Ciphertext-Policy Attribute-Based Encryption with User Revocation Support

A. Balu[1] and K. Kuppusamy[2]

[1] Department of Computer Science and Engg.,
Alagappa University, Karaikudi
[2] Department of Computer Science and Engg.,
Alagappa University, Karaikudi
balusuriya@yahoo.co.in, kkdiksamy@yahoo.com

Abstract. In Ciphertext-Policy Attribute-Based Encryption(CP-ABE) schemes, the encryptor may set the policy in such a way that who can decrypt the encrypted message. The policy may be formed with the help of attributes. Recent CP-ABE schemes are constructed based on Linear Secret Sharing Scheme. In this paper, we use the Linear Integer Secret Sharing Scheme (LISS) for the construction. Lewko et al.[7] proposed a direction revocation method, based on that we then present a construction of CP-ABE scheme with the ability to do the direct revocation of user. The proposed construction is selectively secure under Decision Bilinear Diffe-Hellman assumption.

Keywords: Attribute-Based Encryption, Linear Integer Secret Sharing, Revocation.

1 Introduction

Attribute-Based Encryption(ABE) has a significant advantage over the traditional PKC primitives as it achieves flexible one-to-many encryption instead of one-to-one. ABE is envisioned as an important tool for addressing the problem of secure and fine-grained data sharing and access control. In an ABE system, a user is identified by a set of attributes. In their seminal paper [10] use biometric measurements as attributes in the following way. A secret key based on a set of attributes ω, can decrypt a ciphertext encrypted with a public key based on a set of attributes $\omega^{'}$,only if the sets ω and $\omega^{'}$ overlap sufficiently as determined by a threshold value t. There are two variants of ABE: Key-Policy Based ABE (KP-ABE) and Ciphertext Policy Based ABE(CP-ABE)[4]. In KP-ABE, the ciphertext is associated with a set of attributes and the secret key is associated with the access policy. The encryptor defines the set of descriptive attributes necessary to decrypt the ciphertext. The trusted authority who generates the user's secret key defines the combination of attributes for which the secret key can be used. In CP-ABE, the idea is reversed: now the ciphertext is associated with the access policy and the encrypting party determines the policy under which the data can be decrypted, while the secret key is associated with a set of attributes.

K. Singh, A.K. Awasthi, and R. Mishra (Eds.): QSHINE 2013, LNICST 115, pp. 696–705, 2013.

1.1 Motivation

Up to date, in most of CP-ABE schemes, the secret exponent s is shared by Linear Secret sharing scheme. Waters [3] proposed three CP-ABE schemes, which are based on Linear Secret Sharing Scheme(LSSS). In 2006, Damgard et al. [6] introduced the notion of Linear Integer Secret Sharing (LISS) scheme. In LISS, the access structure is expressed using AND, OR operators. The following is the advantages of LISS over LSSS.

1. The computations in LISS are done directly over the Integer, while LSSS is done over a finite field.
2. In LISS, the secret sharing cost is less.

In LISS, it is possible to represent the threshold access policy. Using these advantages, it is very easy to express the access policy effectively and share the secret exponent s efficiently in our CP-ABE construction.

1.2 Our Contribution

We present a new scheme for constructing a CP-ABE based on Linear Integer Secret Sharing Scheme (LISS), which allows to do the direct revocation. In this scheme, the access policy $\mathscr{P}$will be expressed using AND, OR operators. We assign a unique id for each user. A ciphertext will be encrypted such that a certain set S = $\{Id_1, ..., Id_r\}$ will be revoked from decrypting it. If the user's private key satisfies the access structure $\mathscr{P}$ and the $ID \notin S$ then the algorithm will decrypt the ciphertext and return the original message. We prove that the proposed scheme is selectively secure under the decisional bilinear Diffie-Hellman assumption.

1.3 Related Work

CP-ABE. The first ciphertext policy ABE was proposed by Bethencourt et al. [4] uses a threshold secret sharing to enforce the policy in the encryption phase. This method requires polynomial interpolation to reconstruct the secret and secure in the generic group model. The CP-ABE proposed by Cheung and Newport [5], in which decryption policies are restricted to a single AND gate, but attributes are allowed to be either positive or negative. In this method, the size of the ciphertext and secret key increases linearly with the total number of attributes in the system. Water's [3] presented three constructions, which are based Linear Secret Sharing Scheme (LSSS) and secure under various difficulty assumptions. Ciphertext and public parameters size were increased in the DBDH assumption [3]. The scheme in [8] is fully secure in the standard model.

