ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 5, May 2016

Audit-Free Cloud Storage with Confutable AB Encryption

[1]T.Nagaraju [2] M .S.R. Laksmi Reddy [3]S.Vidyullatha

1Assistant Professor, Dept of CSE ,CMRIT Kandlakoya(v), Hyderabad, India

2Assistant Professor, Dept of CSE ,CMRIT Kandlakoya(v), Hyderabad, India.

3 Assistant professor, Dept of CSE, CMRIT Kandlakoya(v), Hyderabad, India

Abstract--In this project, Cloud storage services are becoming growingly popular. Because of the importance of privacy, many encryption schemes in the cloud storage have been initiated to preserve data from those who do not have access. All such schemes believed that cloud storage providers are safe and cannot be hacked; although, in practice, some sovereignty (i.e., coercers) may force cloud storage providers to betray user information or confidential data on the cloud, thus altogether bypass storage encryption schemes. In this

paper, we present our design for a advanced cloud storage encryption scheme that enables cloud storage providers to create dummy user secrets to protect user privacy. Since coercers cannot tell if acquire secrets are true or not, the cloud storage providers protect that user privacy is still secure..

I. INTRODUCTION

Cloud storage services have rapidly become increasingly popular. Users can store their data on the cloud and access their data anywhere at any time. Because of user privacy, the data stored on the cloud is typically encrypted and protected from access by other users. Considering the collaborative property of the cloud data, attribute-based encryption (AB ENCRYPTION) is regarded as one of the most suitable encryption schemes for cloud storage. There are numerous AB ENCRYPTION schemes that have been proposed, including [1].

Most of the proposed schemes assume cloud storage service providers or trusted third parties handling key management are trusted and cannot be hacked; although, in experimentally, some entities may intercept Communications between users and (csp) cloud storage providers and then force storage providers to release user secrets by using government power or other means. In this case, encrypted data are assumed to be known and storage providers are requested to release user secrets. As an example, in 2010, without notifying its users, Google released user documents to the FBI after receiving a search warrant. In 2013, Edward Snowden disclosed the existence of global surveillance programs that collect such cloud data as emails, texts, and voice messages from some technology companies. Once cloud storage providers are compromised, all encryption schemes lose their effectiveness. Though we hope cloud storage providers can fight against such entities to maintain user privacy through legal way, it is seemingly more and more difficult. As one example, Lavabit was an email service company that protected all user emails from outside compulsion; unfortunately, it failed and decided to

shut down its email service Since it is difficult to fight against outside compulsion, we aimed to build an encryption scheme that could help cloud storage providers avoid this predicament. In our approach, we offer cloud storage providers means to create fake user secrets. Given such fake user secrets, outside coercers can only obtained forged data from a user's stored ciphertext. Once coercers think the received secrets are real, they will be satisfied and more importantly cloud storage providers will not have revealed any real secrets. Therefore, user privacy is still protected.

This concept comes from a special kind of encryption scheme called **confutable encryption**, first proposed in confutable encryption involves senders and receivers creating convincing fake evidence of forged data in ciphertexts such that outside coercers are satisfied. Note that deniability comes from the fact that coercers cannot prove the proposed evidence is wrong and therefore have no reason to reject the given evidence. This approach tries to altogether block coercion efforts since coercers know that their efforts will be useless. We make use of this idea such that cloud storage providers can provide audit-free storage services. In the cloud storage scenario, data owners who store their data on the cloud are just like senders in the deniable encryption scheme. Those who can access the encrypted data play the role of receiver in the deniable encryption scheme, including the cloud storage providers themselves, who have systemwide secrets and must be able to decrypt all encrypted data1.

In this work, we describe a deniable AB ENCRYPTIONncryption scheme for cloud storage services. We make use of AB ENCRYPTIONncryption characteristics for securing stored data with a fine-grained access control mechanism and deniable encryption to prevent outside auditing. Our scheme is based on Waters ciphertext policy-attribute based encryption (CP-AB ENCRYPTION) scheme [4]. We enhance the Waters scheme from prime order bilinear groups to composite order bilinear groups. By the subgroup decision problem assumption, our scheme enables users to be able to provide fake secrets that seem legitimate to outside coercers.

1.1 Previous Work on AB Encryption

Sahai and Waters first introduced the concept of AB Encryption in which data owner can embed how they want to share data in terms of encryption [1]. That is, only those who match the owner's conditions can successfully decrypt stored data. We note here that AB ENCRYPTION is encryption for privileges, not for users. This makes AB ENCRYPTION a very useful tool for cloud storage services since data sharing is an important feature for such services. There are so many cloud storage users that it is impractical for data owners to encrypt their data by pairwise keys. Moreover, it is also impractical to encrypt data many times for many people. With AB ENCRYPTION, data owners decide only which kind of users can access their encrypted data. Users who satisfy the conditions are able to decrypt the encrypted.

