1-1-2014

Attribute-based data transfer with filtering scheme in cloud computing

Jinguang Han
University of Wollongong, jh843@uowmail.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Yi Mu
University of Wollongong, ymu@uow.edu.au

Jun Yan
University of Wollongong, jyan@uow.edu.au

Attribute-based data transfer with filtering scheme in cloud computing

Abstract

Data transfer is a transmission of data over a point-to-point or point-to-multipoint communication channel. To protect the confidentiality of the transferred data, public-key cryptography has been introduced in data transfer schemes (DTSs). Data transfer is a transmission of data over a point-to-point or point-to-multipoint communication channel. To protect the confidentiality of the transferred data, public-key cryptography has been introduced in data transfer schemes (DTSs). Unfortunately, there exist some drawbacks in the current DTSs. First, the sender must know who the real receivers are. This is undesirable in a system where the number of the users is very large, such as cloud computing. In practice, the sender only knows some descriptive attributes of the receivers. Secondly, the receiver cannot be guaranteed to only receive messages from the legal senders. Therefore, it remains an elusive and challenging research problem on how to design a DTS scheme where the sender can send messages to the unknown receivers and the receiver can filter out false messages according to the described attributes. In this paper, we propose an attribute-based data transfer with filtering (ABDTF) scheme to address these problems. In our proposed scheme, the receiver can publish an access structure so that only the users whose attributes satisfy this access structure can send messages to him. Furthermore, the sender can encrypt a message under a set of attributes such that only the users who hold these attributes can obtain the message. In particular, we provide an efficient filtering algorithm for the receiver to resist the denial-of-service attacks. Notably, we propose the formal definition and security models for ABDTF schemes. To the best of our knowledge, it is the first time that a provable ABDTF scheme is proposed. Hence, this work provides a new research approach to ABDTF schemes. must know who are the real receivers. This is undesirable in a system where the number of the users is very large, such as cloud computing. In practice, the sender only knows some descriptive attributes of the receivers. Second, the receiver cannot be guaranteed to only receive messages from the legal senders. Therefore, it remains an elusive and challenging research problem on how to design a DTS scheme where the sender can send messages to the unknown receivers and the receiver can filter out false messages according to the described attributes. In this paper, we propose an attribute-based data transfer with filtering (ABDTF) scheme to address these problems. In our proposed scheme, the receiver can publish an access structure so that only the users whose attributes satisfy this access structure can send messages to him. Furthermore, the sender can encrypt a message under a set of attributes such that only the users who hold these attributes can obtain the message. In particular, we provide an efficient filtering algorithm for the receiver to resist the denial-of-service (DoS) attacks. Notably, we propose the formal definition and security models for ABDTF schemes. To the best of our knowledge, it is the first time that a provable ABDTF scheme is proposed. Hence, this work provides a new research approach to ABDTF schemes.

Keywords

attribute, filtering, scheme, cloud, computing, data, transfer

Disciplines

Engineering | Science and Technology Studies

Publication Details

Han, J., Susilo, W., Mu, Y. & Yan, J. (2014). Attribute-based data transfer with filtering scheme in cloud computing. The Computer Journal, 57 (4), 579-591.

Attribute-based Data Transfer with Filtering Scheme in Cloud Computing

Jinguang Han[1], Willy Susilo[1], Yi Mu[1] and Jun Yan[2]

[1] *Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, NSW2522, Australia.*
[2]*School of Information Systems and Technology, University of Wollongong, NSW2522, Australia.*
Email: jh843@uowmail.edu.au

Data transfer is a transmission of data over a point-to-point or point-to-multipoint communication channel. To protect the confidentiality of the transferred data, public-key cryptography has been introduced in data transfer schemes (DTSs). Unfortunately, there exist some drawbacks in the current DTSs. First, the sender must know who are the real receivers. This is undesirable in a system where the number of the users is very large, such as cloud computing. In practice, the sender only knows some descriptive attributes of the receivers. Second, the receiver cannot be guaranteed to only receive messages from the legal senders. Therefore, it remains an elusive and challenging research problem on how to design a DTS scheme where the sender can send messages to the unknown receivers and the receiver can filter out false messages according to the described attributes. In this paper, we propose an attribute-based data transfer with filtering (ABDTF) scheme to address these problems. In our proposed scheme, the receiver can publish an access structure so that only the users whose attributes satisfy this access structure can send messages to him. Furthermore, the sender can encrypt a message under a set of attributes such that only the users who hold these attributes can obtain the message. In particular, we provide an efficient filtering algorithm for the receiver to resist the denial-of-service (DoS) attacks. Notably, we propose the formal definition and security models for ABDTF schemes. To the best of our knowledge, it is the first time that a provable ABDTF scheme is proposed. Hence, this work provides a new research approach to ABDTF schemes.

Keywords: Data Transfer, Access Control, Attribute-Based Encryption, Filtration, Cloud Computing

Received ; revised

1. INTRODUCTION

Cloud computing provides computing resources (software and hardware) as a service over network or in the data center. This is called as Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) [1, 2]. Although it has brought benefits to users including availability, cost saving and reliability, the confidentiality and efficiency of the transferred and stored data in cloud computing have been the primary focus of them. In open communication environments, to provide users with a secure communication channel and avoid sharing a session key prior to the communication, public-key cryptography has been addressed [3, 4, 5, 6, 7, 8]. In these schemes, in order to send sensitive data to the intended receivers, the sender must know all the identities (or public keys) of the receivers and communicate with them separately [3, 4, 5, 6]. Furthermore, since anyone who knows the identity of the receiver can send messages to him, the receiver cannot determine whether those messages are from the legal sender [5, 6, 7, 8]. These problems are particularly serious in cloud computing. To clarify these issues, we provide the following scenario. Due to the large number of users in cloud computing, each user is unable to know and communicate with all the other users in the system. In the scenario, suppose a user U would like to purchase a personal computer $PC = \{Brand = Apple,\ Year = 2011\}$, he must set conversations with the multiple unknown sellers. The best solution is that the user specifies an access structure such that only the sellers whose product attributes satisfy this access structure can contact him and negotiate with him. This system will not only reduce the communication cost, but also protect the user's privacy. Additionally, if a seller sells the machines $PC = \{Brand = Apple,\ Price \leq 5000,\ Type = Student,\ Year = 2011\}$,

he will not do any deal with the buyer who is *not* a student. Or else, he will face the furious denial-of-service (DoS) attacks [9, 10, 11, 12], as any user who does not hold the required attributes can also contact him. DoS attacks are initialized by malicious adversaries to consume the resource of the host or network such that the legal users cannot be serviced. There are two kinds of attacks [9]: logic attacks and flooding attacks. In the logic attacks, the adversaries use the flaws in the current software to degrade its performance. In the flood attacks, the adversaries send or inject lots of false messages to consume the user's resource or paralyze the system. We note that filtering schemes can be used as efficient primitives to resist the DoS attacks. Before processing the received messages, the receiver can efficiently filter out the false ones. One of the efforts to improve the security of cloud computing is to protect against DoS attacks [13, 14].

In this paper, we introduce a filtering scheme to an attribute-based data transfer scheme to protect the sender's privacy and save the receiver from the DoS attacks.

1.1. Related Work

In this section, we review the work related to our ABDTF scheme.

1.1.1. Data Transfer with filtering schemes

Filtering in DTS schemes is an efficient tool to help the receiver filter out the false data [15, 16, 17, 18, 19]. Furthermore, it has been used to protect against the DoS attacks.

Bloom [15] proposed a filtering scheme based on the hashing-code methods to detect the membership in a given set of messages. Subsequently, Mitzenmacher [16] proposed a compressed Bloom filter to improve the preference and transmission of Bloom's scheme.

Little [17] proposed an efficient algorithm for nonrecursive and recursive digital filters, where the filtering speed is related to the memory space and the time is independent of the order of the filter. Yuen [19] improved Little's scheme by representing the data in two complement forms, instead of the *biased* form.

Filtering schemes used to filter out the false report in the wireless sensor networks (WSN) have been proposed [3, 4, 20, 21, 22, 23]. To name a few, Ye, Luo, Lu and Zhang [20] proposed a statistical en-route filtering scheme to filter out the false report during the forwarding process in sensor networks. In this scheme, each sensor generates a keyed message authentication code (MAC). For an event report, multiple MACs are attached to it. As the report is forwarded, the sensor verifies the correctness of the MACs probabilistically and detects the false report.

Zhu, Setia, Jajodia and Ning [3] proposed an interleaved hop-by-hop authentication scheme where the false report can be detected by the base station (sink) if no more than a certain number of sensors are compromised . They also provided an upper bound for the number of hops that a false report can be forwarded prior to being detected if the compromised sensors are under the certain number.

Yang, Ye, Yuan, Lu and Arbaugh [21] proposed a location-based approach where the key is bound to the geographic location to resist the compromised sensors to compute the false report. Ren, Lou and Zhang [22] proposed a location-aware end-to-end data transfer scheme to provide the end-to-end security and filter out the false report. Both [21] and [22] use the symmetric key systems where each sensor must share a key with his upper and lower sensors. Zhang, Liu, Lou and Fang [23] proposed a location-based compromise-tolerant mechanism based on public-key systems to detect the false report.

Yu and Guan [24] proposed a dynamic en-route filtering scheme where each sensor holds a keyed hash chain to validate the report. They used the *hill climbing* key dissemination to guarantee that the sensor close to the sink has strong filtering ability, and broadcast property to resist the DoS attacks.

