Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization

Brent Waters*

University of Texas at Austin
bwaters@cs.utexas.edu

Abstract. We present a new methodology for realizing Ciphertext-Policy Attribute Encryption (CP-ABE) under concrete and noninteractive cryptographic assumptions in the standard model. Our solutions allow any encryptor to specify access control in terms of any access formula over the attributes in the system. In our most efficient system, ciphertext size, encryption, and decryption time scales linearly with the complexity of the access formula. The only previous work to achieve these parameters was limited to a proof in the generic group model.

We present three constructions within our framework. Our first system is proven selectively secure under a assumption that we call the decisional Parallel Bilinear Diffie-Hellman Exponent (PBDHE) assumption which can be viewed as a generalization of the BDHE assumption. Our next two constructions provide performance tradeoffs to achieve provable security respectively under the (weaker) decisional Bilinear-Diffie-Hellman Exponent and decisional Bilinear Diffie-Hellman assumptions.

1 Introduction

Public-Key encryption is a powerful mechanism for protecting the confidentiality of stored and transmitted information. Traditionally, encryption is viewed as a method for a user to share data to a targeted user or device. While this is useful for applications where the data provider knows specifically which user he wants to share with, in many applications the provider will want to share data according to some policy based on the receiving user's credentials.

Sahai and Waters [35] presented a new vision for encryption where the data provider can express how he wants to share data in the encryption algorithm itself. The data provider will provide a predicate $f(\cdot)$ describing how he wants to share the data and a user will be ascribed a secret key associated with their credentials X; the user with credentials X can decrypt a ciphertext encrypted

* Supported by NSF CNS-0716199, CNS-0915361, and CNS-0952692, Air Force Office of Scientific Research (AFO SR) under the MURI award for "Collaborative policies and assured information sharing" (Project PRESIDIO), Department of Homeland Security Grant 2006-CS-001-000001-02 (subaward 641), a Google Faculty Research award, and the Alfred P. Sloan Foundation.

D. Catalano et al. (Eds.): PKC 2011, LNCS 6571, pp. 53–70, 2011.

with predicate f if $f(X) = 1$. Sahai and Waters [35] presented a particular formulation of this problem that they called Attribute-Based Encryption (ABE), in which a user's credentials is represented by a set of string called "attributes" and the predicate is represented by a formula over these attributes. Several techniques used by SW were inspired by prior work on Identity-Based Encryption [36, 13, 23, 18, 10]. One drawback of the Sahai-Waters approach is that their initial construction was limited to handling formulas consisting of one threshold gate.

In subsequent work, Goyal, Pandey, Sahai, and Waters [27] further clarified the concept of Attribute-Based Encryption. In particular, they proposed two complementary forms of ABE. In the first, Key-Policy ABE, attributes are used to annotate the ciphertexts and formulas over these attributes are ascribed to users' secret keys. The second type, Ciphertext-Policy ABE, is complementary in that attributes are used to describe the user's credentials and the formulas over these credentials are attached to the ciphertext by the encrypting party. In addition, Goyal et al. [27] provided a construction for Key-Policy ABE that was very expressive in that it allowed the policies (attached to keys) to be expressed by *any* monotonic formula over encrypted data. The system was proved selectively secure under the Bilinear Diffie-Hellman assumption. However, they left creating expressive Ciphertext Policy ABE schemes as an open problem.

The first work to explicitly address the problem of Ciphertext-Policy Attribute-Based Encryption was by Bethencourt, Sahai, and Waters [7]. They described an efficient system that was expressive in that it allowed an encryptor to express an access predicate f in terms of any monotonic formula over attributes. Their system achieved analogous expressiveness and efficiency to the Goyal et al. construction, but in the Ciphertext-Policy ABE setting. While the BSW construction is very expressive, the proof model used was less than ideal — the authors only showed the scheme secure in the generic group model, an artificial model which assumes the attacker needs to access an oracle in order to perform any group operations[1].

Recently, ABE has been applied in building a variety of secure systems [34, 40, 9, 8]. These systems motivate the need for ABE constructions that are both foundationally sound and practical.

Ciphertext Policy ABE in the Standard Model. The lack of satisfaction with generic group model proofs has motivated the problem of finding an expressive CP-ABE system under a more solid model. There have been multiple approaches in this direction.

First, we can view the Sahai-Waters[35] construction most "naturally" as Key-Policy ABE for a threshold gate. In their work, Sahai and Waters describe how to realize Ciphertext-Policy ABE for threshold gates by "grafting" so called "dummy attributes" over their basic system. Essentially, they transformed a KP-ABE system into a CP-ABE one with the expressiveness of a single threshold

[1] Alternatively, we could derive a concrete, but interactive and complicated assumption directly from the scheme itself and argue that the scheme is secure under this assumption. However, this view is also not very satisfactory.

gate[2]. Cheung and Newport[22] provide a direct construction for constructing a policy with a single AND gate under the Bilinear Diffie-Hellman assumption. Their approach has the drawbacks that it only allows a fixed number of system attributes and is limited to an AND gate (does not enable thresholds). In retrospect these two limitations actually make it less expressive than the SW transformation, although this wasn't necessarily immediately apparent.

Most recently, Goyal, Jain, Pandey, and Sahai [26] generalized the transformational approach to show how to transform a KP-ABE system into a CP-ABE one using what they call a "universal access tree". In particular, they provided a mapping onto a "universal" (or complete) access tree of up to depth d formulas consisting of threshold gates of input size m, where m and d are chosen by the setup algorithm. They applied a similar "dummy attribute" approach.

