Attribute-Based Encryption: Applications and Future Directions

Bruhadeshwar Bezawada(✉) and Indrakshi Ray

Computer Science Department, Colorado State University,
Fort Collins, CO 80523, USA
{Bru.Bezawada,Indrakshi.Ray}@colostate.edu

Abstract. This survey focuses on the cryptographic access control technique, attribute-based encryption (ABE), its applications and future directions. Since its inception, there has been a tremendous interest in applying this technique to solve various problems related to access control. Significant research efforts have been devoted to design efficient constructions and operational parameters to suit various applications. The main functionality of ABE is to enforce cryptographic access control with help of policies specified over a set of system defined attributes. A key generator maps the attributes, in an access policy, into encryption and decryption keys for a resource access request. ABE is categorized into Key-Policy ABE (KP-ABE) and Cipher-text Policy ABE (CP-ABE), depending on the approach used to map the attributes to the encryption and decryption keys. Implementations of ABE have relied on mathematical primitives such as elliptic curves, pairing functions, generalized secret sharing notions and on the hardness of problems like computing discrete logarithm and computational Diffie-Hellman problem over elliptic curves. As they are essentially public-key systems, these schemes are usually proven secure under the semantically secure adaptive chosen cipher-text attack (IND-CCA). ABE has been utilized in solving a number of problems in different application domains including network privacy, broadcast encryption for on-demand television programming, health data access control, cloud security, and verifiable computation. In this survey, we discuss the evolution of ABE, covering significant developments in this area, the applications of ABE across various domains, and the future directions for ABE.

1 Introduction

1.1 Motivation

Access control of sensitive data is a central problem for information security and assurance. The goal is to ensure that only authorized entities are allowed access to sensitive data following certain system specific access policies. The ability to specify fine-grained expressive policies to capture all possible authorization contexts has been the holy grail of access control models. Attribute-based access control is an interesting model wherein a combination of attributes, which are

P. Samarati et al. (Eds.): Jajodia Festschrift, LNCS 11170, pp. 353–374, 2018.
https://doi.org/10.1007/978-3-030-04834-1_18

arbitrary strings, enables system administrators[1] to specify policies that are almost in natural language. For the rest of the survey, we will assume that the system administrator prefers to express access policies in terms of attributes defined over natural language.

In the modern computing scenario, the use of distributed storage has become the *de-facto* approach for the storage management problem. One interesting problem in this context is to secure data-at-rest and protect it from leakage through malicious channels. An attacker could obtain copies of the sensitive data through covert side-channels. Therefore, when data is stored in such third-party servers, there needs to be some assurance on the security of the data against such attacks. Data encryption protects against such leakages as an attacker obtaining a copy of the encrypted data through malicious channels will not be able to decrypt it.

When data is encrypted, the major challenge for a system administrator is in specifying access control policies using user attributes and to effectively create the bridge between the user attributes and the decryption keys for the encrypted data. This problem has been addressed by the cryptographic technique, *"attribute-based encryption"* (ABE), which describes algorithms to specify a data access policy in terms of attributes and to create mapping of such a policy to a decryption key. Due to its vast potential, ABE has received widespread attention in the community and has been the subject of active research. In the following discussion, we trace the development of attribute-based encryption starting with the general foundational concepts of identity-based encryption and fuzzy identity-based encryption.

1.2 Background: Identity-Based Encryption

The genesis of attribute-based encryption can be traced back to the notion of identity-based encryption (IBE) posed by Shamir in the 1984 paper [27]. The question was whether it is possible to use any generic public string as a public-key in a public-key cryptosystem. The answer to this question is to use a master private-key generator (PKG) that is responsible for providing the decryption keys that are tied to a generic identity such as an email address. Such a cryptosystem consists of four algorithms [9], `setup`, which generates a `master-key`, `extract`, which uses the `master-key` to map a private key to an arbitrary public key string $ID \in \{0,1\}^*$, `encrypt`, which encrypts messages using ID, and `decrypt`, which decrypts messages using the mapped private key[2]. The user possessing the identity ID needs to authorize himself to the PKG to obtain the necessary decryption keys.

[1] The system administrator is used in the generic sense and covers other designations like "data owner", "data base owner", "system designer", "reference monitor", "key generator" and so on.

[2] As much as possible the original notation of these seminal papers has been retained as a mark of honor to the inventors of these techniques. Additional notes have been added to help a broader audience to appreciate the nuances of these techniques.

Several non-trivial challenges needed to be addressed to achieve this task, specifically, there was need for a provably secure scheme under standard complexity assumptions based on well-known problems like the discrete logarithm problem (DLP) or the computational Diffie-Hellman problem (CDH). In their seminal work in [9] in 2001, Boneh and Franklin designed such a construction and proved it secure under the chosen-ciphertext attack [7,22], as is the standard for public-key cryptosystems[3]. Their approach used the findings of Joux [15] as basis, which showed that Weil pairing can be successfully used for developing cryptographic primitives. The Weil pairing provided the bridge for mapping a random public string -the user's identity, to a cryptographic public key -to encrypt data sent to this user, and allowed for the generation of a suitable private key -that is used to decrypt messages encrypted with the public key. We will first give some preliminaries for this scheme and then proceed to describe the construction in detail.

Bilinear Pairing. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of order q for some large prime q. An admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ between these two groups must satisfy the following properties:

1. *Bilinear*: We say that a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is bilinear if $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$.
2. *Non-degenerate*: As $\mathbb{G}_1, \mathbb{G}_2$ are groups of prime order and if P is a generator of $\mathbb{G}_1$ then $e(P, P)$ is a generator of $\mathbb{G}_2$ and hence, $e(P, P) \neq 1$.
3. *Computable*: There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}_1$.

The group $\mathbb{G}_1$ is a subgroup of the additive group of points of an elliptic curve $E/\mathbb{F}_p$ and $\mathbb{G}_2$ is a subgroup of the multiplicative group of a finite field $\mathbb{F}^*_{p^2}$. For rest of the survey, we assume that all schemes use elliptic curves on which admissible bilinear pairings exist subject to additional constraints as required by hardness problem described in the following.

A bilinear pairing can be used for building a cryptosystems only if the discrete logarithm problem is intractable for that elliptic curve. The decision problem of Diffie-Hellman (DDH) in this setting is easy as shown in [16], which is to distinguish between the distributions $\langle P, aP, bP, abP\rangle$ and $\langle P, aP, bP, cP\rangle$ where a, b, c are random in $\mathbb{Z}^*_q$ and P is random in $\mathbb{G}^*_1$. The computational Diffie-Hellman (CDH) problem, however, is still believed to be intractable. The CDH problem is as follows: given $\langle P, aP, bP\rangle$ in $\mathbb{G}_1$ to find abP in $\mathbb{G}_1$ and this is equivalent to the hardness of the discrete logarithm problem (DLP) in cyclic groups. To prove the security of their IBE scheme, Boneh and Franklin defined a modified version of the CDH problem on bilinear pairing called Bilinear Diffie-Hellman (BDH) assumption.