Lu, Lin, Zhu, Liang and Shen [4] proposed a bandwidth-efficient cooperative authentication mechanism with filtering. They introduced the random graph characteristics of sensor nodes and the cooperative bit-compressed authentication technique to WSN to save the energy of detecting the false report and reduce the burden of the sink.

1.1.2. Attribute-based Encryption (ABE)

Attribute-based encryption (ABE) was introduced by Sahai and Waters [7], where both the secret key and the ciphertext are attached with a set of attributes. The user can decrypt a ciphertext if there is a match between the attributes which he holds and the attributes listed in the ciphertext. The original idea of ABE was to design a fuzzy (error-tolerant) identity-based encryption (IBE) scheme [5, 6, 25, 26, 27]. Currently, there are two kinds of ABE schemes:

Key-policy ABE (KP-ABE). In this scheme, the ciphertext is attached with a set of attributes, while the secret key is associated with an access structure [7, 28, 29, 30, 31, 32, 33].

Cipher-Policy ABE (CP-ABE). In this scheme, the ciphtertext is associated with an access structure, while the secret key is attached with a set of attributes [8, 34, 35, 36, 37].

Access structure has been deployed to restrict an unauthorized user to access sensitive data. An access structure is monotonic if a set A satisfies the access structure and $A \subseteq B$, then B satisfies the access structure. A (k, n)-threshold access structure is a special access structure where a secret is divided into n shares. A user can reconstruct the secret if and

only if he obtains at least k shares. In a KP-ABE scheme, the central authority (CA) can specify a (k, n)-threshold access structure and issues secret keys to users according this access structure. An encrypter can encrypt a message under the k-out-of-n public attributes. Consequently, if a user holds the required k attributes, he can use his secret keys to decrypt the ciphertext.

In Sahai-Water's seminal scheme [7], (k, n)-threshold access structure was presented. In order to express a more complex access structure, Goyal, Pandey, Sahai and Waters [28] proposed a new KP-ABE scheme with a fine-grained access control where any access structure can be expressed using the *access tree* technique. An access tree is a tree where each leaf node represents an attributes and each non-leaf node δ consists of a threshold gate (k_δ, n_δ), where n_δ is the number of the children of the node δ and $k_\delta \leq n_\delta$. If $k_\delta = 1$, it is an OR gate. However, if $k_\delta = n_\delta$, it is an AND gate. A secret is embedded in the root node. If the attributes of a user satisfy the access structure, he can use his secret keys to reconstruct the secret. Then, Waters [8] proposed a CP-ABE scheme where any access structure can be expressed with a linear secret sharing scheme (LSSS) [38].

Attrapadung and Imai [39] proposed a dual-policy ABE scheme which combines KP-ABE scheme with CP-ABE scheme. In this scheme, there are two access structures: one is for the *subjective* attributes which are attached to the secret key and the other is for the *objective* attributes which are ascribed to the ciphertext. A user can decrypt the ciphertext if his attributes satisfy the first access structure and the attributes list in the ciphertext satisfy the second one. Nevertheless, both the CA and the sender can specify an access structure in this scheme, and hence, the receiver has no control on the received messages which may not be useful to him. Thus, this scheme may be susceptible against the DoS attacks.

Due to its ability to express a flexible access structure and support one-to-many communication, ABE has been used to design DTS in various large database systems, such as wireless sensor networks [40] and cloud computing [2].

Security Model. ABE scheme should be secure against the collusion attacks [7], namely no group of users can combine their secret keys to decrypt the ciphertext which none of them can decrypt by himself. The original ABE scheme [7] is secure under the selective-set model where the adversary must submit a set of attributes on which he wants to be challenged before he can obtain the public parameters. Lewko, Okamoto, Sahai, Takashima and Waters [37] proposed a stronger security model called adaptive security model where the adversary can get the public parameters directly without the above mentioned limitations.

1.2. Our Contribution

Since ABE does not depend on the public-key infrastructure (PKI), it has been used as a building block to design DTS in the distributed systems. However, distributed systems are susceptible against the DoS attacks. These attacks consume users' resources, and furthermore it may paralyze the system. Therefore, it is a challenging work to design an attribute-based DTS which can resist the DoS attacks. In this paper, we propose the formal definition and security models for attribute-based data transfer with filtering (ABDTF) scheme. Then, an efficient ABDTF is proposed and proven to be secure in the proposed security models under the standard complexity assumptions (DBDH and CBDH). In our scheme, the sender can encrypt a message under a set of attributes such that only the receiver who holds these attributes can obtain the message. Additionally, the receiver can also broadcast an access structure such that only the sender whose attributes satisfy this access structure can send data to him. Prior to processing the received data, the receiver can efficiently verify whether it comes from the legal senders. Note that the receiver can update his access structure dynamically without the need of re-initializing the system and re-issuing the secret keys to the users. Furthermore, the authentication key stored by the receiver and the authentication information from the sender are short. The authentication key and the authentication information can be computed off-line by the receiver and the sender, respectively. We also implement any access structure by using the access tree technique in [28]. To the best of our knowledge, it is the first time that a provable ABDTF scheme is proposed. Therefore, our work provides a formal treatment on the research of ABDTF schemes. We implement our scheme in the PBC library [41].

1.3. Organization of The Paper

The remainder of this paper is organized as follows. In Section 2, we review the preliminaries which are used throughout this paper. We propose an ABDTF scheme and prove its security in Section 3. In Section 4, we implement our scheme in the PBC library. Finally, Section 5 concludes this paper.

2. PRELIMINARIES

In this section, we review the preliminaries used throughout this paper.

In the rest of this paper, we denote $x \xleftarrow{R} X$ as x is selected randomly from X. If X is a finite set, by $x \xleftarrow{R} X$ and $|X|$, we denote that x is selected from X uniformly and the cardinality of X, respectively. By $A(x) \rightarrow y$, we denote y is computed by running the algorithm A on input x. We say that a function $\epsilon : \mathbb{Z} \rightarrow \mathbb{R}$ is negligible if for all $k \in \mathbb{Z}$, there exists a $z \in \mathbb{Z}$ such

that $\epsilon(x) < \frac{1}{x^k}$ for all $x > z$. By $p(x) \xleftarrow{R} \mathbb{Z}_p[x]$, we denote the polynomial $p(x)$ is randomly selected from the polynomial ring $\mathbb{Z}_p[x]$ consisting of the polynomials that coefficients are from the finite field $\mathbb{Z}_p$.

Lagrange Interpolation. Let $p(x) \xleftarrow{R} \mathbb{Z}_p[x]$ be a $(k-1)$ degree polynomial. Given any k different values $p(x_1)$, $p(x_2)$, $\cdots$, $p(x_k)$, the polynomial $p(x)$ can be reconstructed as follows:

$$p(x) = \sum_{x_i \in S} p(x_i) \prod_{x_j \in S, x_j \neq x_i} \frac{x - x_j}{x_i - x_j} = \sum_{x_i \in S} p(x_i) \Delta_{S,x_i}(x)$$

where $S = \{x_1, x_2, \cdots, x_k\}$. The Lagrange coefficient for x_i in S is defined as

$$\Delta_{S,x_i}(x) = \prod_{x_j \in S, x_j \neq x_i} \frac{x - x_j}{x_i - x_j}.$$

Consequently, given any k different polynomial values, we can compute $p(x)$ for $\forall x \in \mathbb{Z}_p$. Additionally, the other polynomial values are unconditionally secure if only $k-1$ different polynomial values are given.

2.1. Attribute-based Data Transfer with Filtering (ABDTF)Scheme

In our attribute-based data transfer with filtering scheme, access structures are employed to control the receivers and senders. We define an access structure as follows.

Definition 2.1. (Access Structure) [38]. *Let $\mathbb{P} = \{P_1, P_2, \cdots, P_n\}$ be a set of parties. We say that a collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ is monotonic if $S_1 \subseteq S_2$ and $S_1 \in \mathbb{A}$ implies $S_2 \in \mathbb{A}$. An access structure (respectively monotonic access structure) is a collection (respectively monotonic collection) $\mathbb{A}$ of the non-empty subset of $2^{\{P_1, P_2, \cdots, P_n\}}$, namely $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}} \setminus \{\phi\}$. We call the set in $\mathbb{A}$ as the authorized set, and the set outside $\mathbb{A}$ as unauthorized set.*

In this paper, we use monotonic access structures.

An ABDTF scheme consists of the following five algorithms:

- $\mathsf{Setup}(1^\ell, \mathcal{U}) \rightarrow (params, MSK)$. This algorithm takes as input a security parameter 1^ℓ and the universal attribute set $\mathcal{U}$, and outputs the public parameters $params$ and the master secret key MSK.
- $\mathsf{KeyGen}(params, MSK, A_U) \rightarrow SK_U$. This algorithm takes as input the public parameters $params$, the master secret key MSK and a set of attributes A_U, and outputs a secret key SK_U.
- $\mathsf{Receiver\text{-}Policy}(params, \mathcal{R}) \rightarrow (\mathbb{A}_R, AK)$. This algorithm takes as input the public parameters $params$ and a set of attributes $\mathcal{R}$, and outputs an access structure $\mathbb{A}_R$ and an authentication key AK.
- $\mathsf{Enc}(params, M, A_C, \mathbb{A}_R, SK_E) \rightarrow (CT, AI)$. This algorithm takes as input the public parameters $params$, a message M, a set of attributes A_C, an access structure $\mathbb{A}_R$ and the encrypter's secret key SK_E, and outputs the ciphertext CT and the authentication information AI. This ciphertext can be decrypted by the user who holds a set of attributes A_U if $A_C \subseteq A_U$.
- Decryption
  1. $\mathsf{Filter}(AK, AI) \rightarrow True/False$. This algorithm takes as input the authentication key AK and the authentication information AI, and outputs $True$ if the attributes of the encrypter A_E satisfy the access structure $\mathbb{A}_R$. Otherwise, it outputs $False$ and aborts the protocol.
  2. $\mathsf{Dec}(params, SK_U, CT) \rightarrow M$. This algorithm takes as input the public parameters $params$, the secret key SK_U and the ciphertext CT, and outputs the message M.

