A Generic Scheme for Secure Data Sharing in Cloud

Yanjiang Yang
Institute for Infocomm Research
Singapore
yyang@i2r.a-star.edu.sg

Youcheng Zhang
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics, China
zyc1111@21cn.com

***Abstract*—Working in various service models ranging from *S*aaS, *P*aaS, to *I*aaS, cloud computing is a new revolution in IT, and could reshape the business model of how the IT industry works today. Storage services are a fundamental component of the cloud computing paradigm. By exploiting the storage services, users outsource their data to the cloud so as to enjoy the reduced upfront maintenance and capital costs. However, a security challenge associated with data outsourcing is how to prevent data abuses by the cloud. It has been commonly accepted that data encryption offers a good solution to this problem. With data encryption, an issue arises when the data owner who outsourced the data wants to revoke some data consumers' access privileges, which normally involves key redistribution and data re-encryption. In this work, we propose a generic scheme to enable fine-grained data sharing over the cloud, which does not require key-redistribution and data re-encryption whatsoever. The main primitives we make use of are attribute-based/predicate encryption and proxy re-encryption, but our construction is not restricted to any specific scheme of its kind. Our scheme has a number of advantages over other similar proposals in the literature.**

***Keywords*-cloud computing; fine grained access control; attribute-based encryption; proxy re-encryption; user revocation**

I. Introduction

As one of the most hyped technology, cloud computing promises efficient and cost-effective computing and information use. Cloud computing mainly operates in three types of service models: software-as-a-service (*S*aaS), where customers rent and run software on the service provider's cloud; platform-as-a-service (*P*aaS), where customers develop applications within the development environment provided by the cloud service provider and offer the services through the provider's cloud; Infrastructure-as-a-service (*I*aaS), where the cloud provider supplies to customers the entire infrastructure to run their businesses.

From the customers' perspective, clouds could be *private* or *public*. Private clouds can be understood to be an emulation of cloud computing on private networks, and as such, customers have control over the cloud infrastructure. In contrast, A public cloud is hosted, operated, and managed by a cloud service provider, and the infrastructure thus goes beyond the control of the customers. While the customers may feel uneasy with less control and oversight over the infrastructure, public clouds indeed benefit customers with lower upfront capital cost and less hands-on management, which are among the main advertised advantages of cloud computing.

Storage services over public cloud, e.g., Microsoft's Azure storage and Amazon's S3, are a fundamental component of cloud computing, which allow the customers to outsource their data to the regime of a cloud. Data outsourcing as such relieves the customers from building and maintaining their proprietary databases, which usually is extremely costly. However, as the data are managed out of the governance of their owners, it is natural for the data owners to worry that their data would be abused without their consent or even awareness. It is now commonly accepted that encryption of the data in outsourcing is a good way to mitigate such concerns of data abuses by the cloud [11]. Indeed, with the possession of the secret key for decryption, data owners are entitled to still get control of their data, although the data physically resides beyond their territory. Better yet, data encryption also helps to solve other issues such as regulatory compliance, and geographic restrictions [21], [26], in the sense that the encrypted data are by no means useful without the decryption capability.

However, data encryption turns out to be a double-edged sword: with the data being in an encrypted form, it however becomes cumbersome to share the data with other users, as this usually requires the data owner to share the decryption key with those users who are deemed to have the access right to the data. A further issue arising from key sharing is *user revocation*, where some users should be deprived of their access rights in certain circumstances, e.g., they resign from their duty. The usual solution to user revocation requires the data owner to invalidate the existing key by re-encrypting the whole set of data with a new key, and in turn re-distributing the new key to the authorized users. This clearly is an enormously involved procedure to the data owner, especially when the data in outsourcing are large in quantity, and the data owner does not locally keep a copy of the outsourced data when using cloud storage.

Recently, Yu *et al.* proposed a solution to this problem by delegating the heavy overhead due to user revocation (e.g., data re-encryption) that would be otherwise imposed upon the data owner to the cloud who has abundant resources [31]: by leveraging on the attribute-based encryption scheme

1530-2016/11 $26.00 © 2011 IEEE
DOI 10.1109/ICPPW.2011.51

in [18] combined with the technique of proxy re-encryption in [4], the cloud is enabled to take the full load of key re-distribution and re-encryption of the requested data by users. The use of attribute-based encryption actually aims at achieving fine-grained access control. Indeed, in reality data sharing is very much likely to be enforced in a fine-grained fashion in terms of which user has the access right to what data. Attribute-based encryption by nature is an idea tool to realize such complex access control policies: a data record or a data file is associated with a set of attributes, and a user's access privileges are specified by a logical expression over these attributes; as such, each user's decryption key is issued in accordance with the logical expression defined by her access privileges. Thus, not surprisingly, there are also several other similar proposals advocating the use of attribute-based encryption to attain fine-grained access control over cloud storage, e.g., [30], [32].