In order to accommodate a general access formula of size n, their scheme first translates this into a balanced formula. Under standard techniques a formula of size n can be "balanced" such that any formula (tree) of size n can be covered by a complete tree of size approximately $O(n^{3.42})$. Their work was the first feasibility result for expressive CP-ABE under a non-interactive assumption. Unfortunately, the parameters of ciphertext and private key sizes add encryption and decryption complexity blow up (in the worst case) by an $n^{3.42}$ factor limiting its usefulness in practice. For instance, in a system with U attributes defined and n nodes the ciphertext overhead will be approximately a factor of $U \cdot n^{2.42}$ greater than that of the BSW system. To give a concrete example, for the modest parameters of universe size $U = 100$ attributes and a formula of 20 nodes the blowup factor relative to BSW is approximately $140,000$.

Our Contribution. We present a new methodology for realizing Ciphertext-Policy ABE systems from a general set of access structures in the *standard model* under concrete and non-interactive assumptions. Both the ciphertext overhead and encryption time scale with $O(n)$ where n is the size of the formula. In addition, decryption time scales with the number of nodes.

Our first system allows an encryption algorithm to specify an access formula in terms of any access formula. In fact our techniques are slightly more general. We express access control by a Linear Secret Sharing Scheme (LSSS) matrix M over the attributes in the system. Previously used structures such as formulas (equivalently tree structures) can be expressed succinctly [6] in terms of a LSSS. *We do not lose any efficiency by using the more general LSSS representation as opposed to the previously used tree access structure descriptions.* Thus, we achieve the same performance and functionality as the Bethencourt, Sahai, and Waters construction, but under the standard model.

In addition, we provide two other constructions that tradeoff some performance parameters for provable security under the respective weaker assumptions of decisional-Bilinear Diffie-Hellman Exponent (d-BDHE) and decisional-Bilinear Diffie-Hellman assumptions. In Table 1 we summarize the comparisons between our schemes and the GJPS and BSW CP-ABE systems in terms of ciphertext and

[2] The Sahai-Waters construction was given prior to the Key-Policy and Ciphertext-Policy distinction; our interpretation is a retrospective one.

key sizes and encryption and decryption times. Taken all together our first scheme realizes the same efficiency parameters as the BSW encryption scheme, but under a concrete security assumption. At the same time, our d-BDH construction is proved under the same assumption as the GJPS system and achieves significantly better performance.

Our Techniques. Our techniques provide a framework for *directly* realizing provably secure CP-ABE systems. In our systems, the ciphertext distributes shares of a secret encryption exponent s across different attributes according to the access control LSSS matrix M.

A user's private key is associated with a set S of attributes and he will be able to decrypt a ciphertext iff his attributes "satisfy" the access matrix associated with the ciphertext. As in previous ABE systems, the primary challenge is to prevent users from realizing collusion attacks. Our main tool to prevent this is to randomize each key with an freshly chosen exponent t. During decryption, each share will be multiplied by a factor t in the exponent. Intuitively, this factor should "bind" the components of one user's key together so that they cannot be combined with another user's key components. During decryption, the different shares (in the exponent) that the algorithm combines are multiplied by a factor of t. Ultimately, these randomized shares are only useful to that one particular key.

Our construction's structures and high level intuition for security is similar to the BSW construction. The main novelty in our paper is provide a method for proving security of such a construction. The primary challenge one comes across is (in the selective model) how to create a reduction that embeds a complex access structure in a short number of parameters. All prior ABE schemes follow a "partitioning" strategy for proving security where the reduction algorithm sets up the public parameters such that it knows all the private keys that it needs to give out, yet it cannot give out private keys that can trivially decrypt the challenge ciphertext. In prior KP-ABE schemes the challenge ciphertext was associated with a set S^* of attributes. This structure could fairly easily be embedded in a reduction as the public parameter for each attribute was simply treated differently depending whether or not it was in S^*. In CP-ABE, the situation is much more complicated as ciphertexts are associated with a potentially large access structure M^* that includes attributes multiple times. In general, the size of M^* is much larger than the size of the public parameters[3]. Consequently, there is not a simple "on or off" method of programming this into the parameters. Arguably, it is this challenge that lead the BSW paper to apply the generic group heuristic and GJPS paper to translate the problem back to KP-ABE.

In this paper, we create a method for directly embedding any LSSS structure M^* into the public parameters in our reduction. In the proofs of our system a simulator can "program" the LSSS matrix M^* of the challenge ciphertext (in the selective model of security). Consider a LSSS matrix M^* of size $l^* \times n^*$. For each row i of M^* the simulator needs to program in ℓ pieces of information

[3] Here we roughly mean size to be number of rows in the LSSS system or nodes in an access tree.

$(M^*_{i,1}, \ldots, M^*_{i,\ell})$ into the parameters related to the attribute assigned to that row. In our most efficient system we program in M^* using the d-Parallel BDHE assumption; however, in Section 5 we show variations of our construction that are provably secure using similar ideas, but under weaker assumptions.

Our methodology of creating a system and proof that directly addresses CP-ABE stands in contrast to the approach of GJPS which essentially maps CP-ABE requirements onto a KP-ABE scheme.