Definition 2.2. *We say that an attribute-based data transfer with filtering (ABDTF) scheme is correct if*

$$\Pr\left[\begin{array}{l|l} \mathsf{Dec}(params, SK_U, CT) = M & \begin{array}{l}\mathsf{Setup}(1^\ell, \mathcal{U}) \rightarrow (params, MSK);\\ \mathsf{KeyGen}(params, MSK, A_U) \rightarrow SK_U;\\ \mathsf{Receiver-Policy}(params, \mathcal{R}) \rightarrow (\mathbb{A}_R, AK);\\ \mathsf{Enc}(params, M, A_C, \mathbb{A}_R, SK_E) \rightarrow (CT, AI);\\ \mathsf{Filter}(AK, AI) \rightarrow True\\ A_C \subseteq A_U\end{array}\end{array}\right] = 1$$

where the probability is taken over the random coins which are consumed by the algorithms in the system.

2.2. Security Models

With respect to the security of the ABDTF scheme, we use the following two games to formalize it. These games are played by a challenger and an adversary. The first game is about the security of the ciphertext, which is similar to the selective-set security model in [7]. The second game is about filtration security. This game is used to formalize the model that only the sender whose attributes satisfy the access structure specified by the receiver can send messages to him.

Game 1: Selective-Set Security Model.

Initialization. The adversary $\mathcal{A}$ submits a set of attributes A_Υ on which he wants to be challenged.

Setup. The challenger runs $\mathsf{Setup}(1^\ell, \mathcal{U})$ to generate the system parameters $(params, MSK)$, and responses $\mathcal{A}$ with $params$.

Phase 1. $\mathcal{A}$ can adaptively query secret keys for sets of attributes $A_{U_1}, A_{U_2}, \cdots, A_{U_{q_1}}$, where the only restriction is $A_\Upsilon \nsubseteq A_{U_i}$ for $i = 1, 2, \cdots, q_1$. The challenger runs $KeyGen(params, MSK, A_{U_i})$ to generate a secret key SK_{U_i} for A_{U_i}, and responds $\mathcal{A}$ with SK_{U_i}, where $i = 1, 2, \cdots, q_1$.

Challenge. $\mathcal{A}$ submits two message M_0 and M_1 with the equal length. The challenger flips an unbiased coin with $\{0, 1\}$, and gets $\varrho \in \{0, 1\}$. He runs $\mathsf{Enc}(params, M_\varrho, A_\Upsilon, \cdot, \cdot)$ to generate the ciphertext CT^*, and responds $\mathcal{A}$ with CT^*.

Phase 2. $\mathcal{A}$ can adaptively query secret keys for sets of attributes $A_{U_{q_1+1}}, A_{U_{q_1+2}}, \cdots, A_{U_q}$, where $A_\Upsilon \nsubseteq A_{U_j}$ for $j = q_1 + 1, q_1 + 2, \cdots, q$. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess ϱ' on ϱ. $\mathcal{A}$ win the game if $\varrho' = \varrho$.

DEFINITION 2.3. *An attribute-based data transfer with filtering (ABDTF) scheme is $(T, q, \epsilon(\ell))$-secure in the selective-set security model if no probabilistic polynomial-time adversary $\mathcal{A}$ who queries secret keys for at most q sets of attributes has the advantage*

$$Adv_{\mathcal{A}} = \left| \Pr[\varrho' = \varrho] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above game.

Game 2: Filtration Security Model.

Initialization. The adversary $\mathcal{A}$ submits a set of attributes A_Ψ on which he wants to be challenged.

Setup. The challenger runs $\mathsf{Setup}(1^\ell, \mathcal{U})$ to generate $(params, MSK)$, and responds $\mathcal{A}$ with $params$.

Phase 1. $\mathcal{A}$ can adaptively query authentication information for access structures $\mathbb{A}_{R_1}, \mathbb{A}_{R_2}, \cdots, \mathbb{A}_{R_{q_1}}$, where the only constraint is $A_\Psi \notin \mathbb{A}_{R_i}$ for $i = 1, 2, \cdots, q_1$. The challenger runs $\mathsf{Enc}(params, \cdot, \cdot, \mathbb{A}_{R_i}, SK_{U_i})$ to generate an authentication information AI_i for $\mathbb{A}_{R_i}$, where $A_{U_i} \in \mathbb{A}_{R_i}$. He responses $\mathcal{A}$ with AI_i for $i = 1, 2, \cdots, q_1$.

Challenge. The challenger runs $\mathsf{Receiver\text{-}Policy}(params, A_\Psi)$ to generate $(\mathbb{A}_\Psi, AK_\Psi)$, and responds $\mathcal{A}$ with $\mathbb{A}_\Psi$.

Phase 2. $\mathcal{A}$ can query authentication information for access structures $\mathbb{A}_{R_{q_1+1}}, \mathbb{A}_{R_{q_1+2}}, \cdots, \mathbb{A}_{R_q}$, where the only constraint is $A_\Psi \notin \mathbb{A}_{R_i}$ for $i = q_1 + 1, q_1 + 2, \cdots, q$. Phase 1 is repeated.

Output. $\mathcal{A}$ outputs an authentication information AI_Ψ for the access structure $\mathbb{A}_\Psi$. $\mathcal{A}$ wins the game if $\mathsf{Filter}(AK_\Psi, AI_\Psi) \rightarrow True$.

DEFINITION 2.4. *An attribute-based data transfer with filtering (ABDTF) scheme is $(T, q, \epsilon(\ell))$-secure in the filtration security model if no probabilistic polynomial-time adversary $\mathcal{A}$ who queries authentication information for at most q sets of attributes has the advantage*

$$Adv_{\mathcal{A}} = \Pr[\mathsf{Filter}(AK, AI_\Upsilon) \rightarrow True] \geq \epsilon(\ell)$$

in the above game.

2.3. Complexity Assumption

Let $\mathbb{G}$ and $\mathbb{G}_\tau$ be two cyclic groups with prime order p, and g be a generator of the group $\mathbb{G}$. A bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ is a map with following properties.

1. Bilinearity. For any $a, b \in \mathbb{Z}_p$ and $\mu, \nu \in \mathbb{G}$, $e(\mu^a, \nu^b) = e(\mu, \nu)^{ab}$.
2. Non-degeneracy. $e(g, g) \neq 1$, where 1 is the identity of the group $\mathbb{G}_\tau$.
3. Computability. There exists an efficient algorithm to compute $e(\mu, \nu)$ for all $\mu, \nu \in \mathbb{G}$.

We denote $\mathcal{GG}(1^\ell)$ as a bilinear group generator which takes as inputs a security parameter 1^ℓ and outputs a bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ with prime order p and a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$.

DEFINITION 2.5. (Computational Bilinear Diffie-Hellman (CBDH) Assumption [6]) *Let $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and g be a generator of $\mathbb{G}$. We say that the CBDH assumption holds in $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ if no probabilistic polynomial-time adversaries $\mathcal{A}$ can compute $e(g, g)^{abc}$ from $(A, B, C) = (g^a, g^b, g^c)$ with advantage*

$$Adv_{\mathcal{A}}^{CBDH} = \Pr[\mathcal{A}(A, B, C) \rightarrow e(g, g)^{abc}] \geq \epsilon(\ell)$$

where the probability is taken over the random choices of $a, b, c \overset{R}{\leftarrow} \mathbb{Z}_p$ and the bits consumed by $\mathcal{A}$.

DEFINITION 2.6. (Decisional Bilinear Diffie-Hellman (DBDH) Assumption [27]) *Let $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and g be a generator of $\mathbb{G}$. We say that the DBDH assumption holds in $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ if no probabilistic polynomial-time adversaries $\mathcal{A}$ can distinguish $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{abc})$ from $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ with advantage*

$$Adv_{\mathcal{A}}^{DBDH} = \left| \begin{array}{l} \Pr[\mathcal{A}(A, B, C, e(g, g)^{abc}) = 1] \\ -\Pr[\mathcal{A}(A, B, C, e(g, g)^z) = 1] \end{array} \right| \geq \epsilon(\ell)$$

where the probability is taken over the random choices of $a, b, c, z \overset{R}{\leftarrow} \mathbb{Z}_p$ and the bits consumed by $\mathcal{A}$.

3. ATTRIBUTE-BASED DATA TRANSFER WITH FILTERING SCHEME

In this section, we propose an attribute-based data transfer with filtering (ABDTF) scheme. Then, the scheme is proven to be secure in the proposed security models. Finally, we analyze the computation cost and communication cost of our scheme.