While the aforementioned existing fine-grained access control proposals (represented by Yu *et al.*'s) manage to push the expensive task of data re-encryption (more precisely, it involves transformation of the attribute-based encryption) and key re-distribution to the cloud which is assumed to be much powerful than the data owner, it inevitably incurs heavy burden to the cloud. On the other hand, cloud computing represents a server-client operation model, where the cloud, as a single point of service, is expected to serve a large number of users; it is thus still important that users should impose as minimal overhead as possible to the cloud. Moreover, in certain charge mode, the cloud service provider may charge a data owner based on the amount of computation she imposes. In such a case, the lower computation overhead, the lower financial cost to the data owner.

Another concern with the existing fine-grained access control proposals, e.g., Yu *et al.*'s scheme, is that the cloud is *stateful*, in the sense that it has to retain the history of user revocation. This information would keep growing over time as the system proceeds. As such, an interesting question arises: is it possible to avoid stateful cloud, and key re-distribution and heavy data re-encryption in case of user revocation, while still enjoys the benefit of fine-grained access control? In this work, we provide an affirmative answer to this question.

Our Contribution. In particular, we present a generic scheme to achieve fine-grained data sharing/access control over the cloud. Like in [31], attribute-based encryption[1] and proxy re-encryption are the main tools that underpin our construction, where the former is used for fine-grained access control and the latter is for user revocation. Specifically, our scheme possesses the following features.

- *Generic construction*. Our scheme is a generic construction, not depending on any specific attribute-based encryption schemes and proxy re-encryption schemes. This allows for a tailored choice of the primitives (in terms of efficiency, fine-grainedness, and security level) when instantiated, catering to different application requirements. In contrast, Yu *et al.*'s scheme [31] is a concrete construction, relying on the combination of the Key Policy Attribute-Based Encryption in [18] and the technique of the bidirectional proxy re-encryption scheme in [4].
- *Efficient user revocation*. Our scheme admits very efficient user revocation: revocation of a user does not affect other non-revoked users at all, requiring no key update/re-distribution to non-revoked users. All that the cloud needs to do is simply destroying the corresponding re-encryption key of the proxy re-encryption, under the command of the data owner. This shows a sharp contrast to the existing fine-grained access control proposes which require key update/re-distribution to the non-revoked users.
- *Stateless cloud*. The cloud in our scheme is stateless, as it is not obliged to keep any information related to user revocation.
- *Direct security guarantee*. Our scheme makes direct use of the underlying cryptographic primitives (e.g., attribute-based encryption, proxy re-encryption), without imposing any change to them. The security properties of the respective primitives are thus not affected at all. Consequently, the security of our construction directly derives from the underlying primitives.

Organization. The rest of the paper is organized as follows. In Section II, we review the related work, respectively, on the cryptographic primitives we use, and on access contro/data sharing in cloud computing. In Section III, we formulate the system model and security requirements for data sharing in cloud computing. Our proposed scheme, together the security and performance analysis as well as a set of discussions, is presented in Section IV. Finally, Section V concludes the work.

II. Related Work

In this section, we review, respectively, the related work on attribute-based encryption (or predicate encryption) and proxy re-encryption, two cryptographic primitives underlying our construction, as well as cryptographic access control mechanisms for cloud computing.

A. Attribute-based/predicate Encryption

Attribute-based encryption (ABE) is a kind of fine-grained public key encryption: in traditional public key encryption, an encryptor encrypts a message under a public key, and the encryptor is assured that only the holder of the corresponding private key can decrypt; in comparison, in ABE, the

[1] As a matter of fact, any encryption mechanism that implements fine-grained access control, e.g., predicate encryption, can be used in our scheme.

encryptor can associate some complex access control policy with an encryption, and only those satisfying the policy can decrypt the ciphertext. The idea first conveyed in [29] was later evolved into attribute-based encryption. Two categories of attribute-based encryption are distinguished: Key-Policy Attributed-Based Encryption (KP-ABE) [18] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [8]. For the former, a ciphertext is associated with a set of attributes while a private key is issued as per a certain access control policy; the latter is the other way around such that a ciphertext is generated according to some access control policies while private keys are issued in association with attributes. The attribute-based encryption schemes in [8], [18] support attributes that are logically organized as a single set. Ciphertext-Policy Attribute-Set-based Encryption (CP-BSBE) proposed in [7] extends this aspect by supporting attributes organized in a recursive set structure. To date, Attribute-based encryption has evolved into *predicate encryption* (e.g., [22], [23], [27], [28]), which targets at handling more flexible and sophisticated access control policies, e.g., ranges, disjunctions, and so on.