Table 1. Comparison of CP-ABE systems in terms of ciphertext size, private key size, encryption and decryption times and assumptions. We let n be the size of an access formula , A be the number of attributes in a user's key, and T be (minimum needed) number of nodes satisfied of a formula by a user's attributes, and U be the number of attributes defined in the system. For our d-BDHE construction of the system defines a parameter $k_{\max}$, which is the maximum number of times a single attribute will appear in a particular formula. In the GJPS construction and our d-BDH one of Section 5 the systems define $n_{\max}$ as a bound on the size any formula. The ciphertext and private key sizes are given in terms of the number of group elements, encryption time in terms of number of exponentiations, and decryption in terms of number of pairing operations.

System	Ciphetext Size	Private Key Size	Enc. Time	Assumption
BSW[7]	$\mathcal{O}(n)$	$\mathcal{O}(A)$	$\mathcal{O}(n)$	Generic Group
GJPS[26]	$\mathcal{O}(U \cdot n_{\max}^{3.42})$	$\mathcal{O}(A \cdot n_{\max}^{3.42})$	$\mathcal{O}(U \cdot n_{\max}^{3.42})$	d-BDH
Section 3	$\mathcal{O}(n)$	$\mathcal{O}(A)$	$\mathcal{O}(n)$	d-Parallel BDHE
Full version [42]	$\mathcal{O}(n)$	$\mathcal{O}(k_{\max} \cdot A)$	$\mathcal{O}(n)$	d-BDHE
Section 5	$\mathcal{O}(n^2)$	$\mathcal{O}(k_{\max} \cdot A + n_{\max})$	$\mathcal{O}(n^2)$	d-BDH

1.1 Related Work

Some of the roots of ABE can be traced back to Identity-Based Encryption [36, 13, 23, 18, 10, 41, 24, 14] (IBE). One can view IBE as a very special case of ABE.

Different authors [38, 32, 4, 17, 3, 5] have considered similar problems without considering collusion resistance. In these works a data provider specifies an access formula such that a group of users can decrypt if the *union* of their credentials satisfies the formula. By only requiring the union of the credentials one does not worry about collusion attacks. In these schemes a setup authority simply assigns a separate public key to each credential and gives the corresponding secret key to each user that possesses the credential. Encryption is done by splitting secrets and then encrypting each share to the appropriate public key. Some of these schemes were inspired by earlier work [21, 20].

Since the introduction of Attribute-Based Encryption by Sahai and Waters [35], there have been several papers [27, 7, 19, 33, 26] that have proposed different varieties of ABE. Most of them have been for monotonic access structures over attributes; one exception is the work of Ostrovsky, Sahai, and Waters [33] that showed how to realize negation by integrating revocation schemes into the GPSW ABE cryptosystem.

Most work on ABE is focused on complex access controls for hiding an encrypted payload of data. A related line of work called predicate encryption or searching on encrypted data attempts to evaluate predicates over the encrypted data itself [39, 12, 1, 16, 15, 37, 29]. These systems have the advantages of hiding the associated access structures themselves and thus providing a level of "anonymity". The concept of predicate encryption is more general than the one we consider. However, the predicate encryption systems realized thus far tend to be much less expressive than access control systems that leave the access structures in the clear.

Other examples of encryption systems with more "structure" added include Hierarchical Identity-Based Encryption [28, 25] and Wildcard IBE [2].

Finally, Lewko et. al. [31] recently leveraged the encoding technique from our work to build an ABE system that achieves adaptive (non-selective) security. The system of Lewko et. al. is based in composite order groups, which results in some loss of practical efficiency compared to our most efficient system. In addition, our BDH system is based off of more standard assumptions than those used in Lewko et al.

2 Background

We first give formal definitions for access structures and relevant background on Linear Secret Sharing Schemes (LSSS). Then we give the security definitions of ciphertext policy attribute based encryption (CP-ABE). Finally, we give background information on bilinear maps.

2.1 Access Structures

Definition 1 (Access Structure [6]). *Let* $\{P_1,P_2,\ldots,P_n\}$ *be a set of parties. A collection* $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ *is monotone if* $\forall B, C$: if $B \in \mathbb{A}$ *and* $B \subseteq C$ *then* $C \in \mathbb{A}$. *An* access structure *(respectively, monotone access structure) is a collection (respectively, monotone collection)* $\mathbb{A}$ *of non-empty subsets of* $\{P_1, P_2, \ldots, P_n\}$, *i.e.,* $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}\setminus\{\emptyset\}}$. *The sets in* $\mathbb{A}$ *are called the* authorized sets, *and the sets not in* $\mathbb{A}$ *are called the* unauthorized sets.

In our context, the role of the parties is taken by the attributes. Thus, the access structure $\mathbb{A}$ will contain the authorized sets of attributes. We restrict our attention to monotone access structures. However, it is also possible to (inefficiently) realize general access structures using our techniques by having the not of an attribute as a separate attribute altogether. Thus, the number of attributes in the system will be doubled. From now on, unless stated otherwise, by an access structure we mean a monotone access structure.

2.2 Linear Secret Sharing Schemes

We will make essential use of linear secret-sharing schemes. We adapt our definitions from those given in [6]:

Definition 2 (Linear Secret-Sharing Schemes (LSSS)). *A secret-sharing scheme Π over a set of parties $\mathcal{P}$ is called* linear *(over $\mathbb{Z}_p$) if*

1. *The shares for each party form a vector over $\mathbb{Z}_p$.*
2. *There exists a matrix an M with ℓ rows and n columns called the share-generating matrix for Π. For all $i = 1, \ldots, \ell$, the i'th row of M we let the function ρ defined the party labeling row i as $\rho(i)$. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then Mv is the vector of ℓ shares of the secret s according to Π. The share $(Mv)_i$ belongs to party $\rho(i)$.*

It is shown in [6] that every linear secret sharing-scheme according to the above definition also enjoys the *linear reconstruction* property, defined as follows: Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$.