3.1. Overview

A DTS scheme should provide the properties: confidentiality, integrity and authentication [42]. To design an ABDTF scheme, we introduce a filtering scheme to a KP-ABE scheme. In our scheme, there is a central authority (CA) who monitors the services provided by the cloud server and issues secret keys to users [43]. At first, CA specifies a (k, n)-threshold access structure $\mathbb{A}$. Then, all users interact with the CA to obtain secret keys for his attributes. Suppose that a receiver and a sender hold sets of attributes A_R and A_S, respectively. To resist the DoS attacks, the receiver can specify a (k, ρ)-threshold access structure $\mathbb{A}_R$ such that only the user (sender) whose attributes satisfies $\mathbb{A}_R$ can send messages to him, where $\rho \leq n$. The receiver stores an authentication key K for $\mathbb{A}_R$ and encapsulates K in $\mathbb{A}_R$. If the sender wants to send a message to the receiver, he must check whether A_S satisfies $\mathbb{A}_R$. If A_S satisfies $\mathbb{A}_R$ ($A_S \in \mathbb{A}_R$), the sender can use his secret keys to reconstruct the authentication key K. Consequently, the sender encrypts the message under a set of attributes A_C ($A_C \subseteq A_R$) and computes a keyed message authentication code (MAC) which is the hash value of the authentication key K and the ciphertext CT. The sender sends the ciphertext CT and the authentication information MAC to the receiver. Receiving (MAC, CT), the receiver checks the MAC by using the stored key K and the ciphertext CT. If the MAC is correct, the receiver checks whether he holds the attributes listed in the ciphertext and uses his secret keys to decrypt the ciphertext. Otherwise, the receiver treats the received (MAC, CT) as a false message and aborts. Therefore, the filtering algorithm in our scheme is based on the MAC technique. We explain our model in Fig.1

3.2. Our Scheme

We describe our attribute-based data transfer with filtering (ABDTF) scheme in Fig.2.

Correctness. The scheme described in Fig.2 is correct. Because, we have

$$\begin{aligned} F_S &= e(D_S, W) \\ &= e(g^{\alpha} h^{\sigma_s}, g^w) \\ &= e(g,g)^{\alpha w} \cdot e(g,h)^{w\sigma_s}, \end{aligned}$$

$$\begin{aligned} F_{v_j} &= e(D_{S,v_j}, E_{v_j}) \\ &= e(h^{\frac{p(a_{v_j})}{t_{v_j}}}, g^{w t_{v_j}})^{\Delta_{\mathcal{Q},a_{v_j}}(0)} \\ &= e(g,h)^{w p(a_{v_j}) \Delta_{\mathcal{Q},a_{v_j}}(0)}, \end{aligned}$$

$$\begin{aligned} e(D_R, C_1) &= e(g^{\alpha} h^{\sigma_r}, g^s) \\ &= e(g,g)^{\alpha s} \cdot e(g,h)^{s\sigma_r} \end{aligned}$$

and

$$\begin{aligned} & e(D_{R,x}, C_x)^{\Delta_{A_C,a_x}(0)} \\ &= e(h^{\frac{p(a_x)}{t_x}}, g^{s t_x})^{\Delta_{A_C,a_x}(0)} \\ &= e(g,h)^{s p(a_x) \Delta_{A_C,a_x}(0)}. \end{aligned}$$

Therefore,

$$\begin{aligned} F &= \frac{F_S}{\prod_{a_{v_j} \in \mathcal{Q}} F_{v_j}} \\ &= \frac{e(g,g)^{\alpha w} e(g,h)^{w\sigma_s}}{\prod_{a_{v_j} \in \mathcal{Q}} e(g,h)^{w p(a_{v_j}) \Delta_{\mathcal{Q},v_j}(0)}} \\ &= \frac{e(g,g)^{\alpha w} e(g,h)^{w\sigma_s}}{e(g,h)^{w \sum_{a_{v_j} \in \mathcal{Q}} p(a_{v_j}) \Delta_{\mathcal{Q},a_{v_j}}(0)}} \\ &= \frac{e(g,g)^{\alpha w} e(g,h)^{w\sigma_s}}{e(g,h)^{w\sigma_s}} \\ &= e(g,g)^{\alpha w} \end{aligned}$$

and

$$\begin{aligned} & C_0 \cdot \frac{\prod_{a_x \in A_C} e(D_{R,x}, C_x)^{\Delta_{A_C,a_x}(0)}}{e(D_R, C_1)} \\ &= C_0 \cdot \frac{\prod_{a_x \in A_C} e(g,h)^{s p(a_x) \Delta_{A_C,a_x}(0)}}{e(g,g)^{\alpha s} \cdot e(g,h)^{s\sigma_r}} \\ &= C_0 \cdot \frac{e(g,h)^{s \sum_{a_x \in A_C} p(a_x) \Delta_{A_C,a_x}(0)}}{e(g,g)^{\alpha s} \cdot e(g,h)^{s\sigma_r}} \\ &= C_0 \cdot \frac{e(g,h)^{s\sigma_r}}{e(g,g)^{\alpha s} \cdot e(g,h)^{s\sigma_r}} \\ &= M \cdot \frac{e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}} \\ &= M. \end{aligned}$$

In our scheme, both the computation costs of the access structure $\mathbb{A}_R$ from the receiver and the authentication information AI from the sender are linear with the number of the required attributes. However, $(K,\ W,\ \{E_{j_c}\}_{a_{j_c} \in R})$ and $(F,\ F_S,\ \{F_{v_j}\}_{a_{v_j} \in \mathcal{Q}})$ can be computed by the receiver and the sender off-line. Note that, the receiver can update the access structure $\mathbb{A}_R$ dynamically without re-initializing the system and re-issuing secret key to the users.

In the practical scenario, the filtration can be done by a proxy server. The receiver can determine an access structure, compute the authentication key K and delegate it to the proxy server. When a message (Γ, CT) arrives, the proxy server checks $\Gamma \stackrel{?}{=} H(K||CT)$. If so, he sends the message to the receiver. Otherwise, he filters it out on behalf of the receiver. This is especially necessary in the e-mail systems [5, 44, 45, 46]. The filter can help the user filter out the junk e-mail and reduce jams. Since the authentication key K in our scheme is only one element (512 bites) in the bilinear group, it can be stored in the software with a limited memory space.

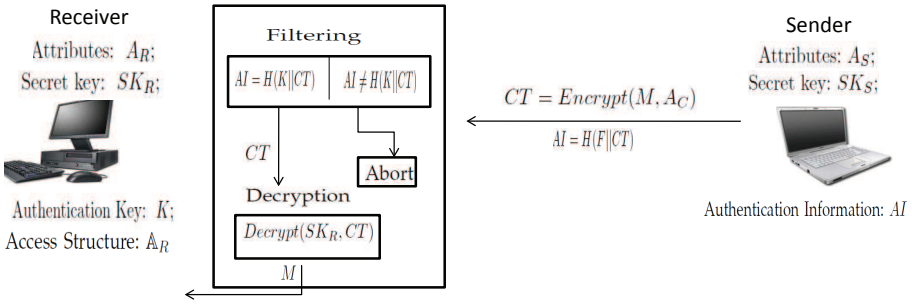

FIGURE 1. The Model of Attribute-based Data Transfer With Filtering Scheme

We provide the comparison of the computation cost and communication cost of our ABDTF scheme with the existing and related schemes in the literature in Table1 and Table 2, where by E, P and T_H, we denote the running time of executing one exponentiation, one paring and one hash function, respectively. By $E_{\mathbb{G}}$ and $E_{\mathbb{G}_\tau}$, we denote the length of one element in $\mathbb{G}$ and one element in $\mathbb{G}_\tau$, respectively. By ℓ_S, ℓ_O and $\ell_{S,max}$, we denote the number of the rows of the matrix used to encrypt messages, the number of the rows of the matrix used to generate secrete keys for users and the maximum number of rows of the matrix used to generate secrete keys for users in [39], respectively. By $--$, we denote that the item is not suitable for the scheme.

Key management in attribute-based systems is a tricky issue, especially key revocation, as multiple users might satisfy the access structure. The common method to deal with this problem is to append to each of the attributes an expiration day [31, 34].

3.3. Security Analysis

THEOREM 3.1. *Our attribute-based data transfer with filtering (ABDTF) scheme is $(T, q, \epsilon(\ell))$ secure in the selective-set security model if the $(T', \epsilon'(\ell))$ decisional bilinear Diffie-Hellman (DBDH) assumption holds in $(e, p, \mathbb{G}, \mathbb{G}_\tau)$, where*

$$T' = \mathcal{O}(T) \quad \textit{and} \quad \epsilon'(\ell) = \frac{\epsilon(\ell)}{2}.$$

Proof. If there exists an adversary $\mathcal{A}$ that can $(T, q, \epsilon(\ell))$ break the ciphertext security of our ABDTF scheme, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the DBDH assumption as follows.

The challenger creates the bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$. Let g be a generator of the group $\mathbb{G}$. He flips an unbiased coin with $\{0, 1\}$, and obtains a bit $\varrho \in \{0, 1\}$. If $\varrho = 0$, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{abc})$ to the algorithm $\mathcal{B}$. Otherwise, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ to $\mathcal{B}$, where $z \xleftarrow{R} \mathbb{Z}_p$. $\mathcal{B}$ will outputs his guess ϱ' on ϱ.

Initialization. The adversary $\mathcal{A}$ submits a set of attributes A^* on which he want to be challenged.