Our construction is neutral to any particular attribute-based/predicate encryption schemes (or other encryption primitives that implement fine-grained access control), and it enjoys the freedom of instantiation choices, which are determined by the requirements of the underlying application, e.g., the fine-grainedness of access control, the level of security (e.g., CPA-security or CCA-security), and so on.

B. Proxy Re-encryption

The notion of proxy re-encryption (PRE) was initially introduced by Blaze, Bleumer and Strauss in [4]: proxy re-encryption (PRE) allows a semi-trusted proxy to convert a ciphertext originally under Alice's public key into one encrypting the same plaintext under Bob's public key[2]. The proxy only needs a re-encryption key given by Alice, and cannot learn anything about the underlying plaintext throughout the conversion process. Since then, a number of proxy re-encryption schemes have been proposed in the literature.

In 2005, Ateniese *et al.* [1], [2] presented two PRE schemes based on bilinear pairings, both of them are secure against chosen-plaintext attack (CPA). Later, Canetti and Hohenberger [9], and Libert and Vergnaud [25] proposed CCA-secure PRE schemes using bilinear pairings, respectively. Deng *et al.* [15] then proposed a CCA-secure bidirectional PRE scheme without pairings.

Proxy re-encryption has also been studied in identity-based scenarios. Based on the ElGamal-type public key encryption system [16] and Boneh-Boyen's identity-based encryption system [3], Boneh, Goh and Matsuo [6] described a hybrid proxy re-encryption system. Based on Boneh and Franklin's identity-based encryption system [5], Green and Ateniese [17] presented CPA and CCA-secure identity-based proxy re-encryption (IB-PRE) schemes in the random oracle model. Chu and Tzeng [13] presented the constructions of CPA and CCA-secure IB-PRE schemes without random oracle.

Again, our construction is not restricted to any particular proxy re-encryption scheme, and the choice of instantiation is application specific, dependent on factors such as security requirement, and efficiency.

C. Cryptographic Access Control Mechanisms for Cloud Computing

Attribute-based encryption is by nature a tool for access control, and thus it is not surprising to see that the primitive is adapted to the setting of cloud computing, where fine-grained access control is desired so as to make the cloud a scalable and convenient platform for data sharing. Yu *et al.* led this line of research [31]. As mentioned earlier, they combine the KP-ABE scheme [18] and the proxy re-encryption scheme [4], and manage to push the workload of key re-distribution and data re-encryption resulting from user revocation to the cloud by exploiting the specific structures of the two primitives. Their scheme represents a considerable improvement over the trivial solution, where the data owner shares the decryption key with all authorized users, and takes the full responsibility for key re-distribution and data re-encryption in case of user revocation.

Wang *et al.* realized hierarchical attribute-based encryption [30] for cloud storage by enhancing CP-ABE with the technique of hierarchical identity-based encryption. Hierarchical attribute-based encryption clearly is able to cope with more complicated application requirements. Their scheme also enjoys the advantage of delegating heavy workload to the cloud. A weakness of this scheme, however, is that it cannot efficiently support compound attributes, where there exist many sets of values satisfying the policy specified by a combination of attributes.

To support compound attributes, Liu *et al.* proposed hierarchical attribute-set-based encryption for cloud computing [24]. Their construction extends the Ciphertext-Policy Attribute-Set-Based Encryption [7] with the support of hierarchical access structures. To achieve efficient user revocation support, the scheme employs a "multiple value assignment" method upon the expiration-time attribute, which in turn entails a update of partial keys and partial re-encryption of data.

By and large, each of the above proposals following the line of enforcing access control for cloud storage with attribute-based encryption depends on a certain specific ABE scheme, and attains relatively good performance for user revocation by taking advantage of the particular structure of the ciphertext of the ABE scheme. In contrast, our scheme

[2] In the context of proxy re-encryption, Alice is called delegator and Bob is delegatee

is generic, not depending on any specific ABE schemes. In fact, the instantiation of our construction is application specific, in that any ABE scheme/predicate encryption scheme or other encryption primitive can be used as long as it meets the access control requirement of the underlying application.

Finally, we mention a proposal by Zhao *et al.*, which is an access control scheme for cloud computing using a different encryption primitive, called progressive elliptic curve encryption [32]. The proposal actually employs an interactive data sharing procedure, where an authorized user has to interact realtime with the data owner so as to decrypt an encrypted data record requested from the cloud. This requires that the data owner has to be online all the time, which offsets to a great extent the advantage of cloud computing where the data owner delegates the management of her data to the cloud.

III. System Model and Security Requirements

A. *System Model*

We consider the use scenario of cloud computing as shown in Figure 1, where a data owner outsources her data to the cloud and authorizes a group of data consumers to access the data. As such, the system includes the following players: the Data Owner (DO), the Cloud (CLD), a set of Data Consumers. As the system assumes a public key setting, where every user (the data owner and the data consumers) has a public/private key pair, there is also an implicit Certificate Authority (CA), who certifies users' public keys.

