\mu = 1$, then pick a random element Z from $\mathbb{G}_2$. Finally, the challenger gives $(\overrightarrow{Y}, Z)$ to the simulator $\mathcal{B}$. $\mathcal{B}$ proceeds as follows:

Init. The simulator $\mathcal{B}$ runs adversary $\mathcal{A}$. $\mathcal{A}$ selects a attribute set ω^* and a user revocation list R_j^* of attribute $j \in \omega^*$ that it wishes to be challenged upon.

Setup. The simulator $\mathcal{B}$ acts as follows:

1) It chooses random $\alpha' \in \mathbb{Z}_p$ and implicitly sets $\alpha = \alpha' + a^{q+1}$ by letting $e(g,g)^{\alpha} = e(g^a, g^{a^q})e(g,g)^{\alpha'}$.
2) Let $R_j^* = \{ID_1, ..., ID_m\}$, where $m \leq q$. Elements $\overrightarrow{H} = (h_1, ..., h_n)^{\top}$ are then defined as follows. For each $k \in [1, m]$, $\mathcal{B}$ considers the vector $\overrightarrow{X_k} = (x_{k,1}, ..., x_{k,n}) = (1, ID_k, ID_k^2, ..., ID_k^{n-1})$ and selects $\overrightarrow{b_k} \in \mathbb{Z}_p$ such that

$$\overrightarrow{b_k}^{\top} \cdot M_{X_k} = \overrightarrow{b_k}^{\top} \cdot \begin{pmatrix} -\frac{x_{k,2}}{x_{k,1}} & ... & -\frac{x_{k,n}}{x_{k,1}} \\ & I_{n-1} & \end{pmatrix} = \overrightarrow{0}$$

The simplest candidate consists of the vector $\overrightarrow{b_k} = (1, \frac{x_{k,2}}{x_{k,1}}, \frac{x_{k,3}}{x_{k,1}}, ..., \frac{x_{k,n}}{x_{k,1}})^{\top}$. Then $\mathcal{B}$ considers the $n \times q$ matrix $B = (\overrightarrow{b_1}|...|\overrightarrow{b_m}|0|...|0)$ whose kth column consists of $\overrightarrow{b_k}$, for $k = 1$ to m, and where the $q - m$ remaining columns are $\overrightarrow{0}$. It defines $\overrightarrow{z} = (z_1, ..., z_q)^{\top} \in (\mathbb{Z})^n$ such that $z_i = a^{q+1-i}$ by setting $g^{\overrightarrow{z}} = (g^{a^q}, ..., g^a)^{\top}$. Then it implicitly sets $\overrightarrow{\alpha} = B \cdot \overrightarrow{z} + \overrightarrow{\delta}$ by randomly choosing $\overrightarrow{\delta} \leftarrow \mathbb{Z}_p^n$ and defining $\overrightarrow{H} = g^{B \cdot \overrightarrow{z}} \cdot g^{\overrightarrow{\delta}}$, which is uniformly distributed as required. Due to the above definition, the matrix B is defined in such a way that, for each $k \in [1, m]$, the kth column of $M_{X_k}^{\top} \cdot B \in (\mathbf{Z}_p)^{(n-1) \times q}$ is $\overrightarrow{0}$, so that $M_{X_k}^{\top} \cdot B \cdot \overrightarrow{z}$ does not contain $z_k = a^{q+1-k}$.
3) Let $\omega^{*'} = \omega^* - \{j\}$, $\mathcal{B}$ first randomly chooses two polynomials $f_0(x), f_1(x)$ of degree m and computes two polynomials as follows:

$$u_0(x) = x^{m-|\omega^*|} \prod_{i \in \omega^*} (x - i),$$
$$u_1(x) = x^{m-|\omega^{*'}|} \prod_{i \in \omega^{*'}} (x - i).$$

Let $c_{0,i}$ and $c_{1,i}$ be the ith term of f_0 and f_1, $d_{0,i}$ and $d_{1,i}$ be the ith term of u_0 and u_1 for $i = 0$ to m. $\mathcal{B}$ then defines $T_0(x) = g^{a \cdot u_0(x) + f_0(x)}, T_1(x) = g^{a \cdot u_1(x) + f_1(x)}$ and simulates $\{t_{0,i}, t_{1,i}\}_{i=0,1,...,m}$ as follows:

$$t_{0,i} = (g^a)^{d_{0,i}} g^{c_{0,i}}, t_{1,i} = (g^a)^{d_{1,i}} g^{c_{1,i}}.$$

Then $\mathcal{B}$ gives the public parameters $\mathbf{pk} = (g, e(g,g)^{\alpha}, \{t_{0,i}, t_{1,i}\}_{i=0,1,...,m}, \overrightarrow{H} = (h_1, ..., h_n)^{\top})$ to the adversary $\mathcal{A}$. Note that, this public key has an identical distribution to that in the actual construction.