Setup. $\mathcal{B}$ sets $Y = e(g, g)^{ab}$ and $h = Ag^\eta$, where $\eta \xleftarrow{R} \mathbb{Z}_p$. If $a_i \in A^*$, he chooses $t_i \xleftarrow{R} \mathbb{Z}_p$ and computes $T_i = g^{t_i}$. Otherwise he chooses $t_i \xleftarrow{R} \mathbb{Z}_p$ and computes $T_i = h^{t_i} = g^{t_i(a+\eta)}$. The public parameters are

$$(e, p, \mathbb{G}, \mathbb{G}_\tau, Y, T_1, T_2, \cdots, T_n).$$

The implicit master secret key is

$$(ab, \{t_i\}_{a_i \in A^*}, \{t_i(a + \eta)\}_{a_i \notin A^*}).$$

$\mathcal{B}$ sends $(e, p, \mathbb{G}, \mathbb{G}_\tau, Y, T_1, T_2, \cdots, T_n)$ to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ queries secret key for a set of attributes $\hat{A}$, where $A^* \nsubseteq \hat{A}$. Suppose that $\hat{A} \bigcap A^* = \{a_{i_1}, a_{i_2}, \cdots, a_{i_l}\}$, where $l < k$. $\mathcal{B}$ chooses $r, y_{i_1}, y_{i_2}, \cdots, y_{i_l}, \cdots, y_{i_{k-1}} \xleftarrow{R} \mathbb{Z}_p$, and computes

$$\hat{D} = B^{-\eta} h^r \tag{1}$$

$$\{\hat{D}_{i_j} = h^{\frac{y_{i_j}}{t_{i_j}}}\}_{a_{i_j} \in A^*} \tag{2}$$

and

$$\{\hat{D}_{\lambda_i} = (B^{-1} g^r)^{\frac{\Delta_{\Omega,0}(a_{\lambda_i})}{t_{\lambda_i}}} \prod_{j=1}^{k-1} g^{\frac{y_{i_j} \Delta_{\Omega,a_{i_j}}(a_{\lambda_i})}{t_{\lambda_j}}}\}_{a_{\lambda_i} \in \hat{A} - A^*} \tag{3}$$

where $\Omega = \{a_{i_1}, a_{i_2}, \cdots, a_{i_l}, \cdots, a_{i_{k-1}}\} \bigcup \{0\}$.

We claim that $\hat{D}$ and $\hat{D}_{i_j}$ are correctly generated.

$$\begin{aligned}
\hat{D} &= B^{-\eta} h^r = g^{-b\eta} g^{r(a+\eta)} \\
&= g^{ab} g^{-ab-b\eta} g^{r(a+\eta)} \\
&= g^{ab} g^{-b(a+\eta)} g^{r(a+\eta)} \\
&= g^{ab} g^{(a+\eta)(r-b)} \\
&= g^{ab} h^{r-b}.
\end{aligned}$$

Let $\hat{r} = r - b$, we have $\hat{D} = g^{ab} h^{\hat{r}}$. By the choices of $y_{i_1}, y_{i_2}, \cdots, y_{i_{k-1}}$, we implicitly defined a $(k-1)$ degree polynomial $\hat{p}(x) \in \mathbb{Z}_p[x]$, where $\hat{p}(0) = r - b$ and $\hat{p}(a_{i_j}) = y_{i_j}$ for $a_{i_j} \in \hat{A} \bigcap A^*$. By Lagrange interpolation, we can reconstruct $\hat{p}(x)$ as follows:

$$\hat{p}(x) = (r - b)\Delta_{\Omega,0}(x) + \sum_{j=1}^{k} y_{i_j} \Delta_{\Omega,a_{i_j}}(x).$$

Setup. Suppose that the universal attribute set $\mathcal{U} = \{a_1, a_2, \cdots, a_n\} \in \mathbb{Z}_p^n$, where $n < p$. This algorithm takes as input a security parameter 1^ℓ, and outputs the bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and an one-way hash function $H : \{0,1\}^* \rightarrow \{0,1\}^\lambda$, where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ and λ is another security parameter ($\lambda < \ell$). Let g and h be the generators of $\mathbb{G}$. It selects $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $Y = e(g,g)^\alpha$. For each attribute $a_i \in \mathcal{U}$, it chooses $t_i \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $T_i = g^{t_i}$. The public parameters are

$$(e, p, \mathbb{G}, \mathbb{G}_\tau, g, h, Y, T_1, \cdots, T_n).$$

The master secret key is $(\alpha, t_1, t_2, \cdots, t_n)$.

KeyGen. To generate a secret key for the user U who holds a set of attribute $A_U \subseteq \mathcal{U}$, this algorithm chooses $\sigma_u \overset{R}{\leftarrow} \mathbb{Z}_p$ and a $(k-1)$ degree polynomial $p(x) \overset{R}{\leftarrow} \mathbb{Z}_p[x]$ with $p(0) = \sigma_u$ and computes

$$D_U = g^\alpha h^{\sigma_u},\ \{D_{U,i} = h^{\frac{p(a_i)}{t_i}}\}_{a_i \in A_U},$$

where $a_i \in A_U$ is one of the user's attributes and $t_i \in \{t_1, t_2, \cdots, t_n\}$ is the master secret key corresponding to the attributes a_i. The secret key for the user U is $SK_U = (D_U, \{D_{U,i}\}_{a_i \in A_U})$ where $D_{U,i}$ is the secret key corresponding to the attribute $a_i \in A_U$.

Receiver Policy. If the receiver only wants to receive messages for the sender who holds k-out-of-ρ attributes $\mathcal{R} = \{a_{j_1}, a_{j_2}, \cdots, a_{j_k}, \cdots, a_{j_\rho}\} \subseteq \mathcal{U}$, he chooses $w \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $K = e(g,g)^{\alpha w}$, $W = g^w$ and $\{E_{j_c} = T_{j_c}^w\}_{c=1}^{\rho}$), where $k \leq \rho \leq n$. The receiver keeps K as the authentication key, and publishes his access structure $\mathbb{A}_R = (W,\ \{a_{j_c},\ E_{j_c}\}_{c=1}^{\rho})$.

Encryption. To encrypt a message $M \in \mathbb{G}_\tau$ under a set of attributes A_C, this algorithm chooses $s \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and a set of attributes $\mathcal{Q} = \{a_{v_1}, a_{v_2}, \cdots, a_{v_k}\} \subseteq A_S \bigcap \mathcal{R}$, and computes

$$C_0 = M \cdot e(g,g)^{\alpha s},\ C_1 = g^s,\ \{C_x = T_x^s\}_{a_x \in \mathbb{A}_C},$$

$$F_S = e(D_S, W),\ \{F_{v_j} = e(D_{S,v_j}, E_{v_j})^{\Delta_{\mathcal{Q}, a_{v_j}}(0)}\}_{a_{v_j} \in \mathcal{Q}} \text{ and } F = \frac{F_S}{\prod_{a_{v_j} \in \mathcal{Q}} F_{v_j}}$$

where A_S and $(D_S, \{D_{S,v_j}\}_{a_{v_j} \in \mathcal{Q}})$ are the set of attributes held by the sender and his partial secret key, respectively. The ciphertext is $CT = (C_0,\ C_1,\ \{C_x\}_{a_x \in A_C})$ and the authentication information is $AI = \Gamma = H(F||CT)$.

Decryption.

1. Filtration. Receiving the ciphertext $CT = (C_0,\ C_1,\ \{C_x\}_{a_x \in A_C})$ and the authentication information $AI = \Gamma$, this algorithm checks $\Gamma \overset{?}{=} H(K||CT)$. If the equation holds, it goes to the next step. Otherwise, it aborts.
2. Decryption. This algorithm takes as input as the ciphertext $CT = (C_0,\ C_1,\ \{C_x\}_{a_x \in A_C})$ and the receiver's secret key $SK_R = (D_R, \{D_{R,i_j}\}_{a_{i_j} \in A_R})$, and outputs

$$M = C_0 \cdot \frac{\prod_{a_x \in A_C} e(D_{R,x}, C_x)^{\Delta_{A_C, a_x}(0)}}{e(D_R, C_1)}$$

FIGURE 2. Attribute-based Data Transfer with Filtering (ABDTF) Scheme

TABLE 1. The Comparison of Computation Cost

Scheme	Computation Cost				
	Setup	KeyGen	Receiver Policy	Encryption	Decryption
SW[7]	$(n+1)E + P$	$\|A_U\|E$	$--$	$(\|A_C\| + 1)E$	$kE + 2kP$
GPSW[28]	$(n+1)E + P$	$\|A_U\|E$	$--$	$(\|A_C\| + 1)E$	$kE + kP$
PTMW[31]	E	$3\|A_U\|E$	$--$	$(\|A_C + 2\|)E$	$2kE + (k+1)P$
AI[39]	$2E$	$(2 + \|A_U\| + \ell_O)E$	$--$	$(2 + 2\ell_S + A_C)E$	$(A_U + A_C)E +$ $(2A_U + 2A_C)P$
Our Scheme	$(n+2)E + P$	$(\|A_U\| + 1)E$	$(\rho + 2)E$	$(\|A_C\| + k + 2)E + (k+1)P + T_H$	$kE + (k+1)P + T_H$

TABLE 2. The Comparison of Communication Cost

Scheme	Communication Cost			
	Setup	KeyGen	Receiver Policy	Encryption
SW[7]	$nE_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$\|A_U\|E_{\mathbb{G}}$	$--$	$\|A_C\|E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$
GPSW[28]	$nE_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$\|A_U\|E_{\mathbb{G}}$	$--$	$\|A_C\|E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$
PTMW[31]	$(n+1)E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$2\|A_U\|E_{\mathbb{G}}$	$--$	$(\|A_C\|+1)E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$
AI[39]	$(n+\|A_C\|+1+\ell_{S,max})E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$(1+\|A_U\|+2\ell_O)E_{\mathbb{G}}$	$--$	$(1+\ell_S+\|A_C\|)E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$
Our Scheme	$nE_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$(\|A_U\|+1)E_{\mathbb{G}}$	$(\rho+1)E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$(\|A_C\|+1)E_{\mathbb{G}} + 2E_{\mathbb{G}_\tau}$