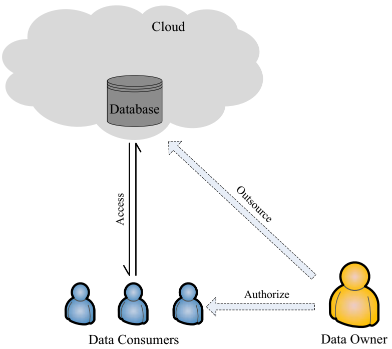

Figure 1. System Model

More specifically, the data owner outsources her data to the cloud for storage and management (by instructing the cloud for updating the database, e.g., add/delete records); the data owner also takes charge of managing authorization of the data consumers to access her data. For security reasons, the data in outsourcing is encrypted by the data owner against the cloud. To facilitate the enforcement of fine-grained access control, a data record is associated with a set of attributes which are meaningful in the context, upon which data encryptions are performed in a way that users' access privileges are specified and differentiated. Each valid data consumer is issued a decryption key by the data owner according to his access right, which can be used to access the encrypted data.

B. *Security Requirements*

From the above discussions, the system is designed for the data owner to share her data with a set of data consumers in accordance with their respective access rights. Hence the major security objective is to achieve data *confidentiality* against unauthorized entities. In particular, we impose two security requirements upon the system as formulated below.

Confidentiality against cloud. While the data owner entrusts the management of her data to the cloud, she does not want the cloud to access the data. The cloud is thus one of the main adversaries against the confidentiality of the outsourced data. In fact, the cloud represents an extreme of unauthorized users (i.e., outsiders), in terms of adversarial capabilities. Indeed, the cloud sees all the encrypted data as well as the replies to the authorized users by processing user access requests; the information gained by the cloud throughout the process is thus by no means less than any unauthorized users.

More formally, this requirement can be specified using the formulation of regular public key encryption security (i.e., CPA or CCA security), extended with the provision of a *transformation* oracle to the adversary, where the oracle transforms any encrypted data provided by the adversary into an access reply of a valid data consumer. The transformation oracle actually formulates the fact that the cloud is entrusted to process the data consumers' access requests. We omit the detailed formalism to avoid verbosity, as the definition of CPA/CCA-security of pubic key encryption is well documented.

Caveat. While the cloud is purported to be the adversary to data confidentiality, it should behave honestly in terms of managing the data owner's data, processing users' access requests, and other administrative activities. For this reason, we are actually considering the cloud to be *honest but curious*. In addition, we do not consider coalition of the cloud and valid/revoked data consumers, as this will complicate the assumption of honest-but-curious cloud. As a matter of fact, as we will see later, in our scheme the cloud colluding with a data consumer cannot get more information than what the data consumer has already been authorized.

Confidentiality against accesses beyond authorized rights. As the data owner wants to enforce fine-grained access control over her data in outsourcing, a natural security requirement is that a data consumer cannot obtain data beyond his authorized access right. The formulation of this requirement is almost identical to the security definition of attribute-based encryption or predicate encryption, except for the provision of the transformation oracle to the adversary like the above. We again omit the detailed formalism, as it becomes straightforward when checking any of the aforementioned related literature on attribute-based encryption/predicate encryption. We should stress that when an authorized consumer is revoked, his/her access right is forfeited and he/she becomes no different from an outsider.

IV. Our Scheme

Our construction makes use of attribute-based encryption (or any variants that can implement fine-grained access control such as predicate encryption) and proxy re-encryption. We thus begin with a brief review of the semantics of these primitives.

A. Preliminaries

Attribute-based encryption: An attribute-based encryption scheme ABE consists of the following functions.

- ABE.Setup: takes as input the security parameter 1^κ and outputs a public key PK and a (master) secret key SK.
- ABE.KeyGen: takes as input SK and a user's access privileges, and outputs a user secret key sk_u.
- ABE.Enc: takes as input PK, a message m, and the access control policy pol, and outputs an ciphertext c. We denote the function as $c = \mathsf{ABE}.\mathrm{Enc}_{PK}(pol, m)$.
- ABE.Dec: takes as input a user secret key sk_u and a ciphertext c, and outputs either a message m if the access privileges associated with sk_u match the access control policy associated with the ciphertext, or a special symbol $\perp$ otherwise.

Proxy re-encryption: A proxy re-encryption scheme PRE is composed of the following algorithms.