[3] Canetti *et al.* gave the first IBE construction in [10] with slightly weaker security.

Bilinear Diffie-Hellman (BDH) Assumption. The BDH assumption is as follows: given an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and the distribution $\langle P, aP, bP, cP \rangle$ in $\mathbb{G}_1$, an adversary has negligible advantage of computing $e(P, P)^{abc}$. At present, this problem is known to be hard [15].

1.3 IBE: Construction from Weil Pairing

Now, given the above background, Boneh and Franklin's Identity-Based Encryption(IBE) scheme is described in the following discussion, which consists of the necessary four algorithms: `setup`, `extract`, `encryption` and `decryption`.

Setup: The PKG[4] uses a system security parameter $k \in G^+$ to generate the necessary parameters in the setup phase.

- *Step 1*: Using k generate a prime q, two groups $\mathbb{G}_1$, $\mathbb{G}_2$ of order q, and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Choose a random generator $P \in \mathbb{G}_1$.
- *Step 2*: Pick a random $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$. The value of s can be viewed as the master-secret held by the PKG and it is used as a link between the user public-identity and the corresponding private key generated from the identity.
- *Step 3*: Choose two cryptographic hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ for some n.

The message space is $\mathcal{M} = \{0,1\}^n$, the ciphertext space is $\mathcal{C} = G_1^* \times \{0,1\}^n$ and, the system parameters are $\texttt{params} = \langle q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_{pub}, H_1, H_2 \rangle$ The master-key is $s \in Z_q^*$ where the security of s, in $P_{pub} = sP$, follows from the intractability of the DLP problem for selected elliptic curves.

Extract: The purpose of this algorithm is to generate the private key for the given public-identity. Given the public string $ID \in \{0,1\}^*$ the algorithm computes $Q_{ID} = H_1(ID) \in G_1^*$, and sets the private key $d_{ID} = sQ_{ID}$ where s is the master key.

Encrypt: Any other user wishing to send a message $M \in \mathcal{M}$, under the public key ID, computes $Q_{ID} = H_1(ID) \in G_1^*$ and chooses a random $r \in Z_q^*$. The parameter r has interesting properties, in that, it adds randomness to the encryption process and it is only unmasked through the Weil pairing operation, making it difficult to subvert this value. Now, set the ciphertext to be $C = \langle rP, M \bigoplus H_2(g_{ID}^r) \rangle$ where $g_{ID} = e(Q_{ID}, P_{pub}) \in \mathbb{G}_2^*$.

[4] Private-key Generator as defined previously.

Decrypt: Let $C = \langle U, V \rangle \in \mathcal{C}$ be a ciphertext encrypted using the public key ID where $U = rP$ and $V = M \bigoplus H_2(g_{ID}^r)$. We assume that the user ID receives the private key d_{ID} from the PKG. To decrypt C using the private key d_{ID} compute
: $V \bigoplus H_2(e(d_{ID}, U))$
$= V \bigoplus (H_2(e(sQ_{ID}, rP)) = V \bigoplus H_2(e(Q_{ID}, P)^{sr})$
$= V \bigoplus H_2(e(Q_{ID}, sP)^r)$[*Now, substitute V's value*]
$= M \bigoplus H_2(g_{ID}^r) \bigoplus H_2(e(Q_{ID}, sP)^r) = M$ □

The key takeaway from the IBE construction is that it showed that any arbitrary string can be used a public-key and there exist strong cryptographic constructs that allow us to generate a usable public-key cryptosystem. In further explorations, Yao *et al.* [30], showed that IBE can be applied to multiple hierarchically arranged identities giving rise to what is known as Hierarchical Identity-Based Encryption (HIBE). The HIBE construction showed that it is possible to encrypt a message under several identities while allowing each identity to decrypt the message. Although HIBE was not deployed in practical applications, it acted as a proof-of-concept for attribute-based encryption where an attribute can be viewed as an identity in HIBE. Sahai and Waters explored this notion further in their work called *fuzzy identity based encryption* (FIBE) [24], which eventually led to the development of efficient attribute-based encryption (ABE) techniques.

1.4 Fuzzy Identity-Based Encryption: FIBE

In chronological terms, FIBE was the precursor to attribute-based encryption and our discussion focuses on this facet of FIBE, although FIBE has other applications as well. The key notion of FIBE is to allow decryption of message with some *"tolerance"* in public-keys, *i.e.*, a user is allowed to produce a public-key that is within a certain *threshold* (of similarity), and be able to decrypt the message encrypted under a large public identity, which exceeds the threshold. To appreciate this notion, the "public-key" is expressed as a set of elements ω, which is derived from an identity. Under FIBE, any user who produces ω' in such a way that $|\omega \bigcap \omega'| \geq d$ for a threshold d, then she will be allowed to decrypt the message encrypted under ω.

Now, extending this notion further, a user's identity can be seen as being a subset of the elements or attributes from the set ω and therefore, the user can utilize her attributes to decrypt a message as long as the user attributes satisfy the condition on the system threshold d. This notion can be intuitively viewed as enforcing access control, *i.e.*, only those users who have the necessary attributes are authorized to access the data. If the attributes are assigned (or verified) by an authority or reference monitor, *e.g.*, a PKG, then attribute-based data access control is possible. A threshold secret distribution system, such as Shamir's secret sharing scheme [26], can be used to achieve the desired functionality. Therefore, FIBE can be viewed as a combination of IBE and Shamir's secret sharing scheme with slightly different complexity assumptions, which we state next.

Decisional Bilinear Diffie-Hellman (BDH) Assumption. Let $a, b, c, z \in Z_p$ be chosen at random. The Decisional BDH assumption is that no polynomial-time adversary is able to distinguish the tuple $(A = g^a, B = g^b, C = g^c, Z = e(g,g)^{abc})$ from the tuple $(A = g^a, B = g^b, C = g^c, Z = e(g,g)^z)$ with non-negligible advantage.

Decisional Modified Bilinear Diffie-Hellman (MBDH) Assumption. Similarly, the Decisional MBDH assumption is that no polynomial-time adversary is able to distinguish the tuple $(A = g^a, B = g^b, C = g^c, Z = e(g,g)^{\frac{ab}{c}})$ from $(A = g^a, B = g^b, C = g^c, Z = e(g,g)^z)$ with non-negligible advantage.