Therefore, for $a_{\lambda_i} \in \hat{A} - A_C$, we have

$$\begin{aligned}\hat{D}_{\lambda_i} &= h^{\frac{\hat{p}(a_{\lambda_i})}{t_{\lambda_i}(a+\eta)}} = g^{\frac{\hat{p}(a_{\lambda_i})}{t_{\lambda_i}}} \\ &= g^{\frac{(r-b)\Delta_{\Omega,0}(a_{\lambda_i})}{t_{\lambda_i}}} \prod_{j=1}^{k-1} g^{\frac{y_{i_j}\Delta_{\Omega,a_{i_j}}(a_{\lambda_i})}{t_{\lambda_i}}} \\ &= (B^{-1}g^r)^{\frac{\Delta_{\Omega,0}(a_{\lambda_i})}{t_{\lambda_i}}} \prod_{j=1}^{k-1} g^{\frac{y_{i_j}\Delta_{\Omega,a_{i_j}}(a_{\lambda_i})}{t_{\lambda_i}}}.\end{aligned}$$

Challenge. The adversary $\mathcal{A}$ submits two equal-length messages M_0 and M_1. $\mathcal{B}$ flips an unbiased coin with $\{0,1\}$, and obtains $\mu \in \{0,1\}$. $\mathcal{B}$ computes

$$C_0 = M_\mu \cdot Z,\ C_1 = C,\ \{C_x = C^{t_x}\}_{a_x \in A^*},$$

where a_x is one of the attributes in the challenged set A^*. $\mathcal{B}$ responds $\mathcal{A}$ with the challenged ciphertext $(C_0, C_1, \{C_x\}_{a_x \in A_C})$. Hence, when $Z = e(g,g)^{abc}$, $(C_0, C_1, \{C_x\}_{a_x \in A^*})$ is the valid ciphertext of M_μ.

Phase 2. Phase 1 is repeated.

Guess. The adversary $\mathcal{A}$ outputs his guess μ' on μ. If $\mu' = \mu$, $\mathcal{B}$ outputs his guess $\varrho' = 0$. If $\mu' \neq \mu$, $\mathcal{B}$ outputs his guess $\varrho' = 1$.

The public parameters and secret keys created in the simulation paradigm are identical to those in the real protocol. Hence, the advantage with which $\mathcal{B}$ can use $\mathcal{A}$ to break the DBDH assumption can be computed as follows.

If $\varrho = 0$, $(C_0, C_1, \{C_x\}_{a_x \in A_C})$ is a valid ciphertext of M_μ. Therefore, $\mathcal{A}$ can output $\mu' = \mu$ with advantage at least $\epsilon(\ell)$, namely $\Pr[\mu' = \mu | \varrho = 0] \geq \frac{1}{2} + \epsilon(\ell)$. Since $\mathcal{B}$ guesses $\varrho' = 0$ when $\mu' = \mu$, we have $\Pr[\varrho' = \varrho | \varrho = 0] \geq \frac{1}{2} + \epsilon(\ell)$.

In the case when $\varrho = 1$, $\mathcal{A}$ cannot obtain any information about μ. In other words, $\mathcal{A}$ can output $\mu' \neq \mu$ with no advantage, namely $\Pr[\mu' \neq \mu | \varrho = 1] = \frac{1}{2}$. Since $\mathcal{B}$ guesses $\varrho = 1$ when $\mu' \neq \mu$, we have $\Pr[\varrho' = \varrho | \varrho = 1] = \frac{1}{2}$.

Therefore, the over advantage which $\mathcal{B}$ can break the DBDH assumption is $|\frac{1}{2}\Pr[\varrho' = \varrho | \varrho = 0] + \frac{1}{2}\Pr[\varrho' = \varrho | \varrho = 1] - \frac{1}{2}| \geq \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \epsilon(\ell) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} \geq \frac{1}{2}\epsilon(\ell)$. □

THEOREM 3.2. *Our attribute-based data transfer with filtering (ABDTF) scheme is $(T, q, \epsilon(\ell))$ secure in the filtration security model if the $(T', \epsilon'(\ell))$ computational bilinear Diffie-Hellman (CBDH) assumption holds in $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ and the hash function H is collusion resistant, where*

$$T = \mathcal{O}(T) \quad \text{and} \quad \epsilon(\ell) = \epsilon'(\ell).$$

Proof. If there exists an adversary $\mathcal{A}$ who can $(T, q, \epsilon(\ell))$ break the filtration security of our scheme, we can construct an algorithm $\mathcal{B}$ that can uses $\mathcal{A}$ to break the CBDH assumption as follows.

The challenger generates the bilinear group $\mathcal{GG}(1^\ell) \to (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and a hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$. Let g be a generator of the group $\mathbb{G}$. The challenger sends $(A, B, C) = (g^a, g^b, g^c)$ to $\mathcal{B}$. $\mathcal{B}$ will outputs $Z = e(g,g)^{abc}$.

Initialization. The adversary $\mathcal{A}$ submits a set of challenged attributes R^*.

Setup. $\mathcal{B}$ sets $Y = e(g,g)^{ab}$ and $h = Ag^\eta$, where $\eta \overset{R}{\leftarrow} \mathbb{Z}_p$. If $a_i \in R^*$, he chooses $t_i \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $T_i = g^{t_i}$. Otherwise, he chooses $t_i \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $T_i = h^{t_i} = g^{t_i(a+\eta)}$. So, $\mathcal{B}$ implicitly defines the public parameters and the master secret key as

$$(e, p, \mathbb{G}, \mathbb{G}_\tau, Y, T_1, T_2, \cdots, T_n)$$

and

$$(ab, \{t_i\}_{a_i \in R^*}, \{t_i(a+\eta)\}_{a_i \notin R^*})$$

respectively. $\mathcal{B}$ sends $(e, p, \mathbb{G}, \mathbb{G}_\tau, Y, T_1, T_2, \cdots, T_n)$ to $\mathcal{A}$. These public parameters are the same as those in the above proof.

Phase 1. The adversary $\mathcal{A}$ queries the authentication information for an access structure $\mathbb{A}_R = (W, \{a_{j_c}, E_{j_c}\}_{c=1}^{\rho})$, where $|R^* \bigcap Q| < k$ and $Q = \{a_{j_1}, a_{j_2}, \cdots, a_{j_\rho}\}$. At first, by using the techniques in (1), (2) and (3), $\mathcal{B}$ generates the secrete key $(D_S, \{D_{S,v_j}\}_{a_{v_j} \in Q'})$, where $Q' \subseteq Q$ and $|Q'| = k$. $\mathcal{B}$ computes $F_S = e(D_S, W)$, $\{F_{v_j} = e(D_{S,v_j}, E_{v_j})\}_{a_{v_j} \in Q'}$ and $F = \frac{F_S}{\prod_{a_{v_j} \in Q'} F_{v_j}}$. Then, $\mathcal{B}$ chooses $s \overset{R}{\leftarrow} \mathbb{Z}_p$, $M \overset{R}{\leftarrow} \mathbb{G}_\tau$ and a set of attributes $L = \{a_{j_1}, a_{j_2}, \cdots, a_{j_k}\} \subseteq \mathcal{U}$, and computes $CT = (C_0, C_1, \{C_{j_x}\}_{x=1}^k)$, where $C_0 = e(g,g)^{\alpha s} \cdot M$, $C_1 = g^s$ and $\{C_{j_x} = T_{j_x}^s\}_{a_{j_x} \in L}$. $\mathcal{B}$ computes $\Gamma = H(F||CT)$ and sends the authentication information $AI = \Gamma$ and the ciphertext CT to $\mathcal{A}$.

Challenge. $\mathcal{B}$ sets $W^* = C$ and computes $\{E_{j_c}^* = C^{t_{j_c}}\}_{a_{j_c} \in R^*}$. $\mathcal{B}$ sends the challenged access structure $\mathbb{A}_{R^*} = (W^*, R^*, \{E_{j_c}^*\}_{a_{j_c} \in R^*})$ to $\mathcal{A}$.

Phase 2. Phase 1 is repeated.

Outputs. At the end, the adversary $\mathcal{A}$ outputs the authentication information $AI = \Gamma^*$.

As shown in the simulation, the public parameters and the secret keys are identical to those in the real protocol. Now, we compute the advantage with which $\mathcal{B}$ can use $\mathcal{A}$ to break the CBDH assumption.

When $W = C = g^c$, it implies $K^* = e(g,g)^{abc}$. If $\mathcal{A}$ can output a valid authentication information $AI = \Gamma^*$ for the access structure $\mathbb{A}_{R^*}$ with advantage at least $\epsilon(\ell)$, he can computes $F^* = e(g,g)^{abc}$ with the same advantage as H is a one-way hash function. Therefore, $\mathcal{B}$ can compute $K^* = F^* = Z = e(g,g)^{abc}$ with the same advantage.