- PRE.Setup: takes as input the security parameter 1^κ and outputs the global parameters $param$, which include a description of the plaintext space.
- PRE.KeyGen: take as input $param$ and a user identifier u, and outputs a public/private key pair (pk_u, sk_u).
- PRE.ReKeyGen: takes as input the private key sk_u of the delegator and the public key pk_v of the delegatee, and outputs a re-encryption key $rk_{u\to v}$, which can transform ciphertexts under pk_u to those under pk_v.
- PRE.Enc: takes as input a public key pk and a plaintext m, and outputs a ciphertext c[3]. We denote this as $c =\mathsf{PRE}.\mathrm{Enc}_{pk}(m)$.
- PRE.ReEnc: takes as input a re-encryption key $rk_{u\to v}$ and a ciphertext c_u under public key pk_u, and outputs another ciphertext c_v under public key pk_v. This is denoted as $c_v =\mathsf{PRE}.\mathrm{ReEnc}(c_u, rk_{u\to v})$.
- PRE.Dec: takes as input a private key sk and a ciphertext c under pk, and outputs a plaintext m.

B. An Overview

To enable the data owner to share her data over the cloud in a fine-grained manner, attribute-based encryption (or any of its variants) is an ideal tool. However, the main issue with the use of attribute-based encryption is how to efficiently handle user revocation. As we mentioned earlier, existing proposals along this line [24], [30], [31] solve this problem by looking into the specific structure of the ciphertext of a certain attribute-based encryption scheme, and delegating the heavy task of key re-distribution and data re-encryption to the cloud. We desire a more general and effective solution.

Our method to address user revocation is to use proxy re-encryption: if a user Bob is a valid data consumer at the discretion of the data owner Alice, a re-encryption key $rk_{A\to B}$ is generated by Alice and is given to the cloud, which helps transform ciphertexts under Alice's public key to ones under Bob's public key; once Bob is revoked, Alice simply commands the cloud to destroy the re-encryption key $rk_{A\to B}$.

Combining attribute-based encryption and proxy re-encryption, the rationale of our construction is briefly described as follows. Data encryption follows folklore of hybrid public key encryption: symmetric key encryption is used for encryption of the actual data, while public key encryption is for key encapsulation. More specifically, to encrypt a data record d, the data owner selects a random key k for symmetric encryption, and encrypts d with the symmetric key encryption; picks another random key k_1, and computes $k_2 = k \otimes k_1$; then encrypts k_1 using attribute-based encryption, and encrypts k_2 with proxy re-encryption under her own public key. It is clear that d can be obtained only if both k_1 and k_2 are present. When a data consumer Bob requests a data record, if he is valid, the cloud should possess a re-encryption key $rk_{A\to B}$; as such, the cloud transforms the part of the ciphertext corresponding proxy re-encryption into one under his public key. Bob thus can obtain k_2, and can also get k_1 as long as his access privileges meet the access control policy governing the encryption of k_1 under attribute-based encryption.

C. Detailed Construction

We are assuming a public key encryption setting, where each of the data owner and the data consumers u possesses a public/private key pair (pk_u, sk_u) certified by the certificate authority, and the key pair admits proxy re-encryption PRE (i.e.,users' key pairs are generated by PRE.KeyGen.). Our scheme consists of the following procedures.

Setup. This is the system initialization step executed by the data owner.

- Determines a suitable attribute-based encryption scheme ABE according to factors such as the system's requirements on the fine-grainedness of access control, the level of security, and efficiency; executes

[3] More precisely, a proxy re-encryption scheme distinguishes between first-level encryption and second-level encryption, and only second-level encryption can be transformed (into first-level encryption). To avoid confusion, we intentionally do not differentiate in this work, and PRE.Enc here is in fact second-level encryption.

ABE.Setup to obtain the master public/secret key (PK, SK) of ABE.
- Selects an appropriate block cipher $E()$ such as AES.
- Publishes all the public system information (including the public key of ABE and description of $E()$), e.g., gives them to the cloud.

New Data Record Generation. This is the procedure, whereby the data owner (say Alice) encrypts a data record d and makes it ready for outsourcing to the cloud.

- Picks a random key k and encrypts the data record d as $E_k(d)$.
- Determines and associates an access control policy pol with the data record; picks a random value k_1, and encrypts it using attribute-based encryption as ABE.Enc$_{PK}(pol, k_1)$.
- Computes $k_2 = k \otimes k_1$; encrypts k_2 with her public key pk_A under proxy re-encryption as PRE.Enc$_{pk_A}(k_2)$. Then an encrypted data record is in the form $\langle c_1, c_2, c_3\rangle = \langle$ABE.Enc$_{PK}(pol, k_1)$, PRE.Enc$_{pk_A}(k_2), E_k(d)\rangle$
- Passes the record $\langle c_1, c_2, c_3\rangle$ to the cloud for storage and management.

User Authorization. In this procedure, the data owner authorizes a user privileges to access some of her data that are managed at the cloud, and the user becomes a valid data consumer.