Furthermore, it is shown in [6] that these constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generating matrix M.

Note on Convention. We note that we use the convention that vector $(1, 0, 0, \ldots, 0)$ is the "target" vector for any linear secret sharing scheme. For any satisfying set of rows I in M, we will have that the target vector is in the span of I.

For any unauthorized set of rows I the target vector is not in the span of the rows of the set I. Moreover, there will exist a vector w such that $w \cdot (1, 0, 0 \ldots, 0) = -1$ and $w \cdot M_i = 0$ for all $i \in I$.

Using Access Trees. Prior works on ABE (e.g., [27]) typically described access formulas in terms of binary trees. Using standard techniques [6] one can convert any monotonic boolean formula into an LSSS representation. An access tree of ℓ nodes will result in an LSSS matrix of ℓ rows. We refer the reader to the appendix of [30] for a discussion of how to perform this conversion.

2.3 Ciphertext-Policy ABE

A ciphertext-policy attribute based encryption scheme consists of four algorithms: Setup, Encrypt, KeyGen, and Decrypt.

Setup(λ, U). The setup algorithm takes security parameter and attribute universe description as input. It outputs the public parameters PK and a master key MK.

*Encrypt(*PK, $M, \mathbb{A}$*).* The encryption algorithm takes as input the public parameters PK, a message M, and an access structure $\mathbb{A}$ over the universe of attributes. The algorithm will encrypt M and produce a ciphertext CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains $\mathbb{A}$.

*Key Generation(*MK, S*).* The key generation algorithm takes as input the master key MK and a set of attributes S that describe the key. It outputs a private key SK.

*Decrypt(*PK, CT, SK*).* The decryption algorithm takes as input the public parameters PK, a ciphertext CT, which contains an access policy $\mathbb{A}$, and a private key SK, which is a private key for a set S of attributes. If the set S of attributes satisfies the access structure $\mathbb{A}$ then the algorithm will decrypt the ciphertext and return a message M.

We now describe a security model for ciphertext-policy ABE schemes. Like identity-based encryption schemes [36, 13, 23] the security model allows the adversary to query for any private keys that cannot be used to decrypt the challenge ciphertext. In CP-ABE the ciphertexts are identified with access structures and the private keys with attributes. It follows that in our security definition the adversary will choose to be challenged on an encryption to an access structure $\mathbb{A}^*$ and can ask for any private key S such that S does not satisfy $\mathbb{A}^*$. We now give the formal security game.

Security Model for CP-ABE.

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

Phase 1. The adversary makes repeated private keys corresponding to sets of attributes $S_1, \ldots, S_{q_1}$.

Challenge. The adversary submits two equal length messages M_0 and M_1. In addition the adversary gives a challenge access structure $\mathbb{A}^*$ such that none of the sets $S_1, \ldots, S_{q_1}$ from Phase 1 satisfy the access structure. The challenger flips a random coin b, and encrypts M_b under $\mathbb{A}^*$. The ciphertext CT^* is given to the adversary.

Phase 2. Phase 1 is repeated with the restriction that none of sets of attributes $S_{q_1+1}, \ldots, S_q$ satisfy the access structure corresponding to the challenge.

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

The advantage of an adversary $\mathcal{A}$ in this game is defined as $\Pr[b' = b] - \frac{1}{2}$. We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2.

Definition 3. *A ciphertext-policy attribute-based encryption scheme is secure if all polynomial time adversaries have at most a negligible advantage in the above game.*

We say that a system is *selectively* secure if we add an Init stage before setup where the adversary commits to the challenge access structure $\mathbb{A}^*$. All of our constructions will be proved secure in the selective security model.

2.4 Bilinear Maps

We present a few facts related to groups with efficiently computable bilinear maps and then give our number theoretic assumptions.

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 be a bilinear map, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

Decisional Parallel Bilinear Diffie-Hellman Exponent Assumption. We define the decisional q-parallel Bilinear Diffie-Hellman Exponent problem as follows. Choose a group $\mathbb{G}$ of prime order p according to the security parameter. Let $a, s, b_1, \ldots, b_q \in \mathbb{Z}_p$ be chosen at random and g be a generator of $\mathbb{G}$. If an adversary is given $\boldsymbol{y}$=

$$g, g^s, g^a, \ldots, g^{(a^q)}, \ , g^{(a^{q+2})}, \ldots, g^{(a^{2q})}$$
$$\forall_{1 \leq j \leq q} \ g^{s \cdot b_j}, \ g^{a/b_j}, \ldots, g^{(a^q/b_j)}, , g^{(a^{q+2}/b_j)}, \ldots, g^{(a^{2q}/b_j)}$$
$$\forall_{1 \leq j,k \leq q, k \neq j} \ g^{a \cdot s \cdot b_k/b_j}, \ldots, g^{(a^q \cdot s \cdot b_k/b_j)}$$

it must remain hard to distinguish $e(g, g)^{a^{q+1}s} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$.

An algorithm $\mathcal{B}$ that outputs $z \in \{0, 1\}$ has advantage ϵ in solving decisional q-parallel BDHE in $\mathbb{G}$ if

$$\left| \Pr\left[\mathcal{B}(\boldsymbol{y}, T = e(g, g)^{a^{q+1}s}) = 0\right] - \Pr\left[\mathcal{B}(\boldsymbol{y}, T = R) = 0\right] \right| \geq \epsilon$$

Definition 1. *We say that the (decision) q parallel-BDHE assumption holds if no polytime algorithm has a non-negligible advantage in solving the decisional q-parallel BDHE problem.*

We give a proof that the assumption generically holds in the full version of our paper [42].