□

3.4. Fine-Grained Access Control

In our ABDTF scheme, we use the (k,n)-threshold access structure. In order to express a complex access structure, we can use the *access tree* introduced by Goyal, Pandey, Sahai and Waters [28]. Let $\mathcal{T}$ denote a tree which specifies an access structure and defines an ordering among the children of the node v from 1 to n_v. Each no-leaf node v in $\mathcal{T}$ defines a threshold gate which comprises a threshold value k_v and the number of its children n_v, where $k_v \leq n_v$. When $k_v = 1$, it defines an OR gate; while $k_v = n_v$, it defines an AND gate. Each leaf-node v in $\mathcal{T}$ is related to an attribute and a threshold value $k_v = 1$. For any access structure, a polynomial can be constructed for each non-leaf node in $\mathcal{T}$ following the top-to-down manner. In our system, the central authority selects an access tree $\mathcal{T}$ for a complex access structure. Starting from the root node r, it sets the degree d_r of the polynomial as $k_r - 1$. To generate secret keys for a user with a set of attributes, he chooses a $k_r - 1$ degree polynomial $p_r(x)$ with $p_r(0) = \sigma_u$. For other nodes in $\mathcal{T}$, it can set $p_v(0) = p_{parent(v)}(index(v))$, where $parent(x)$ denotes the parent of node v and $index(v)$ denotes the ordering number labeled with v. This technique can also used by the receiver to specify a complex access structure.

4. PERFORMANCE EVALUATION

The efficiency of a pairing-based scheme depends on the employed elliptic curve. Literatures [41, 47, 48] suggested the selection of elliptic curves for efficient cryptosystems. In order to select a secure elliptic curve, two important factors must be considered: the group size l of the elliptic curve and the embedding degree d. To achieve the security of 1,024-bit RSA, the group size and the embedding degree should satisfy $l \times d \geq 1024$. Although the elliptic curve with high embedding degree can result in a short size of elements, the pairing operations on this curve is expensive. Most of pairing-based schemes are implemented in the elliptic curves: Type A and Type D [41]. Type A is supersingular curve $y^2 = x^3 + x$ and the group order is a Solinas prime. On a Type A curve where $\mathbb{G}_1 = \mathbb{G}_2$, the pairing operation is fastest. Meanwhile, a Type D curve is an MNT curve $y^2 = x^3 + \lambda_1 x + \lambda_0$, where $\mathbb{G}_1 \neq \mathbb{G}_2$. On a Type D curve, the element can be shorter but the pairing operation is more expensive.
We implement our scheme on Type A curve: $y^2 = x^3 + x$, where $l = 512$, $d = 2$ and p is a 160-bit prime number. We use $SHA-1$ as the one-way hash function.

4.1. Benchmark Time

We test the running time of different operations on the bilinear group on a DELL E630 with Intel(R) Core™ 2 Duo CPU (T8100@2.10GHz) and 2GB RAM running Ubuntu 10.10. The running time is obtained by computing the average of running the operation 10 times with random inputs using the text code from the PBC library [41]. The running times of different operations on the bilinear group from Type A curve are described in Table 3. By PP_p, PP_1, PP_2 and PP_τ, we denote the running time of a pairing operation with preprocessing, an exponential operation on group $\mathbb{G}_1$ with preprocessing, an exponential operation on group $\mathbb{G}_2$ with preprocessing and an exponential operation on group $\mathbb{G}_\tau$ with preprocessing, respectively. By multi-based, we denote the running time of executing exponential operations on more than one base, such as $\eta = g^{a_1} h^{a_2}$.
The running time and communication cost in different phases of our scheme are described in Table 4 and Table 5, respectively.

4.2. Implementations of Our Scheme

In our implementation, we set the number of the universal attributes as $|\mathcal{U}| = 32$, the threshold value as $k = 8$, the number of the attributes used to encrypt messages as $|A_C| = 8$ and $\rho = 16$, namely the central authority specifies a $(8, 32)$-threshold access structure and the receiver specifies a $(8, 16)$-threshold access structure. The running time and the communication cost in different phases are described in Fig. 3, Fig. 4, Fig. 5 and Fig. 6. We separate the running time and the communication cost of KeyGen from Fig. 4 and Fig. 6 as it depends on the number of the attributes held by each user.

We observe that it takes 14.304 ms to setup the scheme. It consumes 6.945 ms to generate an access structure by the receiver. To encrypt a message and decrypt a ciphertext, it takes 12.643 ms and 8.938 ms, respectively. The time consumed by KeyGen is linear in the number of the attributes held by each user.

Type A curve has symmetric bilinear groups and both the elements in groups $\mathbb{G}$ and $\mathbb{G}_\tau$ can be represented with 512 bits. Therefore, the lengths of the authentication key and the ciphertexts are 64 bytes and 660 bytes, respectively.

TABLE 3. Benchmark Time of Different Operations on a Type A Curve and SHA-1 (ms)

Curve	Pairing		$\mathbb{G}_1$		$\mathbb{G}_2$		$\mathbb{G}_\tau$		SHA-1
	Normal	PP_p	PP_1	Multi-based	PP_2	Multi-based	PP_τ	Multi-based	
Type A	2.001	0.939	0.405	3.847	0.407	3.933	0.060	0.448	0.007

TABLE 4. Running Time of Our Scheme on a Type A Curve (ms)

Curve	Setup	KeyGen	Receiver Policy	Encryption	Decryption
Type A	$1.344 + 0.405 \times n$	$(\|A_U\| + 1) \times 0.405$	$0.465 + \rho \times 0.405$	$1.411 + \|A_C\| \times 0.405 + k \times 0.999$	$0.946 + k \times 0.999$

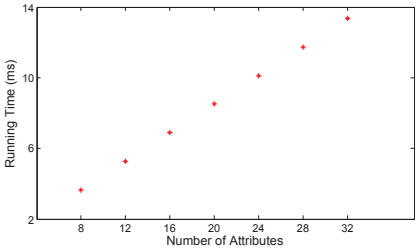

FIGURE 3. The running Time of KeyGen

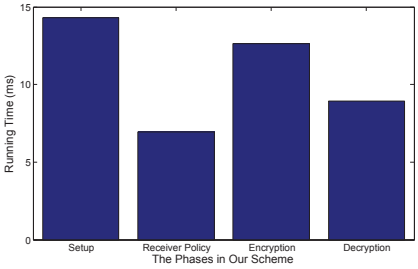

FIGURE 4. The Running Time of Setup, Receiver Policy, Encryption and Decryption

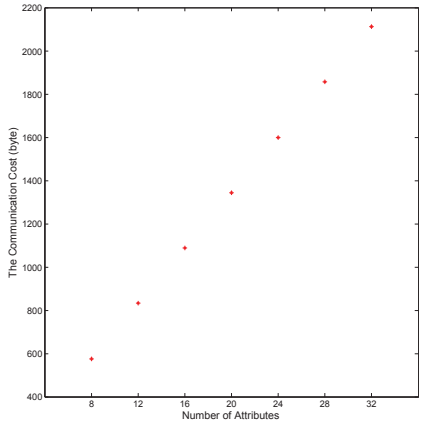

FIGURE 5. The Communication Cost of KeyGen

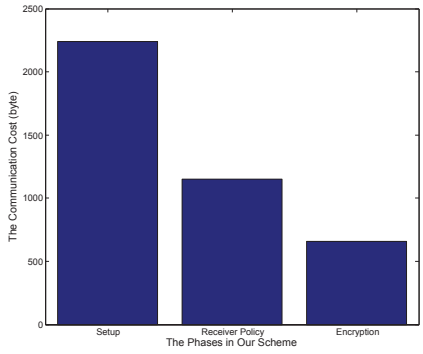

FIGURE 6. The Communication Cost of Setup, Receiver Policy, Encryption and Decryption

5. CONCLUSION

The confidentiality of the sensitive data and the DoS attacks attract lots of interests in data transfer research community. Even though data transfer system (DTS) have been extensively studied recently, there is no scheme that discusses how to transfer and filter data according the required attributes. In this paper, we proposed the formal definition and security models for attribute-based data transfer with filtering (ABDTF) scheme, which provides a formal treatment for the research of ABDTF schemes. Subsequently, we designed an ABDTF scheme and proved its security in the proposed security models. In the proposed scheme, both the authentication key and the authentication information are short. Note that the authentication key can be updated offline without re-initializing the system and re-issuing secret key to users. To update an authentication key, the receiver selects a random number and computes a new authentication key and access structure by using the selected random number and the public parameters without any help from the CA. And also, the authentication information can be computed by the sender off-line. Furthermore, we implemented any access structure using the access tree technique. Finally, we implement our scheme in the PBC library.

ACKNOWLEDGEMENT

The first author was supported by PhD scholarships of Smart Services Cooperative Research Centre (CRC) and University of Wollongong. The second author was supported by ARC Future Fellowship FT0991397. This work is supported by ARC Discovery Project DP130101383.

TABLE 5. The Communication Cost of Our Scheme on Type A Curve (bits)

Curve	Setup	KeyGen	Receiver Policy	Encryption
Type A	$(n+3)\times 512$	$(\|A_U\|+1)\times 512$	$(\rho+2)\times 512$	$(\|A_C\|+2)\times 512+160$

REFERENCES

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., id Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010) A view of cloud computing. *Communications of the ACM*, **53**, 50–58.

[2] Yu, S., Wang, C., Ren, K., and Lou, W. (2010) Achieving secure, scalable, and fine-grained data access control in cloud computing. *Proc. IEEE INFOCOM 2010*, San Diego, CA, USA, March 15-19, pp. 534–542. IEEE, Washington, DC, USA.

[3] Zhu, S., Setia, S., Jajodia, S., and Ning, P. (2004) An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. *Proc. S & P 2004*, Berkeley, CA, USA, May 9-12, pp. 259–271. IEEE, Washington, DC, USA.

[4] Lu, R., Lin, X., Zhu, H., Liang, X., and Shen, X. S. (2010) BECAN: a bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* , **23**, 32–43.

[5] Shamir, A. (1985) Identity-based cryptosystems and signature schemes. *Proc. CRYPTO 1984*, Santa Barbara, CA, USA, August 19-22, Lecture Notes in Computer Science, **196**, pp. 47–53. Springer, Berlin.