- Decides the access privileges the user, say Bob, should have, and accordingly executes ABE.KeyGen to generate a secret key ABE.sk_B, and passes it secretly to Bob.
- Executes PRE.ReKeyGen to generate a re-encryption $rk_{A\to B}$, and passes it secretly to the cloud, who adds a new entry $(\text{Bob}, rk_{A\to B})$ to the *authorization list* which contains similar entries for all authorized data consumers.

Data Access. Upon receipt of a data access request from a user, say Bob, the cloud does the following:

- Check the authorization list to see if there is an entry $(\text{Bob}, rk_{A\to B})$ for Bob. If yes, for each record $\langle c_1, c_2, c_3\rangle$ requested by Bob, computes $c_2' =$ PRE.ReEnc$(c_2, rk_{A\to B})$, and replies $\langle c_1, c_2', c_3\rangle$ to Bob. If no entry is found for Bob, simply aborts.

At the consumer's side, Bob does the following:

- Decrypts c_1 using ABE.sk_B to get k_1.
- Decrypts c_2' with sk_B for proxy re-encryption to get k_2.
- Computes $k = k_1 \otimes k_2$, and uses k to decrypt c_3.

User Revocation. To revoke a data consumer, the data owner simply informs the cloud to erase the entry corresponding to the consumer from the authorization list.

Data Deletion. To remove a data record from sharing, the data owner simply instructs the cloud to erase the record from the database.

D. Correctness

The correctness of the scheme is straightforward. In an access reply $\langle c_1, c_2', c_3\rangle$ returned to Bob, c_2' is a ciphertext (proxy re-encryption) under Bob's public key, so he (and only he) can decrypt using his own private key; c_1 is a ciphertext of attribute-based encryption, so if has been granted the due privileges by the data owner, he should possess the corresponding secret key for decryption. With the ability to decrypt both elements, he can certainly decrypt c_3.

E. Performance

We do not intend to compare our scheme with other existing proposals (e.g., those reviewed in Section II-C) in terms of performance. This is because our scheme is a generic construction, while others are specific ones. Thus, there does not exist a fair base for comparison. We list in Table I the computation performance of the main operations of our scheme. The statistics show that the scheme is rather computationally efficient.

Operation	Computation Cost
New Record Generation	ABE.Enc + PRE.Enc
User Authorization	ABE.KeyGen + PRE.ReKeyGen
Data Access (per record)	Cloud: PRE.ReEnc Consumer: ABE.Dec + PRE.Dec
User Revocation	$\mathcal{O}(1)$
Data Deletion	$\mathcal{O}(1)$

Table I
COMPUTATION PERFORMANCE

For ciphertext size expansion, the length of a ciphertext in our scheme elongates the size of the original data record by $|$ABE.Enc$| + |$PRE.Enc$|$ bits.

F. Security Analysis

We analyze that our scheme satisfies the security requirements set out in Section III.

Confidentiality against cloud. Data encryption in our scheme follows folklore of hybrid KEM (Key Encapsulation Mechanism) + DEM (Data Encapsulation mechanism), i.e., using public key encryption for key encapsulation and symmetric key encryption for data encapsulation. The security of the paradigm of KEM + DEM has been well studied, e.g., [10], [12], [14], [19], [20]. Thus confidentiality (against cloud) of our scheme follows directly from these well-established results.

We point out that in our scheme, the cloud acts as the proxy of the proxy re-encryption scheme, so in the formulation of this security requirement (i.e., confidentiality against cloud), we provide a transform oracle to the adversary, which formulates the fact that the adversary holds re-encryption keys (see Section III). Also for this reason, the cloud actually represents the most powerful extreme of the unauthorized users, in terms of adversarial capabilities.

Confidentiality against accesses beyond authorized rights. In our scheme, the enforcement of fine-grained access control rests exclusively on attribute-based encryption. It is clear that the security of our scheme in terms of confidentiality against accesses beyond authorized rights follows directly from the underlying attribute-based encryption scheme.

Remark. While we do not formally consider collusion of revoked data consumers and the cloud, it is easy to see that for a revoked user, if the cloud has erased the corresponding re-encryption key, then the collusion between the two entities does not in any way help the the revoked user. However, if the cloud intentionally deviates from the rules by keeping a copy of the re-encryption key of the revoked user, then the collusion cannot get more information than what has been authorized to the revoked consumer.

G. Discussions

In this section, we discuss the advantages of our scheme, to justify our claims made in Section I on the features of our construction (See "Our Contribution").