Direct Revocation scheme. Direct revocation enforces revocation directly by the sender who specifies these revocation lists while encrypting. An advantage of the direct method is that it does not require key the update phase for all non

revoked users interacting with the key authority. Ostrovsky et al. [9] showed a connection between revocation schemes and achieving non-monotonic access formulas in ABE, to negate an attribute in an access formula one applies a revocation scheme using the attribute as an identity to be revoked. Lewko et al. [7] proposed the direct revocation method based on a new "two equation " technique for revoking users. Attrapadung et al. [2] proposed Broadcast ABE for Key-Policy and Ciphertext-Policy to support direct revocation mechanism. Attrapadung et al. [1] proposed another construction, which supports direct and indirect revocation in a single scheme.

2 Preliminaries

2.1 Access Structures

Definition 1. (Access structure) *Let* $\{1, 2, ...n\}$ *be a set of parties. A collection* $\Gamma \subseteq 2^{\{1,2,...,n\}}$ *is monotone if* $\forall B, C$: *if* $B \in \Gamma$ *and* $B \subseteq C$ *then* $C \in \Gamma$. *An access structure(respectively, monotone access structure) is a collection (respectively, monotone collection) A of non-empty subsets of* $\{1, 2..., n\}$ *i.e* $\Gamma \subseteq 2^{\{1,2,..,n\}} \setminus \phi$. *The sets in* Γ *are called the authorized sets, and the sets not in* Γ *are called the unauthorized sets.*

2.2 Linear Integer Secret Sharing

In the LISS scheme, the secret is an integer chosen from a (publically known) interval, and each share is computed as an integer linear combination of the secret and some random numbers chosen by the dealer. Reconstruction of the secret is done by computing a linear combination with integer coefficients of the shares in a qualified set. Let $P = \{1, 2, ..n\}$ denote the n share holders and D the dealer. Let Γ be a monotone access structure on P. Let ℓ be an integer constant. The dealer D wants to share a secret s from the publically known interval $\left[-2^{\ell}, 2^{\ell}\right]$ to the shareholders P over Γ, such that every set of shareholders $A \in \Gamma$ can reconstruct s, but a set of shareholders $A \notin \Gamma$ gets no or little information on s.

We say that a subset $A \subseteq P$ is qualified if the parties in A jointly are allowed to reconstruct the secret s. In a LISS scheme, the shares consist of a collection of integers,$\{s_i\}_{i \in I}$, where for each $i \in I$, the integer s_i belongs to exactly one party and s_i is computed by a linear integer combination of s and some randomness chosen by the dealer. Given a qualified subset of shares $\{s_i\}_{i \in I'}$, then the secret can be reconstructed by a linear combination $s = \sum_{i \in I'} \lambda_i s_i$, where $\{\lambda_i\}_{i \in I'}$ are integer coefficients that are determined by the index I'. We use a distribution matrix $M \in Z^{d \times e}$ and a corresponding surjective function $\Psi : \{1, ..d\} \rightarrow P$. We say that the i-th row is labeled by $\Psi(i)$ or owned by party $P_{\Psi(i)}$. We use a distribution vector $\rho = (s, \rho_2, ..., \rho_e)$ where s is the secret, and the $\rho_i' s$ are uniformly random chosen integers in $\left[-2^{\ell_0+k}, 2^{\ell_0+k}\right]$, where k is the security parameter and ℓ_0 is a constant. The dealer D calculates shares by

$$M.\rho = (s_1, ..., s_d)^T \tag{1}$$

where we denote each s_i as a share unit for $1 \leq i \leq d$. The i'th share unit is then given to the $\Psi(i)'th$ shareholder. If $A \subseteq P$ is a set of shareholders, then M_A denotes the restriction of M rows jointly owned by A.