3 Our Most Efficient Construction

We now give our main construction that both realizes expressive functionality and is efficient and is provably secure under a concrete, non-interactive assumption.

In our construction the encryption algorithm will take as input a LSSS access matrix M and distribute a random exponent $s \in \mathbb{Z}_p$ according to M. Private keys are randomized to avoid collusion attack.

Setup(U). The setup algorithm takes as input the number of attributes in the system. It then chooses a group $\mathbb{G}$ of prime order p, a generator g and U random group elements $h_1, \ldots, h_U \in \mathbb{G}$ that are associated with the U attributes in the system. In addition, it chooses random exponents $\alpha, a \in \mathbb{Z}_p$.

The public key is published as

$$\text{PK} = \quad g, \ e(g, g)^{\alpha}, \ g^a, \ h_1, \ldots, h_U.$$

The authority sets $\text{MSK} = g^{\alpha}$ as the master secret key.

*Encrypt(*PK, $(M, \rho), \mathcal{M}$ *).* The encryption algorithm takes as input the public parameters PK and a message $\mathcal{M}$ to encrypt. In addition, it takes as input an LSSS access structure (M, ρ). The function ρ associates rows of M to attributes.

Let M be an $\ell \times n$ matrix. The algorithm first chooses a random vector $\boldsymbol{v} = (s, y_2, ..., y_n) \in \mathbb{Z}_p^n$. These values will be used to share the encryption exponent s. For $i = 1$ to ℓ, it calculates $\lambda_i = \boldsymbol{v} \cdot M_i$, where M_i is the vector corresponding to the ith row of M. In addition, the algorithm chooses random $r_1, \ldots, r_\ell \in \mathbb{Z}_p$.

The ciphertext is published as CT =

$$C = \mathcal{M} e(g,g)^{\alpha s},\ C' = g^s$$
$$(C_1 = g^{a\lambda_1} h_{\rho(1)}^{-r_1},\ D_1 = g^{r_1}), \ldots, (C_\ell = g^{a\lambda_\ell} h_{\rho(\ell)}^{-r_n},\ D_\ell = g^{r_\ell})$$

along with a description of (M, ρ).

*KeyGen(*MSK, S*).* The key generation algorithm takes as input the master secret key and a set S of attributes. The algorithm first chooses a random $t \in \mathbb{Z}_p$. It creates the private key as

$$K = g^\alpha g^{at} \quad L = g^t \quad \forall x \in S\ K_x = h_x^t.$$

*Decrypt(*CT,SK*).* The decryption algorithm takes as input a ciphertext CT for access structure (M, ρ) and a private key for a set S. Suppose that S satisfies the access structure and let $I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be a set of constants such that if $\{\lambda_i\}$ are valid shares of any secret s according to M, then $\sum_{i \in I} \omega_i \lambda_i = s$. (Note there could potentially be different ways of choosing the ω_i values to satisfy this.)

The decryption algorithm first computes

$$e(C', K) / \left(\textstyle\prod_{i \in I} (e(C_i, L) e(D_i, K_{\rho(i)}))^{\omega_i}\right) =$$
$$e(g,g)^{\alpha s} e(g,g)^{ast} / \left(\textstyle\prod_{i \in I} e(g,g)^{ta\lambda_i \omega_i}\right) = e(g,g)^{\alpha s}$$

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

3.1 Proof

An important step in proving our system secure will be for the reduction to "program" the challenge ciphertext into the public parameters. One significant obstacle that we will encounter is that an attribute may be associated with multiple rows in the challenge access matrix (i.e. the ρ function is not injective). This is analogous to an attribute appearing in multiple leafs in an access tree.

For example, suppose that in our reduction we want to program our parameters such that for h_x based on the i-th row of M^* if $\rho^*(i) = x$. However, if there exist $i \neq j$ such that $x = \rho(i) = \rho(j)$ then there is an issue since we must program *both* row i and row j in the simulation. Intuitively, there is a potential conflict in how to program the parameters.

In this reduction we resolve this by using different terms from the parallel BDHE assumption to program multiple rows of M^* into one group element corresponding to an attribute. The extra terms provided allow us to do so without ambiguity[4]. In Section 5 we show a tradeoff where our reduction can program the information using just the decisional Bilinear Diffie-Hellman assumption, but at some loss of efficiency.

We prove the following theorem.

Theorem 1. *Suppose the decisional q-parallel BDHE assumption holds. Then no polytime adversary can selectively break our system with a challenge matrix of size* $\ell^* \times n^*$*, where* $\ell^*, n^* \leq q$.

Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage $\epsilon = \mathsf{Adv}_{\mathcal{A}}$ in the selective security game against our construction. Moreover, suppose it chooses a challenge matrix M^* where both dimensions are at most q. We show how to build a simulator, $\mathcal{B}$, that plays the decisional q-parallel BDHE problem.

Init. The simulator takes in a q-parallel BDHE challenge $\boldsymbol{y}, T$. The adversary gives the algorithm the challenge access structure (M^*, ρ^*), where M^* has n^* columns.

Setup. The simulator chooses random $\alpha' \in \mathbb{Z}_p$ and implicitly sets $\alpha = \alpha' + a^{q+1}$ by letting $e(g,g)^{\alpha} = e(g^a, g^{a^q}) e(g,g)^{\alpha'}$.