Phase1. At any time, the adversary $\mathcal{A}$ may make a private key extraction query of user ID with an LSSS access structure (M, ρ). Let M be a $p \times l$ matrix. such that $\omega^{*'}$ doesn't satisfy the access structure, where $\omega^{*'} = \omega^*$ if $ID \notin R_j^*$, or $\omega^{*'} = \omega^* - \{j\}$ if $ID \in R_j^*$. The simulator $\mathcal{B}$ acts as follows to generate the private key $SK_{ID,(M,\rho)}$:

1) When $ID \notin R_j^*$(in this case, we have $\omega^{*'} = \omega^*$): As ω^* doesn't satisfy the access structure, $\mathcal{B}$ first finds a vector $\overrightarrow{\theta} = (\theta_1, ..., \theta_l)^{\top} \in \mathbb{Z}_p^{n^*}$ such that $\theta_1 = 1$ and for all i where $\rho(i) \in \omega^*$ we have that $M_i \cdot \overrightarrow{\theta} = 0$. Then $\mathcal{B}$ chooses two random vectors $\overrightarrow{\eta_0} = (r, \eta_{0,2}, ..., \eta_{0,l})^{\top}, \overrightarrow{\eta_1} = (0, \eta_{1,2}, ..., \eta_{1,l})^{\top}$ and defines two vectors as follows:

$$\overrightarrow{u_0} = \alpha_1 \overrightarrow{\eta_0} + \alpha \overrightarrow{\theta}, \overrightarrow{u_1} = \overrightarrow{\eta_1} + \alpha \overrightarrow{\theta}.$$

Notice that, the first term of $\overrightarrow{u_0}$ and $\overrightarrow{u_1}$ are $\alpha + r\alpha_1$ and α.

a) For $\rho(i) \in \omega^*$, $\mathcal{B}$ first computes:

$$g^{\lambda_{i,0}} = g^{\overrightarrow{M_i} \cdot \overrightarrow{u_0}} = (g^{\alpha_1})^{(\overrightarrow{M_i} \cdot \overrightarrow{\eta_0})}, g^{\lambda_{i,1}} = g^{(\overrightarrow{M_i} \cdot \overrightarrow{\eta_1})}$$

$\mathcal{B}$ then randomly chooses $r_{i,0}, r_{i,1} \in \mathbb{Z}_p$ and computes:

$$D_{1,0}^i = g^{\lambda_{i,0}} T_0(\rho(i))^{r_{i,0}}, D_{2,0}^i = g^{r_{i,0}},$$

$$D_{1,1}^i = g^{\lambda_{i,1}} T_1(\rho(i))^{r_{i,1}}, D_{2,1}^i = g^{r_{i,1}}.$$

b) For $\rho(i) \notin \omega^*$, $\mathcal{B}$ first computes:

$$g^{\lambda_{i,0}} = g^{\overrightarrow{M_i} \cdot \overrightarrow{u_0}} = g^{\alpha_1 \overrightarrow{M_i} \cdot \overrightarrow{\eta_0} + \alpha \overrightarrow{M_i} \cdot \overrightarrow{\theta}}$$

$$g^{\lambda_{i,1}} = g^{\overrightarrow{M_i}\cdot\overrightarrow{u_1}} = g^{\overrightarrow{M_i}\cdot\overrightarrow{\eta_1}+\alpha\overrightarrow{M_i}\cdot\overrightarrow{\theta}}$$