2.3 Integer Span Program

Definition 2. *$\mathscr{M} = (M, \Psi, \xi)$ is called an Integer Span Program(ISP) if $M \in Z^{d \times e}$ and the d rows of M are labeled by a surjective function $\Psi : \{1, ..d\} \to P$. Finally $\xi = (1, 0, 0, ...0)^T \in Z^e$ is called the target vector. We define $size(\mathscr{M}) = d$, where d is the number of rows of M.*

Definition 3. *Let Γ be a monotone access and let$\mathscr{M} = (M, \Psi, \xi)$ be a integer span program. Then $\mathscr{M}$ is an ISP, if for all $A \subseteq \{1, .., n\}$ the following holds.*
1. If $A \in \Gamma$, then there is a vector $\lambda \in Z^d$ such that $M_A^T \lambda = \xi$.
2. If $A \notin \Gamma$, then there exists $\mathbf{k} = (k_1, ..., k_e)^T \in Z^e$ such that $M_A \cdot \mathbf{k} = \mathbf{0} \in Z^d$ with $k_1 = 1$, which is called the sweeping vector for A.

If we have an ISP $\mathscr{M} = (M, \Psi, \xi)$ which computes Γ, we build a LISS scheme for Γ as follows: We use M as the distribution matrix and
$\ell_0 = \ell + \lceil \log_2(k_{max}(e-1)) \rceil + 1$, where ℓ is the length of the secret and
$k_{max} = max\{|a|/a$ is an entry in some sweeping vector $\}$.

Secret Reconstruction. An authorized set A can computes the secret by taking a linear combination of their values, since there exists $\lambda_A \in Z^{d_A}$ such that $M_A^T \cdot \lambda_A = \xi$ (as per Definition 5).
With the secret shares of s_A it is justified to reconstruct the secret s by the following way

$$s_A^T \cdot \lambda_A = (M_A \cdot \rho)^T \cdot \lambda_A (\text{by eqn (1)})$$

$$= \rho^T \cdot (M_A^T \cdot \lambda_A) = \rho^T \cdot \xi = s$$

2.4 Bilinear Maps

Let G_0, G_1, be two multiplicative cyclic groups of prime order p. Let g be a generator of G_0 . Let e be a bilinear map, $e : G_0 \times G_0 \to G_1$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in G_0$, and $a, b \in \mathbb{Z}_p^*$,

we have $e\left(u^a, v^b\right) = e\left(u, v\right)^{ab}$

2. Non-degeneracy: $\hat{e}\left(g, g\right) \neq 1$. -

The map e is symmetric since $e\left(g^a, g^b\right) = e\left(g, g\right)^{ab} = e\left(g^b, g^a\right)$

2.5 Decisional Bilinear Diffie- Hellman Assumption

We define the Decisional Bilinear Diffie-Hellman problem as follows. A challenger chooses a group G_0 is of prime order p according to the security parameter. Let

$a, b, s \in \mathbb{Z}_p^*$ be chosen at random and g be a generator G_0 . The adversary given $\left(g, g^a, g^b, g^s\right)$ must distinguish a valid tuple $e\left(g,g\right)^{abs} \in G_1$ from a random element R in G_1. An algorithm A that outputs {0,1} has the advantage ϵ in solving decisional BDH in G_0 if

$$Pr\left[\mathcal{A}\left(g, g^a, g^b, g^s, D = e\left(g,g\right)^{abs}\right)\right) = 0] - Pr\left[\mathcal{A}\left(g, g^a, g^b, g^s, D = R\right) = 0\right] \geq \epsilon$$

2.6 Ciphertext Policy Attribute Based Encryption with User Revocation(CP-ABE-UR)

A cipher text policy attribute based encryption scheme consists of four fundamental algorithms: Setup, Key Generation, Encryption and Decryption.

Setup. The setup algorithm takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK.

KeyGen (MK,PK, L, ID). The key generation algorithm takes as input the master key MK, public key PK, an identity ID and the attribute list L . It outputs a private key SK_L for the attribute list L .

Encrypt (S, PK, $\mathscr{P}$, m). The encryption algorithm takes as input a revocation set S of identities, public parameters PK, the message m, and an access policy $\mathscr{P}$ over the universe of attributes. The algorithm will encrypt m and produce a ciphertext CT such that any user with a key for an identity $ID \notin S$ and the attribute list L satisfies the access policy can decrypt.

Decrypt(CT,SK_L, ID,S). The decryption algorithm takes as input the ciphertext CT that was generated for the revoked set S, as well as an identity and a private key SK_L for the attribute list L. If the list L of attributes satisfies the access policy $\mathscr{P}$ and the $ID \notin S$ then the algorithm will decrypt the ciphertext and return a message M.

2.7 Security Model for CP-ABE-UR

The selective security notion for CP-ABE-UR is defined in the following game.