Generic Construction. Our scheme is a generic construction, not restricted to any specific attribute-based encryption scheme and proxy re-encryption scheme. This offers an instantiation of our scheme the flexibility to choose schemes with appropriate fine-grainedness and proper level of security, with respect to the requirements of the underlying application. For example, it is not necessary to adopt an attribute-based encryption scheme with finer-grained access control if the underlying application does not require that level of fine-grainedness. Another example is that if data encryption with CPA security suffices for an application, then it makes no sense to employ a CCA secure attribute-based encryption scheme and proxy re-encryption scheme. One more flexibility provided by the generic construction is that an instantiation are entitled to choose the most efficient cryptographic scheme within its class satisfying a certain level of fine-grainedness and security.

Efficient User Revocation. User revocation in our scheme does not result in any key re-distribution and data re-encryption. Non-revoked users are not affected at all. All that caused by user revocation is that the data owner simply instructs the cloud to erase the the re-encryption key of the revoked user.

Stateless Cloud. It is a clear fact that the cloud in our scheme is not required to retain any information related to user revocation.

Direct Security Guarantee. Our scheme does not make any change to the adopted attribute-based encryption scheme, proxy re-encryption scheme, and symmetric key encryption scheme. Hence, the security properties of the respective cryptographic primitives are well respected, and the security of the whole construction derives directly from the security properties of these underlying primitives.

H. Potential Issues

It is clear that our scheme can efficiently revoke an authorized consumer's access privileges, and turn him/her into an outsider. However, it appears that the scheme is not competent in dealing with the scenarios that a revoked user rejoins the system and is authorized with different access privileges. This is due to the fact that the revoked user still holds the decryption key corresponding to attribute-based encryption; when provided with the decryption capability for the proxy re-encryption part (e.g., by rejoining the system), the revoked user will re-gain the access privileges associated with the attributed-based encryption part. We attribute this problem to the "loose" combination between attribute-based encryption and proxy re-encryption. A remedy must seek to seamlessly embed proxy re-encryption to attribute-based encryption (or vice versa), e.g., attribute-based proxy re-encryption, which is our future work.

For the same reasons, a collusion between a revoked consumer and an authorized consumer helps the two entities possess the access privileges once assigned to the revoked consumer. This is probably not a desired consequence for a practical system. The same remedy solution can help address the issue.

V. Conclusion

Cloud computing is a new revolution in IT, and has the potential to reshape the business model of the IT industry. Storage services are a fundamental component of the cloud computing paradigm. In using the storage services, users outsource their data to the cloud, and share the data with many other data consumers. Associated with the data in outsourcing is the concern that the cloud may abuse the data without the consent of the data owners, as the data physically reside outside the data owners' domain. Encryption of the data in outsourcing promises a good solution to this problem. However, data encryption would require key re-distribution and data re-encryption in case of user revocation. In this work, we propose a generic scheme towards solving this issue while implementing fine-grained data sharing. User revocation in our construction does not result in any key re-distribution and data re-encryption whatsoever. Our scheme demonstrate advantages over many other proposals of its kind. On the other hand, there are potential issues unsolved in our scheme, which will be our future work.

Acknowledgment

The authors would like to thank Xuhua Ding, Singapore Management University, for pointing to us the potential issue arising from a revoked user re-joining the system.

REFERENCES

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. *Improved proxy re-encryption schemes with applications to secure distributed storage*, Proc. Network & Distributed System Security Symposium, NDSS'05, pp. 29-43, 2005.

[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. *Improved proxy re-encryption schemes with applications to secure distributed storage*, ACM Transactions on Information and System Security (TISSEC), Vol 9(1), pp. 1-30, 2006.

[3] D. Boneh, and X. Boyen. *Efficient selective-ID secure identity based encryption without random oracles*. Proc. Advances in Cryptology - Eurocrypt'04, pp. 223-238, 2004.

[4] M. Blaze, G. Bleumer, and M. Strauss. *Divertible protocols and atomic proxy cryptography*, Proc. Advances in Cryptology - Eurocrypt'98, 1998.

[5] D. Boneh, and M. Franklin. *Identity based encryption from the Weil pairing*, Proc. Advances in Cryptology - Crypto'01, pp. 213-229, 2001.

[6] D. Boneh, E.-J. Goh, and T. Matsuo. *Proposal for P1363.3 Proxy Re-encryption*, http://grouper.ieee.org/groups/1363/IBC/submissions/ NTTDataProposal-for-P1363.3-2006-08-14.pdf.

[7] R. Bobba, H. Khurana, and M. Prabhakaran. *A pracitically motivated enhancement to attribute-based encryption*. Proc. European Symposium on Research in Computer Security, Esorics'09, 2009.

[8] J. Bethencourt, A. Sahai, and B. Waters. *Ciphertext-policy attribute-based encryption*. Proc. IEEE Symposium on Security & Privacy, S&P'07.

[9] R. Caneti and S. Hohenberger. *Chosen-ciphertext secure proxy re-encryption*. Proc. ACM Computer and Communications Security Conference, CCS'07, pp. 185-194, 2007.