FIBE Construction: The identities are sets of attributes and d represents the error-threshold for the intersection of sets, *i.e.*, it is the minimum possible size of the intersection. Now, when the PKG creates a private key for a user she will associate a random $d-1$ degree polynomial, $q(x)$, with each user with the condition that each polynomial has the same valuation at point 0, that is $q(0) = y$, which represents the secret that will be used to unmask the encryption of the cipher-text. Given d points of a polynomial of degree d, we can reconstruct the polynomial using Lagrange's polynomial interpolation method. The Lagrange co-efficient, $\Delta_{i,S}$ for point $i \in Z_p$ and set S of elements is defined as follows:

$$\Delta_{i,S} = \prod_{j \in S, j \neq i} \frac{x - i}{i - j}$$

Setup. Let $\mathcal{U} \subset \mathbb{Z}_p^*$ denote the universe of elements. Choose $t_1, \cdots, t_{|U|}$ and y uniformly at random from $\mathbb{Z}_p$. Now, the published public parameters are:

$$T_1 = g^{t_1}, \cdots, T_{|U|} = g^{t_{|U|}}, Y = e(g,g)^y.$$

The master key is: $t_1, \ldots, t_{|U|}$ and y.

Key Generation. To generate a private key for identity $\omega \subset \mathcal{U}$ the following steps are taken. A $d1$ degree polynomial q is randomly chosen such that $q(0) = y$. The private key consists of components, $(D_i)_{i \in \omega}$, where $D_i = g^{\frac{q(i)}{t_i}}$ for every $i \in \omega$. The aim of FIBE is to ensure that this key can decrypt a message that is encrypted with a public identity $\omega' \leq \omega$ while subject to the necessary tolerance threshold.

Encryption. Given a public key ω' and message $M \in \mathbb{G}_2$, a random value $s \in Z_p$ is chosen. The ciphertext is then published as:

$$E = (\omega', E' = MY^s, \{E_i = T_i^s\}_{i \in \omega'}).$$

The intuition of this construction is that the secret s needs to be unmasked to extract the message M.

Decryption. Now, consider that a ciphertext, E, is encrypted with a key for identity ω' and the user has a private key for identity ω where $|\omega \bigcap \omega'| \geq d$. Choose an arbitrary d-element subset, S, of $\omega \bigcap \omega'$. The decryption is as follows:

$$E' / \prod_{i \in S} (e(D_i, E_i))^{\Delta_{i,S}(0)} = Me(g,g)^{sy} / \prod_{i \in S} (e(g^{\frac{q(i)}{t_i}}, g^{st_i}))^{\Delta_{i,S}(0)}$$

$$= Me(g,g)^{sy} / \prod_{i \in S} (e(g,g)^{sq(i)})^{\Delta_{i,S}(0)} = M \qquad \square$$

The last step is an addition of the d Lagrange coefficients in the denominator's exponent and evaluates the polynomial at point 0, which is y. This subsequently cancels out the $e(g,g)^{sy}$ term in the denominator.

Complexity of FIBE. The size of the cipher-text is linear in the size of the identity being encrypted. The number of exponentiations are linear in the size of the identity description and d bilinear pairings per decryption.

The FIBE system showed that it is possible create an ABE that will allow users with different attributes to share access to the same data item. However, the FIBE system is unsuitable for general access control as it's use of threshold secret sharing is not very expressive in terms of specifying access control policies. Any user with d or more attributes will be able to decrypt the message. In real-world applications, access control policies are usually specified as a boolean function of the attributes with AND and OR conditions. These considerations are handled by the general ABE techniques, which we will describe in detail in the following sections.

Organization. In Sect. 2, we will describe two popular construction of ABE, Key-Policy ABE and Ciphertext-Policy ABE. In Sect. 3, we describe the various applications of ABE and show the applicability of ABE across a wide variety of application domains. In Sect. 4, we describe the various challenges in ABE and point out possible future directions in this area of research and make concluding remarks in Sect. 5.

2 Attribute-Based Encryption

Attribute-based encryption (ABE) can be viewed as a technique for enforcing cryptographic access control on data where the access policy is specified over attributes such as: {Name = "John" AND (Age = "30" OR (Location = "Virginia" AND Role = "Manager")) AND Department = "Finance"}. In general, an access control policy is specified as a boolean function over the attributes as it is the most intuitive and expressive approach. A key advantage of ABE is that a single encryption is likely to encompasses a wide range of access policies due to the expressive nature of boolean logic. ABE primarily comes in two flavors, depending on the way in which the decryption keys are mapped to the attributes, Key-Policy ABE (KP-ABE) [13] and Ciphertext-Policy ABE (CP-ABE) [8]. The mathematical constructs used in ABE are almost same as in FIBE, *i.e.*, elliptic-curve pairings and linear secret sharing schemes (LSSS), with

some modifications necessary for expressing the complex access control policies. ABE constructions are primarily based on the Bilinear Diffie-Hellman (BDH) assumption (cf. Sect. 1.2).

2.1 Access Structures

Let $P = \{P_1, P_2, \cdots, P_n\}$ be a set of parties. In ABE, these are equivalent to the set of user specific attributes. Intuitively, an access structure is a collection of all authorized subsets of P. Now, an authorized collection $\mathcal{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathcal{A}$ and $B \subseteq C$ then $C \in \mathcal{A}$[5].

Linear Secret Sharing Schemes (LSSS). In a linear secret sharing scheme [26], an authorized party distributes "shares" of a secret among a group of users. An authorized group of users can recover the secret by using a linear combination of these shares.

Monotone Span Programs (MSP). Let $\mathcal{K}$ be a field, and $\{x_1, x_2, \cdots, x_n\}$ be a set of variables. A span program over $\mathcal{K}$ is labeled $M(M, \rho)$ where M is a matrix over $\mathcal{K}$ and ρ is a labeling of the rows of M by literals from $\{x_1, \cdots, x_n\}$ or $\{\bar{x_1}, \cdots, \bar{xn}\}$ and every row is labeled with one literal. Now, for an input $\delta \in \{0,1\}^n$, define sub-matrix M_δ of M consisting of rows whose labels are set to 1 by δ, *i.e.*, rows are either labeled by some x_i and $\delta_i = 1$, or rows labeled by $\bar{xi}$ and $\delta_i = 0$. The span program accepts δ if and only if there exists some linear combination of rows induced by δ that generates the all **1**'s row. A span program is called monotone span program (MSP) if the labels of the rows are only positive literals $\{x_1, \cdots, x_n\}$ where the MSP computes monotone functions. An MSP is said to compute a boolean function f if every δ where $f(\delta) = 1$ is accepted by the MSP. There is an equivalence relation between any LSSS and a MSP [6], which is a fact used by most ABE schemes to generate the LSSS matrix from the MSP. Lewko and Waters [18] provide an efficient algorithm to generate the LSSS matrix from the boolean function representation.