We describe how the simulator "programs" the group elements $h_1, \ldots, h_U$. For each x for $1 \leq x \leq U$ begin by choosing a random value z_x. Let X denote the set of indices i, such that $\rho^*(i) = x$. The simulator programs h_x as:

$$h_x = g^{z_x} \prod_{i \in X} g^{a M^*_{i,1}/b_i} \cdot g^{a^2 M^*_{i,2}/b_i} \cdots g^{a^{n^*} M^*_{i,n^*}/b_i}.$$

Note that if $X = \emptyset$ then we have $h_x = g^{z_x}$. Also note that the parameters are distributed randomly due to the g^{z_x} value.

Phase I. In this phase the simulator answers private key queries. Suppose the simulator is given a private key query for a set S where S does not satisfy M^*.

The simulator first chooses a random $r \in \mathbb{Z}_p$. Then it finds a vector $\boldsymbol{w} = (w_1, \ldots, w_{n^*}) \in {\mathbb{Z}_p}^{n^*}$ such that $w_1 = -1$ and for all i where $\rho^*(i) \in S$ we have that $\boldsymbol{w} \cdot M^*_i = 0$. By the definition of a LSSS such a vector must exist. *Note that if such a vector did not exist then the vector* $(1, 0, 0, \ldots, 0)$ *would be in the span of* S*. See the discussion in Section 2.*

The simulator begins by implicitly defining t as

$$r + w_1 a^q + w_2 a^{q-1} + \cdots + w_{n^*} a^{q-n^*+1}.$$

It performs this by setting $L = g^r \prod_{i=1,\ldots,n^*} (g^{a^{q+1-i}})^{w_i} = g^t$.

[4] We note that certain assumptions have been leveraged to "program" a large amount of information into single group elements in other contexts. Gentry's reduction [24] embeds a degree q polynomial into a single group element.

We observe that by our definition of t, we have that g^{at} contains a term of $g^{-a^{q+1}}$, which will cancel out with the unknown term in g^{α} when creating K. The simulator can compute K as:

$$K = g^{\alpha'} g^{ar} \prod_{i=2,\ldots,n^*} (g^{a^{q+2-i}})^{w_i}.$$

Now we must calculate $K_x \; \forall x \in S$. First, we consider $x \in S$ for which there is no i such that $\rho^*(i) = x$. For those we can simply let $K_x = L^{z_x}$.

The more difficult task is to create key components K_x for attributes $x \in S$, where x is used in the access structure. For these keys we must make sure that there are no terms of the form g^{a^{q+1}/b_i} that we can't simulate. However, we have that $M_i^* \cdot \boldsymbol{w} = 0$; therefore, all of these terms cancel.

Again, let X be the set of all i such that $\rho^*(i) = x$. The simulator creates K_x in this case as follows.

$$K_x = L^{z_x} \prod_{i \in X} \prod_{j=1,\ldots,n^*} \left(g^{(a^j/b_i)r} \prod_{\substack{k=1,\ldots,n^* \\ k \neq j}} (g^{a^{q+1+j-k}/b_i})^{w_k} \right)^{M^*_{i,j}}$$

Challenge. Finally, we build the challenge ciphertext. The adversary gives two messages $\mathcal{M}_0, \mathcal{M}_1$ to the simulator. The simulator flips a coin β. It creates $C = \mathcal{M}_\beta T \cdot e(g^s, g^{\alpha'})$ and $C' = g^s$.

The tricky part is to simulate the C_i values since this contains terms that we must cancel out. However, the simulator can choose the secret splitting, such that these cancel out. Intuitively, the simulator will choose random $y'_2, \ldots, y'_{n^*}$ and the share the secret using the vector

$$\boldsymbol{v} = (s, sa + y'_2, sa^2 + y'_3, \ldots, sa^{n-1} + y'_{n^*}) \in {\mathbb{Z}_p}^{n^*}.$$

In addition, it chooses random values $r'_1, \ldots, r'_\ell$.

For $i = 1, \ldots, n^*$, we define R_i as the set of all $k \neq i$ such that $\rho^*(i) = \rho^*(k)$. In other words, the set of all other row indices that have the same attribute as row i. The challenge ciphertext components are then generated as

$$D_i = g^{-r'_i} g^{-sb_i}$$

$$C_i = h_{\rho^*(i)}^{r'_i} \left(\prod_{j=2,\ldots,n^*} (g^a)^{M^*_{i,j} y'_j} \right) (g^{b_i \cdot s})^{-z_{\rho^*(i)}} \cdot \left(\prod_{k \in R_i} \prod_{j=1,\ldots,n^*} (g^{a^j \cdot s \cdot (b_i/b_k)})^{M^*_{k,j}} \right)$$

Phase II. Same as phase I.

Guess. The adversary will eventually output a guess β' of β. The simulator then outputs 0 to guess that $T = e(g,g)^{a^{q+1}s}$ if $\beta = \beta'$; otherwise, it outputs 1 to indicate that it believes T is a random group element in $\mathbb{G}_T$.

When T is a tuple the simulator $\mathcal{B}$ gives a perfect simulation so we have that

$$\Pr\left[\mathcal{B}\left(\boldsymbol{y}, T = e(g,g)^{a^{q+1}s}\right) = 0\right] = \frac{1}{2} + \mathsf{Adv}_{\mathcal{A}}.$$

When T is a random group element the message $\mathcal{M}_\beta$ is completely hidden from the adversary and we have $\Pr[\mathcal{B}(\boldsymbol{y},T = R) = 0] = \frac{1}{2}$. Therefore, $\mathcal{B}$ can play the decisional q-parallel BDHE game with non-negligible advantage.

