**INTRODUCTION**

IN the recent decades, cloud-based storage service has attracted considerable attention from both academia and industries. It may be widely used in many Internet-based commercial applications (e.g., Apple iCould) due to its long-list benefits including access flexibility and free of local data management. Increasing number of individuals and companies nowadays prefer to outsource their data to remote cloud in such a way that they may reduce the cost of upgrading their local data management facilities/devices. However, the worry of security breach over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage service. In many practical applications, outsourced data may need to be further shared with others. For example, a Dropbox user Alice may share photos with her friends. Without using data encryption, prior to sharing the photos, Alice needs to generate a sharing link and further share the link with friends. Although guaranteeing some level of access control over unauthorized users (e.g., those are not Alice’s friends), the sharing link may be visible within the Dropbox administration level (e.g., administrator could reach the link).

Since the cloud (which is deployed in an open network) is not be fully trusted, it is generally recommended to encrypt the data prior to being uploaded to the cloud to ensure data security and privacy. One of the corresponding solutions is to directly employ an encryption technique (e.g., AES) on the outsourced data before uploading to cloud, so that only specified cloud user (with valid decryption key) can gain access to the data via valid decryption.

To prevent shared photos being accessed by the “insiders” of the system, a straightforward way is to designate the group of authorized data users prior to encrypting the data. In some cases, nonetheless, Alice may have no idea about who the photo receivers/users are going to be. It is possible that Alice only has knowledge of attributes w.r.t. photo receivers. In this case, traditional public key encryption (e.g., Paillier Encryption), which requires the encryptor to know who the data receiver is in advance, cannot be leveraged. Providing policy-based encryption mechanism over the outsourced photos is therefore desirable, so that Alice makes use of the mechanism to define access policy over the encrypted photos to guarantee only a group of authorized users is able to access the photos.

In a cloud-based storage service, there exists a common attack that is well-known as resource-exhaustion attack. Since a (public) cloud may not have any control over download request (namely, a service user may send unlimited numbers of download request to cloud server), a malicious service user may launch the denial-of-service (DoS)/distributed denial-of-service (DDoS) attacks to consume the resource of cloud storage service server so that the cloud service could not be able to respond honest users’ service requests. As a result, in the “pay-as-you-go” model, economic aspects could be disrupted due to higher resource usage. The costs of cloud service users will rise dramatically as the attacks scale up. This has been known as Economic Denial of Sustainability (EDoS) attack [32], [33], which targets to the cloud adopter’s economic resources. Apart from economic loss, unlimited download itself could open a window for network attackers to observe the encrypted download data that may lead to some potential information leakage (e.g., file size). Therefore, an effective control over download request for outsourced (encrypted) data is also needed.

In this paper, we propose a new mechanism, dubbed dual access control, to tackle the above aforementioned two problems. To secure data in cloud-based storage service, attribute-based encryption (ABE) [9] is one of the promising candidates that enables the confidentiality of outsourced data as well as fine-grained control over the outsourced data.

In particular, Ciphertext-Policy ABE (CP-ABE) [5] provides an effective way of data encryption such that access policies, defining the access privilege of potential data receivers, can be specified over encrypted data. Note that we consider the use of CP-ABE in our mechanism in this paper. Nevertheless, simply employing CP-ABE technique is not sufficient to design an elegant mechanism guaranteeing the control of both data access and download request.

A strawman solution to the control of download request is to leverage dummy ciphertexts to verify data receiver’s decryption rights. It, concretely, requires data owner, say Alice, to upload multiple “testing” ciphertexts along with the “real” encryption of data to cloud, where the “testing” ciphertexts are the encryptions of dummy messages under the same access policy as that of the “real” data. After receiving a download request from a user, say Bob, cloud asks Bob to randomly decrypt one of the “testing” ciphertexts. If a correct result/decryption is returned (i.e. indicating Bob is with valid decryption rights), Bob is authorized by Alice to access the ”real” data, so that the cloud allows Bob to download the corresponding ciphertext.

Nevertheless, several disadvantages of the above approach may be identified as follows. First of all, the data owner, Alice, is required to encrypt a number of dummy ciphertexts under the same policy as the “real” ciphertext. This may yield a considerable computational overhead for Alice, which may bring inconvenience in practice, for example, Alice just wants to upload one photo to iCloud from her cellphone, but needs to prepare more than one ciphertexts. Second, all ciphertexts, including dummy ones, are uploaded to cloud at the same time. This inevitably imposes extra cost on network bandwidth (as well as prolonging data uploading time), which may not be applicable to some service users whose cellular network is under pay-as-you-go plan or equipped with old generation of broadband cellular network technology (e.g., 3G). Third, a data receiver/user, Bob, has to additionally decrypt a random-chosen “testing” ciphertext from cloud, as a test of his valid download request. As a result, Bob has to “pay” double (decryption price) for accessing to the “real” data, which again may not be scalable in resource constrained setting.