Access Trees. Access trees are used to represent the boolean functions defined over the attributes. The decryption keys are identified by a tree-access structure $\mathcal{T}$ in which each interior node of the tree is a threshold gate and the leaves are associated with attributes. This setting is very expressive as it is possible to represent a tree with AND and OR gates by using respectively 2-of-2 and 1-of-2 threshold gates. Each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value. If num_x is the number of children of a node x and k_x is its threshold value, then $0 < k_x \leq num_x$. When $k_x = 1$, the threshold gate is an OR gate and when $k_x = num_x$, it is an AND

[5] Although most ABE techniques in literature primarily work with monotone access structures, as defined next, there are schemes [21] that support non-monotone access structures as well.

gate. Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$. The parent of the node x in the tree is denoted by $parent(x)$. The function $att(x)$ is defined only if x is a leaf node and denotes the attribute associated with the leaf node x. The children of a node x are numbered from 1 to num denoted by the function $index(x)$, which returns the number associated with a child node of x.

A user will be able to decrypt a ciphertext if and only if there is an assignment of attributes to the leaf nodes of the tree such that the threshold gate of the root of the tree is eventually satisfied with this assignment. Let R denote the root of an access tree and $\mathcal{T}_x$ denote a sub-tree rooted at node x with threshold condition k_x. An access tree is said to be satisfied, if for some set of attributes, a recursive evaluation of the tree, starting from the leaf nodes corresponding to these attributes, satisfies the threshold condition k_R of the root node R. Since all intermediate nodes x are threshold gates, they need to be satisfied before the root node threshold condition is satisfied.

Attribute Generation. The general approach to generate attributes is to first express the boolean function as an access tree and generate labeling of the leaf nodes, which can then be represented as rows of an MSP as described in [18]. The threshold gates are expanded into AND or OR gates. Since a straightforward expansion of threshold gates into AND or OR gates might generate a large access tree, there have been various optimization methods [19] that create smaller MSPs to minimize the number of AND gates. Finally, each row of the resulting MSP corresponds to an attribute.

2.2 Key-Policy Attribute-Based Encryption KP-ABE

The logical intuition of KP-ABE is to encode the access policies within the decryption keys of the user depending on the attributes of the user, *i.e.*, the decryption key of the user encapsulates the access policies of that user. The construction follows the standard procedures of Setup, Key Generation, Encryption and Decryption as in IBE. While the Setup and Key Generation procedures of KP-ABE are identical to those of FIBE, the difference is in the Encryption and Decryption procedures, which we describe next.

Encryption. Choose a random polynomial q_x for each node x, including the leaves, in the access tree $\mathcal{T}$, such that the degree d_x of the polynomial is one less than the threshold value k_x of that node, *i.e.*, $d_x = k_x - 1$. For the root node R, set $q_R(0) = y$ and choose d_R other points of the polynomial q_R randomly to define it completely. The intermediate nodes are encoded based on the polynomial defined for their respective parent nodes. Specifically, the secret of a child node is generated as a random point of the polynomial associated with the parent node. For an intermediate node x, set $q_x(0) = q_{parent(x)}(index(x))$ and choose

d_x other points randomly[6] to completely define q_x. For each leaf node x, such that $i = att(x)$, a secret value is associated as follows:

$$D_x = g^{\frac{q_x(0)}{t_i}}$$

The set of all such values is the decryption key $D = \{D_x = g^{\frac{q_x(0)}{t_i}}\}\ \forall x$. A user will receive a subset of D depending on her attributes.

Decryption. Define a recursive algorithm, $DecryptNode(E, D, x)$ that takes as input the ciphertext $E = (\gamma, E', \{E_i\}_{i\in\gamma})$, the private key D and a node x in the tree. The algorithm outputs a group element of $\mathbb{G}_2$ or $\perp$. Let $i = att(x)$ and if the node x is a leaf node then:

$$DecryptNode(E, D, x) = \begin{cases} e(D_x, E_i) = e(g^{\frac{q_x(0)}{t_i}}, g^{s.t_i}) = e(g,g)^{s.qx(0)}, & \text{if } i \in \gamma \\ \perp, & \text{otherwise} \end{cases}$$

For any other node x, the $DecryptNode$ algorithm is applied recursively. If there are at least $k_x \in S_x$ child nodes that satisfy the condition for a set S_x, the decryption algorithm is as follows:

$$F_x = \prod_{z\in S_x} F_z^{\Delta_{i,S'_x}(0)} \text{ where } i = index(z) \text{ and } S'_x = index(z) : z \in S_x$$

$$= \prod_{z\in S_x} (e(g,g)^{s.q_z(0)})^{\Delta_{i,S'_x}(0)} = \prod_{z\in S_x} \left(e(g,g)^{s.q_{parent(z)}index(z)}\right)^{\Delta_{i,S'_x}(0)}$$

$$= \prod_{z\in S_x} e(g,g)^{s.q_x(i).\Delta_{i,S'_x}(0)} = e(g,g)^{s.q_x(0)}$$

Proceeding recursively, at root node $DecryptNode(E, D, R) = e(g,g)^{ys} = Y^s$, if and only if the user attributes satisfy the tree and given that $E' = MY^s$, it is straightforward to obtain M by dividing out Y^s.

The size of the public parameters are linear in the number of attributes and the decryption complexity determines the number of pairings required. Various techniques [4,18,23,29] have been proposed to optimize this process.

2.3 Cipher-Text Policy Attribute-Based Encryption CP-ABE

The CP-ABE [8] is a more popular version of ABE due its structure and inherent ability to protect data on outsourced servers. In KP-ABE, the encrypter may not have control on who will be able to decrypting the cipher-text as the decryption keys are with the users. CP-ABE is a dual of KP-ABE *i.e.*, the access policies are encoded inside the cipher-text and the user needs to provide valid attributes to be able to decrypt the message. The PKG only needs to check the user attributes and perform the decryption accordingly. The CP-ABE differs primarily in the encryption and decryption steps. The remaining constructs of access tree and the complexity assumptions are the same as in KP-ABE.

[6] The choice of random points is essential due to the condition on $q_x(0)$. A randomly defined polynomial will not satisfy this property.

Setup. The setup algorithm chooses a bilinear group $\mathbb{G}_0$ of prime order p with generator g and two random exponents $\alpha, \beta \in \mathbb{Z}_p$. The public key is published as:

$$PK = \mathbb{G}_0, g, h = g^{\beta}, f = g^{1/\beta}, e(g,g)^{\alpha}$$

and the master key MK is (β, g^{α}).

Encrypt. The access tree $\mathcal{T}$ is included with the cipher-text. A secret parameter s is chosen and let $q_R(0) = s$ for $\mathcal{T}$. This parameter is used to encode the message. Let Y denote the set of leaf nodes in $\mathcal{T}$ and $att(y)$ denote the attribute value associated with leaf node $y \in Y$.

$$CT = \left(\mathcal{T}, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)}\right).$$

Note that, the attributes are tied to the cipher-text in this construction through C'_y and will be used to recover the message.