[6] Boneh, D. and Franklin, M. K. (2001) Identity-based encryption from the Weil pairing. *Proc. CRYPTO 2001*, Santa Barbara, CA, USA, August 19-23, Lecture Notes in Computer Science **2139**, pp. 213–229. Springer, Berlin.

[7] Sahai, A. and Waters, B. (2005) Fuzzy identity-based encryption. *Proc. EUROCRYPT 2005*, Aarhus, Denmark, May 22-26, Lecture Notes in Computer Science **3494**, pp. 457–473. Springer, Berlin.

[8] Waters, B. (2011) Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. *Proc. PKC 2011*, Taormina, Italy, March 6-9, Lecture Notes in Computer Science **6571**, pp. 53–70. Springer, Berlin.

[9] Moore, D., Voelker, G. M., and Savage, S. (2001) Inferring internet denial-of-service activity. *Proc. Usenix Security Symposium 2001*, Washington, USA, August 13-17, pp. 1-14. USENIX, Berkeley, CA, USA.

[10] Loukas, G. and Ö̈ke, G. (2010) Protection against denial of service attacks: A survey. *The Computer Journal*, **53**, 1020–1037.

[11] Sharafat, A. R. and Fallah, M. S. (2004) A framework for the analysis of denial of service attacks. *The Computer Journal*, **47**, 179–192.

[12] Thomas, M. and Dhillon, G. (2012) Interpreting deep structures of information systems security. *The Computer Journal*, **55**, 1148–1156.

[13] Liu, H. (2010) A new form of DoS attack in a cloud and its avoidance mechanism. *Proc. CCSW 2010*, Chicago, IL, USA, October 8, pp. 65–76. ACM, NewYork, NY, USA.

[14] Jansen, W. A. (2011) Cloud hooks: Security and privacy issues in cloud computing. *Proc. HICSS 2011*, Koloa, Kauai, HI, USA, January 4-7, pp. 1–10. IEEE, Washington, DC, USA.

[15] Bloom, B. H. (1970) Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, **13**, 422–426.

[16] Mitzenmacher, M. (2002) Compressed Bloom filters. *IEEE/ACM Transactions on Networking*, **10**, 604–612.

[17] Little, W. D. (1974) An algorithm for high-speed digital filters. *IEEE Transactions on Computers*, **C-23**, 466–469.

[18] Hua, Y., Xiao, B., and Wang, J. (2009) BR-Tree: a scalable prototype for supporting multiple queries of multidimensional data. *IEEE Transactions on Computers*, **58**, 1585–1598.

[19] Yuen, C. K. (1977) On Little's digital filtering algorithm. *IEEE Transactions on Computers*, **C-26**, 309–309.

[20] Ye, F., Luo, H., Lu, S., and Zhang, L. (2004) Statistical en-route filtering of injected false data in sensor networks. *Proc. INFOCOM 2004*, HongKong, March 7-11, pp. 2446–2457. IEEE, Washington, DC, USA.

[21] Yang, H., Ye, F., Yuan, Y., Lu, S., and Arbaugh, W. (2005) Toward resilient security in wireless sensor networks. *Proc. MOBIHOC 2005*, Urbana-Champaign, Illinois, USA, May 25-27, pp. 34–45. ACM, NewYork, NY, USA.

[22] Ren, K., Lou, W., and Zhang, Y. (2006) Providing location-aware end-to-end data security in wireless sensor networks. *Proc. INFOCOM 2006*, Barcelona, Spain, April 23-29, pp. 1–12. IEEE, Washington, DC, USA.

[23] Zhang, Y., Liu, W., Lou, W., and Fang, Y. (2006) Location-based compromise-tolerant security mechanisms for wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, **24**, 247–260.

[24] Yu, Z. and Guan, Y. (2010) A dynamic en-route filtering scheme for data reporting in wireless sensor networks. *IEEE/ACM Transactions on Networking*, **18**, 150–163.

[25] Waters, B. (2005) Efficient identity-based encryption without random oracles. *Proc. EUROCRYPT 2005*, Aarhus, Denmark, May 22C26, Lecture Notes in Computer Science **3494**, pp. 114–127. Springer, Berlin.

[26] Gentry, C. (2006) Practical identity-based encryption without random oracles. *Proc. EUROCRYPT 2006*, Petersburg, Russia, May 28-June 1, Lecture Notes in Computer Science **4004**, pp. 445–464. Springer, Berlin.

[27] Boneh, D. and Boyen, X. (2004) Efficient selective-ID secure identity-based encryption without random oracles. *Proc. EUROCRYP 2004*, Interlaken, Switzerland, May 2-6, Lecture Notes in Computer Science **3027**, pp. 223–238. Springer, Berlin.

[28] Goyal, V., Pandey, O., Sahai, A., and Waters, B. (2006) Attribute-based encryption for fine-grained access control of encrypted data. *Proc. CCS 2006*, Alexandria, VA, USA, October 30 - November 3, pp. 89–98. ACM, NewYork, NY, USA.

[29] Chase, M. (2007) Multi-authority attribute based encryption. *Proc. TCC 2007*, Amsterdam, The Netherlands, February 21-24, Lecture Notes in Computer Science **4392**, pp. 515–534. Springer, Berlin.

[30] Chase, M. and Chow, S. S. (2009) Improving privacy and security in multi-authority attribute-based encryption. *Proc. CCS 2009*, Chicago, Illinois, USA, November 9-13, pp. 121–130. ACM, NewYork, NY, USA.

[31] Pirretti, M., Traynor, P., McDaniel, P., and Waters, B. (2006) Secure attributebased systems. *Proc. CCS 2006*, Alexandria, VA, USA, October 30-November 3, pp. 99–112. ACM, NewYork, NY, USA.

[32] Han, J., Susilo, W., Mu, Y., and Yan, J. (2012) Privacy-preserving decentralized key-policy attribute-based encryption. *IEEE Transactions onParallel and Distributed Systems*, **23**, 2150–2162.

[33] Han, J., Susilo, W., Mu, Y., and Yan, J. (2012) Attribute-based oblivious access control. *The Computer Journal*, **55**, 1202–1215.

[34] Bethencourt, J., Sahai, A., and Waters, B. (2007) Ciphertext-policy attribute-based encryption. *Proc. S & P 2007*, Oakland, California, USA, May 20-23, pp. 321–34. IEEE, Washington, DC, USA.

[35] Cheung, L. and Newport, C. (2007) Provably secure ciphertext policy ABE. *Proc. CCS 2007*, Alexandria, Virginia, USA, October 28-31, pp. 456–465. ACM, NewYork, NY, USA.

[36] Herranz, J., Laguillaumie, F., and Ráfols, C. (2010) Constant size ciphertexts in threshold attribute-based encryption. *proc. PKC 2010*, Paris, France, May 26-28, Lecture Notes in Computer Science **6056**, pp. 19–34. Springer, Berlin.

[37] Lewko, A., Okamoto, T., Sahai, A., Takashima, K., and Waters, B. (2010) Fully secure functional encryption: Attribute- based encryption and (hierarchical) inner product encryption. *Proc. EUROCRYPT 2010*, Riviera, French, May 30-June 3, Lecture Notes in Computer Science **6110**, pp. 62–91. Springer, Berlin.

[38] Beimel, A. (1996) Secure Schemes for Secret Sharing and Key Distribution. PhD thesis Israel Institute of Technology Technion, Haifa, Israel.

[39] Attrapadung, N. and Imai, H. (2009) Dual-policy attribute based encryption. *Proc. ACNS 2009*, Paris-Rocquencourt, France, June 2-5, Lecture Notes in Computer Science **5536**, pp. 168–185. Springer, Berlin.

[40] Yu, S., Ren, K., and Lou, W. (2011) FDAC: Toward fine-grained data access control in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, **22**, 673–686.

[41] Lynn, B. (2006). Pbc library: The pairing-based cryptography library. http://crypto.stanford.edu/pbc/.

[42] Deng, R. H., Gong, L., and Lazar, A. A. (1995) Securing data transfer in asynchronous transfer mode networks. *Proc. GlobeCom 1995*, SINGAPORE, November 14-16, pp. 1198 – 1202. IEEE, Washington, DC, USA.

[43] Pearson, S., Shen, Y., and Mowbray, M. (2009) A privacy manager for cloud computing. *Proc. CloudCom 2009*, Beijing, China, December 1-4, Lecture Notes in Computer Science **5931**, pp. 90–106. Springer, Berlin.

[44] Mont, M. C., Harrison, K., and Sadler, M. (2003) The HP time vault service: Exploiting IBE for timed release of confidential information. *Proc. WWW 2003*, Budapest, Hungary, May 20-24, pp. 160–169. ACM, NewYork, NY, USA.

[45] Jaeger, T. and Prakash, A. (1994) Support for the file system security requirements of computational e-mail systems. *Proc. CCS 1994*, Fairfax Va., USA, November 2-4, pp. 1–9. ACM, NewYork, NY, USA.

[46] Khurana, H., Slagell, A., and Bonilla, R. (2005) SELS: a secure e-mail list service. *Proc. SAC 2005*, Santa Fe, New Mexico, USA, March 13-17, pp. 306–313. ACM, NewYork, NY, USA.

[47] National Institute of Standards and Technology (July, 1999) *Recomeeded Elliptic Curves for Federal Government Use.* http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf.

[48] Certicom Research (September, 2000) *Standards for Efficient Cryptography SEC 2: Recommended Elliptic Curve Domain Parameters.* http://www.secg.org/collateral/sec2-final.pdf.