$\mathcal{B}$ randomly chooses $r'_{i,0}, r'_{i,1} \in \mathbb{Z}_p$ and sets $r_{i,0} = r'_{i,0} - \frac{a^q}{u_0(\rho(i))}(\overrightarrow{M_i}\cdot\overrightarrow{\theta}), r_{i,1} = r'_{i,1} - \frac{a^q}{u_1(\rho(i))}(\overrightarrow{M_i}\cdot\overrightarrow{\theta})$, then it can computes:

$$\begin{aligned} D^i_{1,0} &= g^{\lambda_{i,0}} T_0(\rho(i))^{r_{i,0}} \\ &= g^{\alpha_1\overrightarrow{M_i}\cdot\overrightarrow{\eta_0}+\alpha'\overrightarrow{M_i}\cdot\overrightarrow{\theta}} \cdot \\ &\quad T_0(\rho(i))^{r'_{i,0}} \cdot (g^{a^q})^{\frac{-f_0(\rho(i))(\overrightarrow{M_i}\cdot\overrightarrow{\theta})}{u_0(\rho(i))}} \end{aligned}$$

$$\begin{aligned} D^i_{1,1} &= g^{\lambda_{i,1}} T_1(\rho(i))^{r_{i,1}} \\ &= g^{\overrightarrow{M_i}\cdot\overrightarrow{\eta_1}+\alpha'\overrightarrow{M_i}\cdot\overrightarrow{\theta}} \cdot \\ &\quad T_1(\rho(i))^{r'_{i,1}} \cdot (g^{a^q})^{\frac{-f_1(\rho(i))(\overrightarrow{M_i}\cdot\overrightarrow{\theta})}{u_1(\rho(i))}} \end{aligned}$$

$$D^i_{2,0} = g^{r_{i,0}} = g^{r'_{i,0} - \frac{a^q}{u_0(\rho(i))}(\overrightarrow{M_i}\cdot\overrightarrow{\theta})}$$

$$D^i_{2,1} = g^{r_{i,1}} = g^{r'_{i,1} - \frac{a^q}{u_1(\rho(i))}(\overrightarrow{M_i}\cdot\overrightarrow{\theta})}$$

Finally, $\mathcal{B}$ computes:

$$D_3 = g^r, K = \{K_i = (h_1^{-\frac{x_i}{x_1}} \cdot h_i)^r\}_{i\in[2,\ldots,n]}.$$

2) When $ID \in R^*_j$(in this case, we have $\omega^{*'} = \omega^* - \{j\}$), let $\{ID = ID_k\}_{k\in[1,m]}$: $\mathcal{B}$ first chooses random $r' \in \mathbb{Z}_p$, and sets $r = r' - a^k$. With the definition of $\overrightarrow{\alpha} = B\cdot\overrightarrow{z} + \overrightarrow{\delta}$, the first coordinate of $\overrightarrow{\alpha}$ equals $\alpha_1 = \delta_1 + \sum_{j=1}^m a^{q+1-j}$, so the simulator $\mathcal{B}$ can compute:

$$\begin{aligned} g^{\alpha+r\alpha_1} &= g^{\alpha'+a^{q+1}} \cdot (g^{\delta_1+\sum_{j=1}^m a^{q+1-j}})^{r'-a^k} \\ &= g^{\alpha'-\delta_1 a^k} \cdot g^{\alpha_1 r'} \cdot g^{-(\sum_{j=1,j\neq k}^m a^{q+1-j+k})} \end{aligned}$$

It then chooses random $\{\eta_i\}_{i\in[2,\ldots,l]} \in \mathbb{Z}_p$ and defines vector $\overrightarrow{\eta} = (\alpha + r\alpha_1, \eta_2, \ldots, \eta_l)^\top$. For $i \in [1,2,\ldots,p]$, let $\overrightarrow{M_i} = (x_{i,1}, x_{i,2}, \ldots, x_{i,l})$, $\mathcal{B}$ compute:

$$g^{\lambda_{i,0}} = g^{\overrightarrow{M_i}\cdot\overrightarrow{\eta}} = (g^{\alpha+r\alpha_1})^{x_{i,1}} g^{\sum_{j=2}^l \eta_j\cdot x_{i,j}}.$$

Then $\mathcal{B}$ chooses random $r_{i,0}$ and computes:

$$D^i_{1,0} = g^{\lambda_{i,0}} T_0(\rho(i))^{r_{i,0}}, D^i_{2,0} = g^{r_{i,0}}.$$

As $\omega^{*'}$ doesn't satisfy the access structure, the simulation for $D^i_{1,1}, D^i_{2,1}$ is exactly the same as the previous case(when $ID \notin R^*_j$).