Init. The adversary chooses the target set S^* and the challenge access policy $\mathscr{P}^*$, gives it to the challenger.

Setup. The challenger runs the Setup algorithm and gives the public parameters, PK to the adversary.

Phase1. The adversary makes a secret key request to the Keygen oracle for any attribute list L , user index ID, with the restriction that $ID \notin S^*, L \nvDash \tau^*$. The Challenger returns Keygen(L, MK, ID, PK).

Challenge. The adversary submits two equal length messages M_0 and M_1. The Challenger flips a random coin d, and encrypts M_d under $(\mathscr{P}^*, S^*)$. The ciphertext CT^* is given to the adversary.

Phase 2. The adversary can continue querying Keygen with the same restriction as during Phase1.

Guess. The adversary outputs a guess $d^{'}$ of d.

Definition 4. *A ciphertext-policy attribute based encryption scheme is said to be secure against chosen-plaintext attack(CPA) if any polynomial time adversaries have only a negligible advantage in the IND-CPA game, where the advantage is defined to be* $\epsilon = \left| Pr[d^{'} = d] - \frac{1}{2} \right|$.

3 Main Construction

In this section, first we specify the method to form the access policy matrix M and then the construction of the CP-ABE-UR scheme.

3.1 Formation of Access Policy Matrix M

The access policy specified by the encryptor can be represented by a LISS matrix M by the following procedure.

Let $M_u \in Z^{1\times 1}$ be the matrix with single entry which is one, i.e $M_u = 1$.

If we have a matrix $M_a \in Z^{d_a \times e_a}$ then we can form $c_a \in Z^e$ to represent the first column in M_a and $R_a \in Z^{(d_a-1)\times e_a}$ to represent all but the first column in M_a.

Given any access policy $\mathscr{P}$, we can construct the distribution matrix M by using the following rules.

Rule 1. Each variable a_i in the access policy $\mathscr{P}$can be expressed by M_u.

Rule 2. For any OR-term $\mathscr{P} = \mathscr{P}_a \bigvee \mathscr{P}_b$. Let $M_a \in Z^{d_a \times e_a}$ and $M_b \in Z^{d_b \times e_b}$ be the matrices which expresses the formulas $\mathscr{P}_a$ and $\mathscr{P}_b$ respectively. We can construct a matrix $M_{OR} \in Z^{(d_a+d_b)(e_a+e_b-1)}$ expressing $\mathscr{P}$, which is defined by letting the first column of M_{OR} be the concatenation of the two column vectors c_a and c_b, then letting the following $d_a - 1$ columns be the columns of R_a expanded with e_b succeeding zero entries, and the last $d_b - 1$ columns be the columns of R_b expanded with e_a leading zero entries. This is visualized by

$$M_{OR} = \begin{array}{|c|c|c|} \hline c_a & R_a & 0 \\ \hline c_b & 0 & R_b \\ \hline \end{array}$$

Rule 3. For any AND-term $\mathscr{P} = \mathscr{P}_a \bigwedge \mathscr{P}_b$. Let $M_a \in Z^{d_a \times e_a}$ and $M_b \in Z^{d_b \times e_b}$ be the matrices which expresses the formulas $\mathscr{P}_a and \mathscr{P}_b$ respectively. We can construct a matrix $M_{AND} \in Z^{(d_a+d_b)(e_a+e_b)}$ which expresses the access policy $\mathscr{P}$. It is defined by letting the first column of M_{AND} be the column vector c_a expanded with e_b succeeding zero entries, the next column to be the concatenation of c_a and c_b the following $d_a - 1$ columns be the columns of R_a expanded with e_b succeeding zero entries, and the last $d_b - 1$ columns be the columns of R_b expanded with e_a leading zero entries. This is visualized by

$$M_{AND} = \begin{array}{|c|c|c|c|} \hline c_a & c_a & R_a & 0 \\ \hline 0 & c_b & 0 & R_b \\ \hline \end{array}$$

3.2 CP-ABE-UR Scheme

Setup (1^k). The setup algorithm chooses a group G_0 of prime order p and a generator g.
Let $A = \{a_1, a_2, ..., a_n\}$ be the set of attributes.
For each attribute a_i, it chooses random element $t_i \in Z_p$, and computes $T_i = g^{t_i}\{1 \leq i \leq n\}$
Let $y = e(g,g)^{\alpha}$ where $\alpha \in Z_p$. Let b be a random element $\in Z_p$.
The Public Key is PK $= \left(g, g^b, y, \{T_i; 1 \leq i \leq n\}\right)$ and the Master Secret Key is MK $= (\alpha, b, t_i\{1 \leq i \leq n\})$.