KeyGen. The key generation algorithm uses the master-key MK and the set of user attributes S to output a decryption key that identifies with S. The algorithm first chooses a random $r \in \mathbb{Z}_p$ and, a random $r_j \in \mathbb{Z}_p$ for each attribute $j \in S$. The decryption key is computed as:

$$SK = \left(D = g^{\frac{(\alpha+r)}{\beta}}, \forall j \in S : D_j = g^r.H(j)^{r_j}, D'_j = g^{r_j}\right).$$

Note that, α is part of the decryption key as it is required to unmask, $e(g,g)^{\alpha s}$ and also, that $H(j)$ is the same as $H(att(y))$ since j and $att(y)$ are attributes.

Decrypt. The decryption operation requires the decryption key SK and the cipher-text $CT = (\mathcal{T}, \tilde{C}, C, \forall y \in Y : C_y, C'_y)$. The algorithm uses the routine $DecryptNode(CT, SK, x)$ where x denotes a node in the access tree $\mathcal{T}$. As in KP-ABE, the algorithm is recursive and invoked at the root node R of $\mathcal{T}$. Assuming that the recursion has reached a leaf node, x, we let $i = att(x)$ where $i \in S$ and perform the following steps:

$$DecryptNode(CT, SK, x) = \frac{e(D_i, C_x)}{e(D'_i, C'_x)} = \frac{e(g^r.H(i)^{r_i}, h^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})} = e(g,g)^{rq_x(0)}$$

Now, if x is a non-leaf node, $DecryptNode(CT, SK, x)$ works as follows: $\forall z$ where z is a child of x, we perform $DecryptNode(CT, SK, z)$ and store the output as F_z. The rest of the interpolation steps are same as in Sect. 2.2. Let the output of this step be $F_x = e(g,g)^{r.q_x(0)}$ and finally, at the root node this will be $F_R = e(g,g)^{r.q_R(0)} = e(g,g)^{r.s}$. The final decryption of M is as follows: $\tilde{C}/(e(C,D)/F_R) = \tilde{C}/\left(e(h^s, g^{(\alpha+r)/\beta})/e(g,g)^{rs}\right) = M$. On an average, the complexity of CP-ABE is close to KP-ABE with slight changes in the setup and the final decryption step.

This concludes the discussion of KP-ABE and CP-ABE. We discuss the various applications of these techniques in the following section.

3 Applications of Attribute-Based Encryption

In this section, we outline the various applications in which ABE has proven to be valuable. First, we discuss where the respective techniques of KP-ABE and CP-ABE are likely to be useful. Second, we discuss the various ABE applications including a commercial case study for commercializing ABE. For each domain, we discuss the details of how ABE was used and adapted to suit the respective application domain.

3.1 KP-ABE or CP-ABE?

The two ABE techniques are dual in nature and this brings in a question of which applications will benefit by KP-ABE and which ones by CP-ABE.

The access control view implemented by KP-ABE is analogous to that of a *capability list*, *i.e.*, a user's decryption keys decide the set of objects that are allowed. The user's private keys encapsulate the mapping between her attributes and the corresponding access policy. Therefore, KP-ABE implements a user-oriented view of access control. For instance, this kind of implementation might be best suited for system controlling access to its users on a local network.

Conversely, the access control view implemented by CP-ABE is analogous to an *access control list*, *i.e.*, the resource or object has a list of authorized users and their respective permissions on that object. In CP-ABE, the object encapsulates the user attributes with the corresponding access control policy. Therefore, CP-ABE implements an object-oriented view of access control. For instance, this kind of implementation might be best suited for outsourced data storage or data-at-rest applications.

Their dual nature allows either ABE technique to be used in any given application and the choice is mainly dictated by the access control policy complexity and performance constraints, among other factors. In the following discussion, for each of the applications, we mention the type of ABE that is best suited for the application.

3.2 On-Demand Live TV Broadcasting

One of the first practical applications where ABE was tested was for encryption of on-demand broadcast content in [28,32]. Such systems are classified under the conditional access (CA) systems. We focus on the work from [28] for this discussion where the authors used an early variation of KP-ABE, which is a technique called "large-universe" variant of FIBE [24] by Sahai and Waters. In [28], the authors identify three forms of on-demand services: subscription channels with a monthly fees, pay-per-view service where a user signs up for a program of interest, and ad-hoc pay-per-view where a user can sign up for a program without any advance notice or setup by hitting the "subscribe" button on his remote control. One approach used by content providers is to arrange users into group as per their subscribed programs. For normal subscription, this solution is scalable and efficient as users are relatively static in a group for a fixed amount

of time. However, for pay-per-view programs, the group membership is dynamic and causes problems to the service provider in terms of group management for large groups or flash crowds. To solve this problem, the solution devised was to create a two-tiered architecture and use ABE to control access to the content.

First, the users are divided into smaller groups and assigned a "group" specific attribute. The "group" attribute needs to be decrypted before gaining access to the final content. At the lower-tier this access is based on the user attributes and control is fine-grained. This tiering ensures that group management is performed within smaller groups and does not affect the entire universe of users accessing the same content. By creating such a tiered architecture, the solution demonstrated that it is possible to use ABE in novel ways to solve practical systems.

The authors validated their architecture by considering systems with 50k, 100k and 500k viewers. The decryption costs for accessing content was of the order of 50 to 100 ms. The system was able to handle several simultaneous joins and tolerated leaves of the group with minimal latency for real-world workloads.

3.3 Online Social Network Privacy

Online social networks (OSN) like Facebook® have often leaked private information of users to third-parties. While there are security and privacy controls in place, these are insufficient as a user cannot achieve fine-grained access control of his data. Users are forced to rely on the OSN service to protect personal information but the OSN provider seeks to benefit from examining and sharing that information. In [5], the authors solve this problem by creating a framework called Persona that enables users to control the flow of their personal information to their OSN connections in a fine-grained manner. The authors compose CP-ABE with public-key cryptosystems and symmetric key management to create a usable access control framework for OSNs.

Intuitively, Persona enables users to create groups and choose which users are part of a given group. With the help of ABE, users can define the attributes necessary to be part of specific groups and users control access to personal data by releasing encrypted data to groups. This allows users to have fine-grained access control over their data without relying on unknown policies used by the commercial OSNs. This entire framework was implemented and tested on mobile phone devices, which represents the majority of OSN user base.