[10] S. Choi, J. Herranz, D. Hofheinz, J. Hwangd, E. Kiltz, D. Lee, M. Yung. *The Kurosawa-Desmedt key encapsulation is not chosen-ciphertext secure*. Information Processing Letters Vol. 109(16), pp. 897-901, 2009.

[11] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, *Controlling data in the cloud: outsourcing computation without outsourcing control*. Proc. IEEE 3^{rd} International Conference on Cloud Computing, pp. 85-90, 2010.

[12] R. Cramer, V. Shoup. *Design and analysis of practical public-key encryption schemes secre against adaptive chosen ciphertext attack*. SIAM Journal on Computing, Vol 33(1), pp. 167-226, 2003.

[13] C. Chu, W. Tzeng. *Identity-based proxy re-encryption without random oracles*, Proc. Information Security Conference, ISC'07, pp. 189-202, 2007.

[14] A.W. Dent. *A designer's guide to KEMs*. Proc Cryptography and Coding: IMA international Conference, LNCS 2898, pp. 133-151, 2003.

[15] R. H. Deng, J. Weng, S. Liu, and K. Chen. *Chosen-ciphertext secure proxy re-encryption without pairings*. Proc. International Conference on Cryptography and Network Security, CANS'08, pp. 1-17, 2008.

[16] T. ElGamal. *A public-key cryptosystem and a signature scheme based on discrete logarithms*. Proc. Advances in Cryptology - Crypto'84, pp. 10-18, 1984.

[17] M. Green, G. Ateniese. *Identity-based proxy re-encryption*. Proc. International Conference on Applied Cryptography and Network Security, ACNS'07, pp. 288-306, 2007.

[18] V. Goyal, O. Pandy, A. Sahai, and B. Waters. *Attribute-based encryption for fine-grained access control of encrypted data*, Proc. ACM Computer and Communications Security Conference, CCS'06.

[19] J. Herranz, D. Hofheinz, and E. Kiltz. *KEM/DEM: necessary and sufficient conditions for secure hybrid encryption*. http://eprint.iacr.org/2006/265, 2006.

[20] K. Kurosawa, and Y. Desmedt. *A new paradigm of hybrid encrption scheme*. Proc. Advances in Cryptology - Crypto'04, LNCS 3152, pp. 426-442, 2004.

[21] S. Kamara and K. Lauter, *Cryptographic cloud storage*. Proc. Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010.

[22] J. Katz, A. Sahai, and B. Waters, *Predicate encryption supporting disjunctions, polynomial equations, and inner products*. Proc. Advances in Cryptology - EUROCRYPT'08, LNCS 4965, pp. 146-162, 2008.

[23] A. Lewko, T. Okamoto, A. Sahai, T. Takashima, and B. Waters. *Fully secure funtional encryption: attibute-based encryption and (hierarchial) inner product encryption*. Proc. Advances in Cryptology - Eurocrypt, LNCS 6110, pp. 62-91, 2010.

[24] J. Liu, Z. Wan, M. Gu. *Hierarchical attribute-set based encryption for scalable, flexible and fine-grained access control in cloud computing*. Proc. 7^{th} Information Security Practice and Experience Conference, ISPEC'11.

[25] B. Libert and D. Vergnaud. *Unidirectional chosen-ciphertext secure proxy re-encryption*, Proc. Public Key Cryptography, PKC'08, pp. 360-379, 2008.

[26] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*. O'Reilly Media, 2009.

[27] T. Okamoto, and K. Takashima. *Hierarchical predicate encrypton for inner-products*. Proc. Advances in Cryptology - ASIACRYPT'09, LNCS 5912, pp. 214-231, 2009.

[28] E. Shi, and B. Waters. *Delegating capabilities in predicate encryption systems*. Proc. International Colloquium on Automata, Languages and Programming, ICALP'08, pp. 560-578, 2008.

[29] A. Sahai, and B. Waters. *Fuzzy idnentity based encryption*. Proc. Advances in Cryptology - Eurocrypt'05, LNCS 3494, pp. 457-473, 2005.

[30] G. Wang, Q. Liu, and J. Wu. *Hierarhical attribute-based encryption for fine-grained access control in cloud storage services*. Proc. ACM conference on Computer and Communications Security, CCS'10.

[31] S. Yu, C. Wang, K. Ren, and W. Lou. *Achieving secure, scalable, and fine-grained data access control in cloud computing*, Proc. IEEE International Conference on Computer Communications, INFOCOM'10.

[32] G. Zhao, C. Rong, J. Li, F. Zhang, and Y. Tang. *Trusted data sharing over untrusted cloud storage providers*. Proc. 2^{nd} IEEE International Conference on Cloud Computing Technology and Sciene, pp. 96-103, 2011.