Finally, $\mathcal{B}$ can compute $D_3 = g^r = g^{r'-a^k}$. For $\{K_i\}_{i\in[2,n]}$, as we know $K_{\mathbf{X}} = (K_2, \ldots, K_n) = g^{r\cdot M_{\mathbf{X}}^\top \overrightarrow{\alpha}}$, the simulator can also compute it from available values since $M_{\mathbf{X}}^\top\overrightarrow{\alpha} = M_{\mathbf{X}}^\top \cdot B\cdot\overrightarrow{z} + M_{\mathbf{X}}^\top\cdot\overrightarrow{\delta}$ is independent of $z_k = a^{q+1-k}$(for each $k \in [1,m]$, the kth column of $M^\top_{X_k}\cdot B \in (\mathbb{Z}_p)^{(n-1)\times q}$ is $\overrightarrow{0}$, so that $M^\top_{X_k}\cdot B\cdot\overrightarrow{z}$ does not contain $z_k = a^{q+1-k}$), so no term a^{q+1} appears in the exponent in $K_{\mathbf{X}}$.

Challenge. The adversary $\mathcal{A}$ chooses two challenge messages $M_0, M_1 \in \mathbb{G}_2$ with equal length and sends to $\mathcal{B}$. $\mathcal{B}$ randomly chooses a bit $b \in \{0,1\}$, and creates $C = \mathcal{M}_b Z\cdot e(g^s, g^{\alpha'})$ and $C_1 = g^s$. It then computes:

$$C_{2,0} = \{C^x_{2,0} = T_0(x)^s = (g^s)^{f_0(x)}\}_{x\in\omega^*}$$

$$C_{2,1} = \{C^x_{2,1} = T_1(x)^s = (g^s)^{f_1(x)}\}_{x\in\omega^*-j}$$

For C_3, the simulator $\mathcal{B}$ first defines $\overrightarrow{Y} = (y_1, \ldots, y_n)^\top$ according to the revoked set R^*_j, it must satisfy $< \overrightarrow{X_k}, \overrightarrow{Y} >= 0$ for $k = 1$ to m. This amounts to say that $\overrightarrow{Y} = M_{X_k}\cdot\overrightarrow{\gamma}$, where $\overrightarrow{\gamma} = (y_2, \ldots, y_n)^\top$, for each $k \in [1,m]$. So we have:

$$< \overrightarrow{Y}, B\cdot\overrightarrow{z} >= \overrightarrow{Y}^\top \cdot B\cdot\overrightarrow{z} = \sum_{k=1}^m z_k\cdot\overrightarrow{Y}^\top\cdot\overrightarrow{b_k} = 0$$

Therefore, the challenge can compute C_3 as:

$$C_3 = (h_1^{y_1}\ldots h_n^{y_n})^s = (g^s)^{<\overrightarrow{Y},\overrightarrow{\alpha}>} = (g^s)^{<\overrightarrow{Y},\overrightarrow{\delta}>}$$

Then $\mathcal{B}$ sends the challenge ciphertext $\mathbf{ct}^* = \{C, C_1, C_{2,0}, C_{2,1}, C_3\}$ to the adversary $\mathcal{A}$. If $\mu = 0$ then $Z = e(g_1, g_q)^s$, the challenge ciphertext $\mathbf{ct}^*$ is a valid random encryption of message M_b. If $\mu = 1$, then Z is a random element of $\mathbb{G}_2$, so $\mathbf{ct}^*$ is also a random element of $\mathbb{G}_2$ from the adversary's view and contains no information of M_b.

Phase2. Phase1 is repeated.

Guess. The adversary outputs the guess b' of b. The simulator then outputs $\mu' = 0$ to guess that $Z = e(g_1, g_q)^s$ if $b' = b$; otherwise, it outputs $\mu' = 1$ to indicate that it believes Z is a random group element in $\mathbb{G}_2$.

Then we analysis the advantage of $\mathcal{B}$ to solve the Decision q-BDHE problem is $\frac{1}{2}Pr[\mu' = \mu|\mu = 0] + \frac{1}{2}Pr[\mu' = \mu|\mu = 1] - \frac{1}{2} = \frac{1}{2}(\frac{1}{2} + \epsilon) + \frac{1}{2}\cdot\frac{1}{2} - \frac{1}{2} = \frac{\epsilon}{2}$. This concludes the proof of Theorem 1.