The authors demonstrated that this system is scalable and can achieve the desired functionality without effecting user's quality of experience. The design of this system allows it to inter-operate with existing OSNs, specifically, the Persona prototype integrates with Facebook. The Persona applications are accessible as Facebook applications and can interact with Facebook's API, providing privacy-enabled applications through the popular interfaces of Facebook. For various work loads, the authors demonstrated that the page load times of pages with encrypted data was in the range of few seconds and did not impose any noticeable change to the experience of users when accessing encrypted content. This system

was a strong validation to show that ABE is practical even on mobile devices, which is the popular platform for accessing OSN data.

3.4 Assurance for Cloud Storage Data

Cloud storage services have become very popular for storing user data and provide good interfaces for sharing such data among several users. However, there are inherent challenges in maintaining the security of such data as it may be leaked through misconfiguration or due to an attack from outside. There are many new challenges for data security and access control when users outsource sensitive data for sharing on cloud servers, which are not within the same trusted domain.

In [31], the authors explored the use of ABE for these problems and described a solution framework that uniquely combined KP-ABE, Proxy Re-Encryption (PRE) and lazy re-encryption where re-encryption is a means of revoking users from a given access control policy. As ABE can involve considerable overhead for the data owner, the authors devised solutions so that the data owner could delegate tasks of data file re-encryption and user secret key update to cloud servers without disclosing data contents or user access policies. For this application, the reason for using KP-ABE as the base ABE is to allow authorized parties to seamlessly access the data from the cloud without burdening the data owner. They used the popular PRE cryptographic primitive in which a semi-trusted proxy is able to convert a ciphertext encrypted under Alice's public key into another ciphertext that can be opened by Bob's private key without accessing the underlying plaintext. They were able to validate the security of these constructions, a result that demonstrated that ABE is capable of solving difficult problems and could be composed with other primitives to create new solution frameworks that could offset any overheads introduced by ABE.

3.5 Fine-Grained Health-Record Access Control

The protection of patient health records has been a critical area of privacy and has received considerable attention. However, controlling access to health data is still an unsolved problem. Furthermore, when records are transmitted among institutions, recipients of the records obtain the plain-text records and this data might be cached in an unprotected way on user devices. On the other hand, most hospital systems require online access control decisions. If the server is unavailable, access control decisions are not possible and the records cannot be obtained. Medical administrators are faced with a tremendous number of records with a wide array of policies associated with them. There are dozens of personnel, *e.g.*, pharmacists, doctors, nurses, billing staff, auditors and so on, with varying levels of authorization, who are attempting to access this sensitive data. A sample illustration of the complexity of access is shown in Fig. 1 [31]. The state of the art solution of using an access control matrix to enumerate access and provide decisions is complex, costly, and error prone.

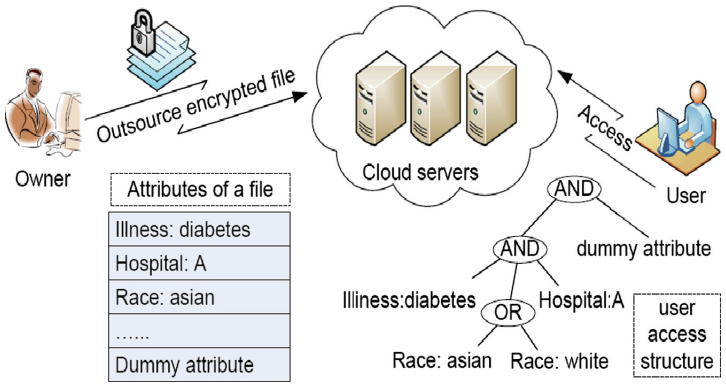

Fig. 1. Access control policies for health records

In [3], the authors provided a solution to this problem by using ABE, which is specifically targeted to mobile phones. They implemented a prototype system with a KP-ABE and CP-ABE library, which included an iPhone® app for storing and managing EMRs offline and enabled for flexible and automated policy generation. For automatic policy generation, they parsed the XML-based health-records, which contained the roles allowed to access the record, to calculate an appropriate access policy and encrypt the data using the policy attributes. Figure 2, shows their system and the interactions of the various entities involved.

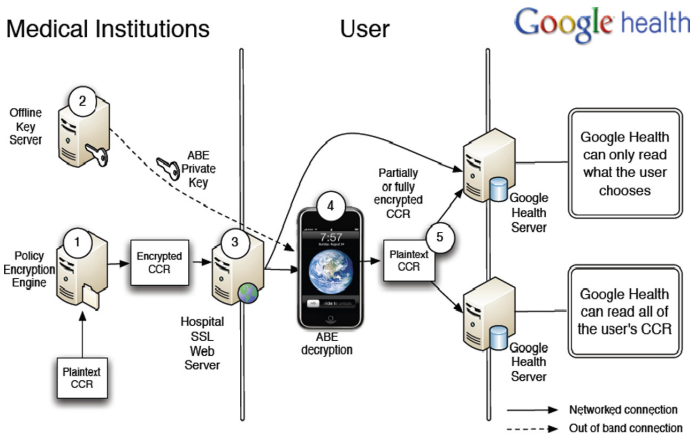

Fig. 2. Protection of health records for mobile phones

To demonstrate the validity of their system, the authors considered the complexity of policy generation, encryption and decryption with variable number of attributes. In practice, the encryption and decryption times were the order

of a few seconds for about 100 attributes. When revoking access to the records was checked, the re-encryption of data was in order of 3.5 s for 50 to 80 users, which represents a significant revocation scenario. This demonstrates that ABE can be integrated with lightweight devices such as mobile phones and provide guarantees of security without affecting user experience.

3.6 Policy Sealed Data

Trusted platform modules (TPMs) are seen as a way of enforcing secure access control on outsourced data. However, accidental or intentional mismanagement of cloud software poses a serious threat to the security of customer data hosted on the cloud. TPMs have a host of problems that do not address this kind of leakage in a satisfactory manner. Mainly, TPM abstractions were designed to protect data on a stand-alone machine and are unsuitable for multi-host and multi-tenant data that has potential of moving seamlessly across the platforms. Furthermore, the state-of-the-art implementation of TPM abstractions is inefficient and introduces scalability bottlenecks to cloud services. An attacker is assumed to be an agent with privileged access to the cloud node's management interface who is typically a cloud provider employee and manages cloud software and behaves inappropriately. The attacker seeks to compromise customer data by extracting it from integrity-protected cloud nodes and is successful if either the data is moved to a machine running insecure software platform or is moved outside the provider's premises.

In [25], the authors proposed a new trusted computing abstraction, *Policy-sealed data*, to resolve these problems. This abstraction allows customer data to be encrypted according to a customer chosen policy and guarantees that only those cloud nodes whose configuration, modeled as attributes, satisfies that policy can decrypt the data. They developed protocols using CP-ABE, which reduced the communication needs between the trusted monitor and production nodes. Their design allowed to implement a system that offered the policy-sealed data primitive with the help of commodity TPMs. They were able to validate the system under standard ABE measurement parameters such an policy generation, encryption and decryption overheads.