KeyGen (ID,MK,PK,L). This algorithm takes as input the user index ID, master secret key, public key and the attribute list of the user and performs the following:

a) Select random values $a, r, \omega \in Z_p$
$d_0 = g^{\alpha} g^{ar} g^{b\omega}$; $d_2 = g^{\omega}$; $d_3 = (g^{bID} g)^{\omega}$
For each attribute in the attribute list L, $d_i^* = g^{art_i^{-1}}$
The secret key is $SK_L = (d_0, d_2, d_3, \forall a_i \in L : d_i^*)$

Encrypt(S,PK, $\mathscr{P}$, m). The encryption algorithm takes as input a user index set $S = \{ID_1, .., ID_r\}$, the public key PK , a message $m \in G_1$ to encrypt and the access policy $\mathscr{P}$.

Step 1. Select a random element $s \in \left[-2^{\ell}, 2^{\ell}\right]$ and compute $C_0 = g^s$.
M is the distribution matrix constructed by the above method for the access policy $\mathscr{P}$. Choose $\rho = (s, \rho_2, ..., \rho_e)^T$, where $\rho_i' s$ are uniformly random chosen integers in $\left[-2^{\ell_0+k}, 2^{\ell_0+k}\right]$.

Step 2.

a) Computes $M \cdot \rho = (s_1, ..., s_d)^T$

b) $C_1 = m \cdot y^s = m \cdot e(g,g)^{\alpha s}$; $C_k^{'} = g^{s_k}$; $C_k^{+} = (g^{bID_k} g)^{s_k}$;k= 1 to r

c) For each attribute in $\mathscr{P}$, compute $C_i^* = T_i^{s_i}$ using the corresponding shares of the attribute a_i.

The ciphertext is published as $CT = \left(C_0, C_1, C_i^*, C_k^{'}, C_k^{+}\right)$.

Decrypt(CT,SK_L,ID,S). The decryption algorithm takes as input the ciphertext CT that was generated for the revoked set S, as well as an identity and a private key SK_L for the attribute list L. If the list L of attributes satisfies the access policy $\mathscr{P}$ and the $ID \notin S$ then there is a vector $\lambda_L \in Z^{d_L}$ such that $M_L^T \lambda_L = \xi$ (as per definition 3). With this, it is possible to reconstruct the secret using $\sum_{i \in L} \lambda_i s_i = s$.

The decryption algorithm computes E= $\frac{e(C_0,d_0)}{\prod_{i \in L} e(C_i^*,(d_i^*)^{\lambda_i})} \prod_{k=1}^{r} \left[\frac{e(d_2,C_k^{+})}{e(d_3,C_k^{'})}\right]^{\frac{\lambda_k}{ID-ID_k}}$

$$= e(g,g)^{\alpha s}$$

where it can compute since $ID \neq ID_k$ for k = 1,..,r , then it computes

$$m = C_1 / E.$$

3.3 Security Analysis

Theorem 1. Suppose the decisional BDH assumption holds, then no polynomial adversary can selectively break our system.

Proof: Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage ϵ in the selective security game against our construction. We show how to use the adversary $\mathcal{A}$ to build a simulator $\mathcal{B}$ that is able to solve the DBDH assumption. The Challenger gives the simulator $\mathcal{B}$ the DBDH challenge : $(g, A, B, C, D) = \left(g, g^a, g^b, g^s, D\right)$.

Init. The adversary chooses the challenge access policy $(M^{'}, \mathscr{P}^*)$, a revocation set $S^* = \{Id_1, Id_2, .., Id_r\}$ and gives it to the simulator.

Setup. The simulator selects at random $a^{'} \in Z_p$ and implicitly sets $\alpha = ab + a^{'}$ by letting $e(g,g)^{\alpha} = e(g^a, g^b) e(g,g)^{a^{'}}$. For all $a_j \in U$ it chooses a random $q_j \in Z_p$ and set $T_j = g^{\frac{1}{(M_{i,j}^{'} q_j)}}$ if $a_j \notin \mathscr{P}^*$, otherwise $T_j = g^{q_j}$. The simulator $\mathcal{B}$ sends the public parameters to $\mathcal{A}$.