4 Constructions from Weaker Assumptions

Our first construction provided a very efficient system, but under a strong (but still non-interactive) assumption. To bridge this gap we introduce two additional constructions that provide a tradeoff of performance versus strength of assumptions. We effectively explore a spectrum between system efficiency and strength of assumption. The final construction is proven secure under the simple decisional-BDH assumption.

Overview. The primary obstacle in achieving security from weaker assumptions is that we must be able to reflect the challenge access structure M^* in the parameters during the reduction. We create two different constructions using the same framework.

In our full version [42] we give a construction provably secure under the existing d-BDHE assumption introduced by Boneh, Boyen and Goh [11]. To accommodate a weaker assumption we introduce a parameter $k_{\max}$ which is the maximum number of times any one attribute can appear in an access formula. A private key in this system will be a factor of $k_{\max}$ larger than our main construction.

Next, in Section 5 we give a construction provably secure under the much more standard decisional Bilinear Diffie-Hellman assumption. To realize security under this assumption our system must additionally introduce a parameter $n_{\max}$, where performance parameters will be a factor of $n_{\max}$ larger than our most efficient construction.

5 Bilinear Diffie-Hellman Construction

While our unrestricted construction realizes a potentially ideal type of efficiency, we would like to also show that secure CP-ABE systems can be realized from static assumptions. Here we show how to realize our framework under the decisional Bilinear Diffie Hellman d-(BDH) assumption.

The primary challenge with realizing a construction provably secure under BDH is we need a way for a reduction to embed the challenge matrix M^* in the parameters. Since the BDH assumption gives the reduction less components to embed this, there is no obvious path for reducing the previous constructions to d-BDH. We surmount this obstacle by expanding our ciphertexts and public parameter space. By doing this we enable our reduction to embed the challenge matrix.

Our construction is parametrized by a integer $n_{\max}$ that specifies the maximum number of columns in a ciphertext policy. The public parameters, keys and ciphertext size will all grow linearly in this parameter[5].

[5] One could achieve smaller ciphertexts by creating multiple systems with different $n_{\max}$ values and use the one that fit the actual policy most tightly.

Like our first construction we restrict $\rho()$ to be an injective function, but can alleviate this restriction by applying a similar transformation to allow an attribute to appear $k_{\max}$ times for some specified $k_{\max}$. Our construction follows.

Setup($U, n_{\max}$). The setup algorithm takes as input, U, the number of attributes in the system U and $n_{\max}$ the maximum number of columns in an LSSS matrix (or number of nodes in an access formula). It then creates a group $\mathbb{G}$ of prime order p and a generator g and chooses random elements $(h_{1,1}, \ldots, h_{1,U}), \ldots, (h_{n_{\max},1}, \ldots, h_{n_{\max},U})$ In addition, it chooses random exponents $\alpha, a \in \mathbb{Z}_p$.

The public key is published as

$$\text{PK} = \quad g,\ e(g,g)^{\alpha},\ g^{a},$$

$$(h_{1,1}, \ldots, h_{1,U}), \ldots, (h_{n_{\max},1}, \ldots, h_{n_{\max},U})$$

The authority sets $\text{MSK} = g^{\alpha}$ as the master secret key.

*Encrypt(*PK, $(M, \rho), \mathcal{M}$ *).* The encryption algorithm takes as input the public parameters PK and a message $\mathcal{M}$ to encrypt. In addition, it takes as input an LSSS access structure (M, ρ). The function ρ associates rows of M to attributes. In this construction we limit ρ to be an injective function, that is an attribute is associated with at most one row of M.

Let M be an $\ell \times n_{\max}$ matrix. (If one needs to create a policy for $n < n_{\max}$, then one can simply "pad out" the rightmost $n_{\max} - n$ columns with all zeros.) The algorithm first chooses a random vector $\boldsymbol{v} = (s, y_2, ..., y_{n_{\max}}) \in \mathbb{Z}_p^n$. These values will be used to share the encryption exponent s.

The ciphertext is published as

$$\text{CT} = C = \mathcal{M} e(g,g)^{\alpha s},\ C' = g^{s},\ \forall_{\substack{i=1,\ldots,\ell \\ j=1,\ldots,n_{\max}}} C_{i,j} = g^{a M_{i,j} v_j} h_{j,\rho(i)}^{-s}$$

along with a description of M, ρ.

*KeyGen(*MSK, S*).* The key generation algorithm takes as input the master secret key and a set S of attributes. The algorithm first chooses a random $t_1, \ldots, t_{n_{\max}} \in \mathbb{Z}_p$. It creates the private key as

$$K = g^{\alpha} g^{a t_1} \quad L_1 = g^{t_1}, \ldots, L_n = g^{t_{n_{\max}}}$$

$$\forall x \in S \quad K_x = \prod_{j=1,\ldots,n_{\max}} h_{j,x}^{t_j}.$$

*Decrypt(*CT,SK*).* The decryption algorithm takes as input a ciphertext CT for access structure (M, ρ) and a private key for a set S. Suppose that S satisfies the access structure and let $I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be a set of constants such that, if $\{\lambda_i\}$ are valid shares of any secret s according to M, then $\sum_{i \in I} \omega_i \lambda_i = s$. (Note there could potentially be different ways of choosing the ω_i values to satisfy this.)