3.7 Forward-Secure Messaging

More recently, in [14], the authors used ABE to address the problem of forward-secure messaging. In this scenario, a user periodically changes her secret key, so that past messages sent over email or SMS remain confidential, in the event that her key is compromised or if the user does not want some parties to be able to read the messages after a designated period. An instance of such an application is the TextSecure protocol used by WhatsApp, which implements a highly fine-grained forward secrecy mechanism. The recently introduced "delete-for-all" feature falls in this category where access to some past messages can be revoked by the sender. An initial proposal for this problem was the forward secure public

key encryption scheme (FS-PKE) [10] FS-PKE describes an efficient update procedure by which a user's secret key can be altered to revoke decryption capability for any cipher-text encrypted during time period $T_{past} < T_{present}$. However, this mechanism does not provide fine-grained control of messages to be deleted, *i.e.*, all messages within a given period are deleted and user has no control on the selection of the messages.

In [14], the authors used a modification of FS-PKE combined with ABE capabilities to describe what is called as "punctured" encryption to achieve fine-grained control of revocation of messages. The approach is a form of tag-based encryption, which on input the current secret key SK and a tag $t \in \{0,1\}^*$, outputs a new secret key SK_0 that will decrypt all ciphertexts not encrypted under tag t. The key effectively "punctures" the decryption capability and this can be repeated many times to realize the capability of fine-grained control and normal message deletion. By combining the punctured encryption with FS-PKE the authors were able to implement a practical forward-secure public key encryption under real-life workloads of messages and revocations. This unified scheme ensures that an attacker who obtains the secrets for time period T and $T + 1$ cannot recombine any portions of the key to obtain access to messages deleted during an earlier time period. The experimental validation considered fixed amount of time (100,000 s) and chose parameters so that each public key of a user covers one year worth of intervals. The scheme was able to deal with message rates of one per second and decryption times of 20 ms, which is completely acceptable for any real-life messaging application.

The authors also showed that this scheme might be applicable to most scenarios where secure deletion is a concern. For instance, secure deletion of files in cloud based storage is a challenging problem and this scheme has possible applicability to it. Also, considering that the cloud storage inherently lends itself to ABE type of access control, such an implementation is more than likely to find wide-spread adoption. The key takeaway is that by combining the punctured encryption primitive with an FS-PKE scheme supported with ABE, the authors demonstrated the applicability of ABE in developing far more stronger cryptographic tools for wide-ranging applications.

3.8 Case Study of Commercial Products: Zeutro

The push for ABE commercialization is beginning to see the light with organizations like Zeutro® [20], which are building products for securing client data on cloud platforms. Zeutro has developed commercial-grade and robust attribute-based encryption toolkit (ZTK) to secure cloud applications while achieving fine-grained access control. They have developed Arethusa®, an advanced data protection and key management system for encrypting enterprise data-at-rest, which is shown in Fig. 3. Arethusa uses ABE to protect all data object and employs a centralized reference monitor to implement online access control.

Zeutro uses CP-ABE and KP-ABE to achieve different forms of access control. They use KP-ABE to achieve what is known as *Content-Based Access Control* wherein the attributes are derived from the content of the message.

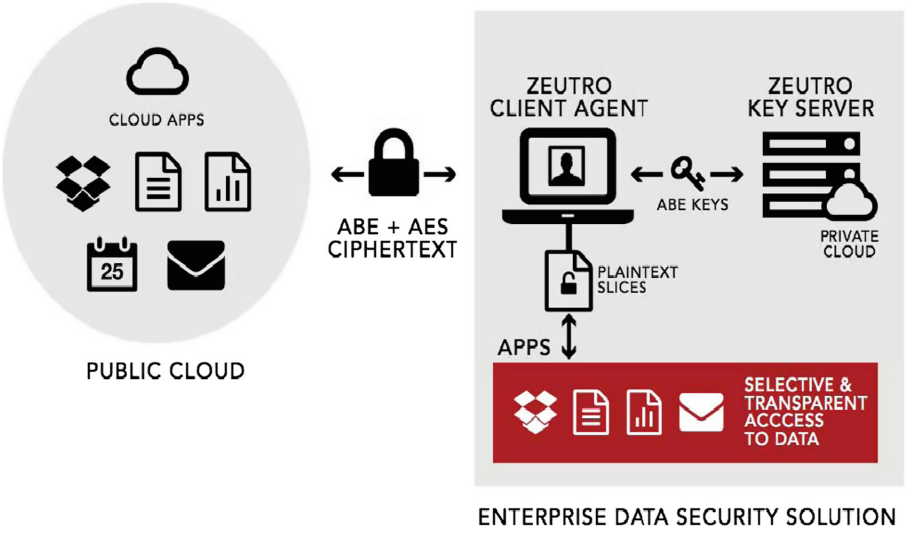

Fig. 3. The Arethusa system for protecting data-at-rest

For example, in a system that encrypts emails, the attributes can be "*To*:" and "*From*:" addresses and the body of the email is encrypted as the secret data. As possible in KP-ABE, the private (decryption) keys can be generated to identify the kind of cipher-text it can decrypt. They use CP-ABE to implement Role-based Access Control wherein the decryption capability of the user depends on her attributes and the cipher-text carries the corresponding policy. For example, one could restrict a ciphertext only to employees who have been with the company since 2012 and worked on "Project A" software project and where the other user attributes are defined as per the context of the operating environment.

4 Challenges and Future Directions

The challenges in deploying ABE arise mainly due to the encryption and decryption times, which are dependent on the number and size of the attributes being used. We give a brief overview of possible challenges faced.

4.1 Sizes of Attribute Sets

The encryption and decryption times of a given ABE system depends on the number and the domain of the attributes involved. Most earlier systems scale linearly with the number of attributes making them inefficient for real-world applications. However, in theory [1,2,4,18,23], some significant advances have been made to address this problem in what are known as the "large-universe" settings. In practice, as demonstrated by the approach in [28], such optimization attempts have been successful and this shows that the proper management

of attribute sets can result in usable ABE systems. However, this remains an interesting and open challenges for wide-scale deployment of ABE.

4.2 Attribute Structure

Existing ABE systems do have issues in using arbitrary strings as attributes. Often times, the attribute sets are constrained to be obtained from a fixed space. However, it is an open challenge to be able use any arbitrary string for achieving ABE.

4.3 Pairing Operations

There is need for a smaller number of pairings for decryption as this another way of scaling the ABE system. Often, the access structure and the decryption policy seems to dictate the number of pairings required. It may happen that due to a poor strategy even a small universe ABE might require sub-optimal number of pairings. An open challenge is to explore the strategy of decryption and/or devise efficient access structures that naturally bound the number of pairings required.