Phase 1. In this phase the simulator answers private key queries. Suppose the simulator is given a private key for a list L where L does not satisfy $\mathscr{P}^*$ and $ID \notin S^*$. On each request $\mathcal{B}$ chooses a random variable $v, \delta \in Z_p$, and finds

a vector $\mathbf{k} = (k_1, k_2, .., k_e)^T \in Z^e$ such that $M^{'} \cdot \mathbf{k} = \mathbf{0}$ with $k_1 = 1$. By the definition of Sweeping vector such a vector must exist. The simulator sets $r = v - k_j b$ and computes
$d_0 = g^{\alpha} g^{ar} g^{b\omega} = g^{ab+a^{'}} g^{a(v-k_j b)} g^{b\delta} = g^{a^{'}} A^v g^{b\delta}$
In calculating d_i^* we have the term $M_{i,j}^{'} a \cdot k_j b$ get canceled because of $M^{'} \cdot \mathbf{k} = \mathbf{0}$
$d_i^* = g^{(v-k_j b) a q_j M_{i,j}^{'}} = A^{v M_{i,j}^{'} q_j}$
$d_2 = g^{\delta}$ $d_3 = (g^{bID} g)^{\delta}$

Challenge. $\mathcal{A}$ submits two messages $m_0, m_1 \in G_1$. The simulator flips a fair binary coin d, and returns the encryption of m_d. The encryption of m_d can be done as follows:
$C_0 = g^s, C_1 = m_d De(g^s, g^{a^{'}})$
The simulator will choose uniformly random integers $z_2, ..., z_h \in \left[-2^{\ell_0+k}, 2^{\ell_0+k}\right]$ and share the secret $x$using the vector $\Phi = (x, z_2, ..., z_h)$.
Create the distribution matrix M, for the access policy $\mathscr{P}^*$. Compute $M \cdot \Phi$ and use the shares to encrypt the access policy with corresponding q_j for the attributes present in the access policy $\mathscr{P}^*$, $C_i^* = T_i^{x_i}$, $C_i^{'} = g^{x_i}, C_i^{+} = (g^{bID_i} g)^{x_i}$

Phase 2. Same as Phase 1.

Guess. $\mathcal{A}$ outputs a guess $d^{'}$ of d. The simulator then outputs 0 to the guesses that $D = e(g,g)^{abs}$ if $d^{'} = d$; otherwise, it outputs 1 to indicate that it believes D is random group element in G_1.

When D is a tuple the simulator $\mathcal{B}$ gives a perfect simulation, so we have that $Pr\left[\mathcal{B}\left(\rho, D = e(g,g)^{abs}\right) = 0\right] = \frac{1}{2} + \epsilon$.

When D is a random group element the message m_d is completely hidden from the adversary, and we have $Pr\left[\mathcal{B}\left(\rho, D = R\right) = 0\right] = \frac{1}{2}$.

4 Conclusion

We constructed the CP-ABE scheme based on a new secret sharing scheme called Linear Integer Secret Sharing Scheme. Our scheme has the ability to do the direct revocation of users. The security of our scheme is provided in the standard model under DBDH assumption.

References

1. Attrapadung, N., Imai, H.: Attribute-Based Encryption Supporting Direct/Indirect Revocation Modes. In: Parker, M.G. (ed.) Cryptography and Coding 2009. LNCS, vol. 5921, pp. 278–300. Springer, Heidelberg (2009)
2. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute-based encryption. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 248–265. Springer, Heidelberg (2009)
3. Waters, B.: Ciphertext policy attribute based encryption: An expressive, efficient, and provably secure realization. In: Cryptology eprint report 2008/290 (2008)

4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext policy attribute based encryption. In: IEEE Symposium on Security and Privacy, pp, pp. 321–334 (2007)
5. Cheung, L., Newport, C.: Provably secure Ciphertext police ABE. In: CCS 2007: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 456–465. ACM Press, New York (2007)
6. Damgård, I.B., Thorbek, R.: Linear Integer Secret Sharing and Distributed Exponentiation. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 75–90. Springer, Heidelberg (2006)
7. Lewko, A., Sahai, A., Waters, B.: Revocation systems with very small private keys. Cryptology eprint report 2009/309 (2009)
8. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (Hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
9. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security, pp. 195–203 (2007)
10. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)