The decryption algorithm first computes

$$e(C', K) / \left(\prod_{j=1,\ldots,n_{\max}} e(L_j, \prod_{i \in I} C_{i,j}^{\omega_i}) \right) \prod_{i \in I} e(K_{\rho(i)}^{\omega_i}, C')$$

$$\begin{aligned}
&= e(C', K) / \Big(\prod_{j=1,\ldots,n_{\max}} e(g^{t_j}, g^{\sum_{i \in I} aM_{i,j} v_j \omega_i}) \cdot \\
&\quad e(g^{t_j}, \prod_{i \in I} h_{j,\rho(i)}^{-s\omega_i}) \Big) \prod_{i \in I} e(K_{\rho(i)}^{\omega_i}, g^s) \\
&= e(C', K) / \prod_{j=1,\ldots,n_{\max}} e(g^{t_j}, g^{\sum_{i \in I} aM_{i,j} v_j \omega_i}) \\
&= e(C', K) / e(g^{t_1}, g^{\sum_{i \in I} aM_{i,1} v_1 \omega_i}) \\
&= e(g^s, g^{\alpha} g^{at_1}) / e(g, g)^{at_1 s} \\
&= e(g, g)^{\alpha s}
\end{aligned}$$

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

5.1 Proof

We prove the following theorem.

Theorem 2. *Suppose the decisional BDH assumption holds. Then no polytime adversary can selectively break our system.*

Due to space limitations we defer the proof of the system to our full version [42].

6 Large Universe of Attributes

One aspect of our main construction is that it defines the set of attributes to be used in the parameters. One useful feature is to be able to dynamically use any string as an attribute. In our full version [42] we show how in the random oracle we can realize any number of attributes with constant size parameters by simply hashing the attribute string. Also in our full version provide a large universe construction in the standard model.

Acknowledgements

We thank Matt Green, Kazuki Yoneyama and anonymous reviewers for useful comments.

References

[1] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005)

[2] Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-based encryption gone wild. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 300–311. Springer, Heidelberg (2006)
[3] Al-Riyami, S.S., Malone-Lee, J., Smart, N.P.: Escrow-free encryption supporting cryptographic workflow. Int. J. Inf. Sec. 5(4), 217–229 (2006)
[4] Bagga, W., Molva, R., Crosta, S.: Policy-based encryption schemes from bilinear pairings. In: ASIACCS, p. 368 (2006)
[5] Barbosa, M., Farshim, P.: Secure cryptographic workflow in the standard model. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 379–393. Springer, Heidelberg (2006)
[6] Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
[7] Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
[8] Baden, R., Bender, A., Spring, N., Bhattacharjee, B., Starin, D.: Persona: An online social network with user defined privacy. In: ACM SIGCOMM (2009)
[9] Bobba, R., Fatemieh, O., Khan, F., Gunter, A.K.C.A., Khurana, H., Prabhakaran, M.: Attribute-based messaging: Access control and confidentiality (2009) (manuscript)
[10] Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
[11] Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
[12] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
[13] Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
[14] Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS, pp. 647–657 (2007)
[15] Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
[16] Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (Without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
[17] Bradshaw, R.W., Holt, J.E., Seamons, K.E.: Concealing complex policies with hidden credentials. In: ACM Conference on Computer and Communications Security, pp. 146–157 (2004)
[18] Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
[19] Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)

[20] Chen, L., Harrison, K., Moss, A., Soldera, D., Smart, N.P.: Certification of Public Keys within an Identity Based System. In: Chan, A.H., Gligor, V.D. (eds.) ISC 2002. LNCS, vol. 2433, pp. 322–333. Springer, Heidelberg (2002)
[21] Chen, L., Harrison, K., Soldera, D., Smart, N.P.: Applications of Multiple Trust Authorities in Pairing Based Cryptosystems. In: Davida, G.I., Frankel, Y., Rees, O. (eds.) InfraSec 2002. LNCS, vol. 2437, pp. 260–275. Springer, Heidelberg (2002)
[22] Cheung, L., Newport, C.C.: Provably secure ciphertext policy abe. In: ACM Conference on Computer and Communications Security, pp. 456–465 (2007)
[23] Cocks, C.: An identity based encryption scheme based on quadratic residues. In: IMA Int. Conf., pp. 360–363 (2001)
[24] Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
[25] Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
[26] Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
[27] Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
[28] Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
[29] Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
[30] Lewko, A., Waters, B.: Decentralizing attribute-based encryption. Cryptology ePrint Archive, Report 2010/351 (2010), `http://eprint.iacr.org/`
[31] 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)
[32] Miklau, G., Suciu, D.: Controlling access to published data using cryptography. In: VLDB, pp. 898–909 (2003)
[33] 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)
[34] Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. In: ACM Conference on Computer and Communications Security, pp. 99–112 (2006)
[35] Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
[36] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
[37] Shi, E., Bethencourt, J., Chan, H.T.-H., Song, D.X., Perrig, A.: Multi-dimensional range query over encrypted data. In: IEEE Symposium on Security and Privacy, pp. 350–364 (2007)

[38] Smart, N.P.: Access Control Using Pairing Based Cryptography. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 111–121. Springer, Heidelberg (2003)
[39] Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: IEEE Symposium on Security and Privacy, pp. 44–55 (2000)
[40] Traynor, P., Butler, K.R.B., Enck, W., McDaniel, P.: Realizing massive-scale conditional access systems through attribute-based cryptosystems. In: NDSS (2008)
[41] Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
[42] Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290 (2008), http://eprint.iacr.org/