4.4 Secure Elliptic Curves

ABE depends on the availability of secure elliptic curves on which the hardness assumptions of the standard problems hold. There is a constant attempt to find curves that are not only secure but also support efficient pairing operations. NIST has made attempts to standardize the types of curves that can be used. However, this remains an open area of exploration in ABE as recent schemes [1,11] have shown the possibility of newer curves being used for ABE, that improve both security and efficiency.

5 Conclusion and Future Directions

In this survey, we described Attribute-Based Encryption (ABE) in detail and demonstrated its applicability in various scenarios. We have taken an application oriented view in this survey as ABE has received considerable attention in the community and already there have been attempts to commercialize these techniques. The goal of this survey is to encourage further ideas for deployment of ABE in real-world settings and to drive more innovations in existing systems. We have described some open challenges that hinder such attempts, but ABE has been resilient to these changes so far and lent itself to deployment across various applications. We will conclude by describing a couple of promising future directions for ABE.

There is considerable push for applying ABE in security for emerging domains, especially, the rapidly evolving Internet-of-Things [17]. In [17], the

authors devised an ABE policy framework, called Secure Identity-Based Broadcast Encryption (SIBBE), that allows a "task manager" to coordinate multiple devices working towards a common task and implement appropriate policy of data sharing across them. A powerful node called "commissioner" is in charge of policy management and revocation details. The authors showed that it is possible to use ABE and create a practical framework for securely managing the IoT devices.

Another interesting future direction is to check for the applicability of ABE for access control in more general sense, say like in XACML, that work for a large class of access control policies. In [12], the authors look at newer constructions based on more generalized secret sharing mechanisms than that of Shamir [26] and prove that it is indeed possible. This line of work marks a vast area of unexplored applications for ABE and scope for development of novel solutions to problems in many domains.

References

1. Agrawal, S., Chase, M.: FAME: fast attribute-based message encryption. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, 30 October–03 November 2017, pp. 665–682 (2017). https://doi.org/10.1145/3133956.3134014
2. Agrawal, S., Chase, M.: Simplifying design and analysis of complex predicate encryption schemes. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10210, pp. 627–656. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_22
3. Akinyele, J.A., Pagano, M.W., Green, M.D., Lehmann, C.U., Peterson, Z.N.J., Rubin, A.D.: Securing electronic medical records using attribute-based encryption on mobile devices. In: Proceedings of the 1st ACM Workshop Security and Privacy in Smartphones and Mobile Devices, Co-located with CCS, SPSM 2011, Chicago, IL, USA, 17 October, pp. 75–86 (2011). https://doi.org/10.1145/2046614.2046628
4. Attrapadung, N.: Dual system encryption via doubly selective security: framework, fully secure functional encryption for regular languages, and more. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_31
5. Baden, R., Bender, A., Spring, N., Bhattacharjee, B., Starin, D.: Persona: an online social network with user-defined privacy. SIGCOMM Comput. Commun. Rev. **39**(4), 135–146 (2009). https://doi.org/10.1145/1594977.1592585
6. Beimel, A.: Secret-sharing schemes: a survey. In: Chee, Y.W., et al. (eds.) IWCC 2011. LNCS, vol. 6639, pp. 11–46. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20901-7_2
7. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055718
8. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, S&P 2007, Oakland, California, USA, 20–23 May 2007, pp. 321–334 (2007). https://doi.org/10.1109/SP.2007.11

9. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
10. 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). https://doi.org/10.1007/3-540-39200-9_16
11. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_20
12. Crampton, J., Pinto, A.: Attribute-based encryption for access control using elementary operations. In: 2014 IEEE 27th Computer Security Foundations Symposium, pp. 125–139, July 2014
13. 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, CCS, Alexandria, VA, USA, 30 October–3 November 2006, pp. 89–98 (2006). https://doi.org/10.1145/1180405.1180418
14. Green, M.D., Miers, I.: Forward secure asynchronous messaging from puncturable encryption. In: IEEE Symposium on Security and Privacy, SP, San Jose, CA, USA, 17–21 May, pp. 305–320 (2015). https://doi.org/10.1109/SP.2015.26
15. Joux, A.: The weil and tate pairings as building blocks for public key cryptosystems. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 20–32. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45455-1_3
16. Joux, A., Nguyen, K.: Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups. J. Cryptol. **16**(4), 239–247 (2003)
17. Kim, J.Y., Hu, W., Sarkar, D., Jha, S.: ESIoT: Enabling secure management of the Internet of Things. In: Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, pp. 219–229. ACM, New York (2017)
18. Lewko, A., Waters, B.: Unbounded HIBE and attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_30
19. Liu, Z., Cao, Z., Wong, D.S.: Efficient generation of linear secret sharing scheme matrices from threshold access trees. Cryptology ePrint Archive: Listing (2010)
20. Zeutro LLC. http://www.zeutro.com/
21. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Proceedings of the ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, 28–31 October 2007, pp. 195–203 (2007). https://doi.org/10.1145/1315245.1315270
22. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_35
23. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, Berlin, Germany, 4–8 November 2013, pp. 463–474 (2013). https://doi.org/10.1145/2508859.2516672
24. Sahai, A., Waters, B.: Fuzzy identity based encryption. IACR Cryptology ePrint Archive 2004, 86 (2004). http://eprint.iacr.org/2004/086

25. Santos, N., Rodrigues, R., Gummadi, K.P., Saroiu, S.: Policy-sealed data: a new abstraction for building trusted cloud services. In: Proceedings of the 21st USENIX Security Symposium, Bellevue, WA, USA, 8–10 August, pp. 175–188 (2012). https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/santos
26. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979). https://doi.org/10.1145/359168.359176
27. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
28. Traynor, P., Butler, K.R.B., Enck, W., McDaniel, P.D.: Realizing massive-scale conditional access systems through attribute-based cryptosystems. In: Proceedings of the Network and Distributed System Security Symposium, NDSS, San Diego, California, USA, 10 February–13 February (2008). http://www.isoc.org/isoc/conferences/ndss/08/papers/06_realizing_massive-scale_conditional.pdf
29. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4
30. Yao, D., Fazio, N., Dodis, Y., Lysyanskaya, A.: Id-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, 25–29 October 2004, pp. 354–363 (2004). https://doi.org/10.1145/1030083.1030130
31. Yu, S., Wang, C., Ren, K., Lou, W.: Achieving secure, scalable, and fine-grained data access control in cloud computing. In: INFOCOM 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, San Diego, CA, USA, 15–19 March, pp. 534–542 (2010). https://doi.org/10.1109/INFCOM.2010.5462174
32. Zhou, Z., Huang, D.: On efficient ciphertext-policy attribute based encryption and broadcast encryption: extended abstract. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, pp. 753–755. ACM, New York (2010). https://doi.org/10.1145/1866307.1866420