Data There are two types of AB Encryption, CP-AB Encryption and Key-Policy AB ENCRYPTION (KP-AB ENCRYPTION). The difference between these two lies in policy checking. KP-AB ENCRYPTIONncryption is an AB ENCRYPTION in which the policy is embedded in the user secret key and the attribute set is embedded in the ciphertext. Conversely, CP-AB ENCRYPTION embeds the policy into the ciphertext and the user secret has the attribute set. Goyal et al. proposed the first KPAB ENCRYPTION in [2]. They constructed an expressive way to relate any monotonic formula as the policy for user secretkeys. Bethencourtet al. proposed the first CP-AB ENCRYPTION in [3]. This scheme used a tree access structure to express any monotonic formula over attributes as the policy in the ciphertext. The first fully expressive CP-AB ENCRYPTION was proposed by Waters in [4],

which used Linear Secret Sharing Schemes (LSSS) to build a ciphertext policy. Lewko et al. enhanced the Waters scheme to a fully secure CP-AB ENCRYPTION, though with some efficiency loss, in [13]. Recently, Attrapadung et al. constructed a CP-AB ENCRYPTION with a constant-size ciphertext in [14] and Tysowski . designed their CP-AB ENCRYPTION scheme for resource-constrained users in [7].

1.2 Previous Work on Deniable Encryption

The concept of deniable encryption was first proposed in [12]. Like normal encryption schemes, deniable encryption can be divided into a deniable shared key scheme and a public key scheme. Considering the cloud storage scenario, we focus our efforts on the deniable public key encryption scheme There are some important deniable public key encryption schemes2. Canetti et al. used translucent sets to construct deniable encryption schemes in [12]. A translucent set is a set containing a trapdoor subset. It is easy to randomly pick an element from the universal set or from the subset; however, without the trapdoor, it is difficult to determine if a given element belongs to the subset. Canetti et al. showed that any trapdoor permutation can be used to construct the translucent set.To build a deniable public key encryption scheme from a translucent set, the translucent set is the public key and the trapdoor is the private key. The translucent setis used to represent one encrypted bit. Elements in the subset are represented by 1 whereas other non-subset elements are represented by 0. The sender can encrypt1 by sending an element in the subset, but can claim the element is chosen from the universal set (i.e., 0). The above is a basic sender-deniable scheme. Canetti et al. also proved that a sender-deniable scheme can be transformed to a receiver-deniable scheme or a bideniable scheme with the help of intermediaries. There is research on how best to design a translucent set. Durmuth et al. designed the translucent set from the samplable encryption in [15]. ONeill et al. designed the bi-translucent set from a lattice in [16], which can build a native bi-deniable scheme.

In addition to the bitranslucent set, there are other proposed approaches to building deniable encryption schemes. ONeill et al. proposed a new deniable method through a simulatable public key system [16]. The simulatable public key system provides an oblivious key generation function and an oblivious ciphertext function. When sending an encrypted bit, the sender will send a set of encrypted data which may be normally encrypted or oblivious. Therefore, the sender can claim some sent messages are oblivious while actually they are not. The idea can be applied to the receiver side such that the scheme is a bi-deniable scheme. In [17], Gastietal. proposed another deniable scheme in which one public private key pair is set up for each user while there are actually two pairs. The sender can send a true message encrypted by one key with a fake message encrypted by the other key. The sender decides which key is released according to the coercer's identity. Gasti et al. also applied this idea to cloud storage services. There are still other deniable encryption schemes, including [18].

Aside from the above deniable schemes, there is research investigating the limitations of the deniable schemes. In 0]. Nielsen states that it is impossible to encrypt unbounded messages by one short key in non-committing schemes, including deniable schemes. In , Bendlin et al. shows that noninteractive and fully receiver-deniable schemes cannot be achieved simultaneously. We construct our scheme under these limitations.

1.3 Our Contributions

In this work, we construct a deniable CP-AB ENCRYPTION scheme that can make cloud storage services secure and auditfree. In this scenario, cloud storage service providers are just regarded as receivers in other deniable schemes. Unlike most previous deniable

encryption schemes, we do not use translucent sets or simulatable public key systems to implement deniability. Instead, we adopt the idea proposed in [17] with some improvements. We construct our deniable encryption scheme through a multidimensional space. All data are encrypted into the multidimensional space. Only with the correct composition of dimensions is the original data obtainable. With false composition, ciphertexts will be decrypted to predetermined fake data. The information defining the dimensions is kept secret. We make use of composite order bilinear groups to construct the multidimensional space. We also use chameleon hash functions to make both true and fake messages convincing.

Our deniable AB ENCRYPTIONncryption has the advantages described below over previous deniable encryption schemes.

• **Blockwise Deniable AB ENCRYPTION**. Most deniable public key schemes (e.g., [12], [15], [16]) are bitwise, which means these schemes can only process one bit a time; therefore, bitwise deniable encryption schemes are inefficient for real use, especially in the cloud storage service case. To solve this problem, O'Neilet al. designed a hybrid encryption scheme that simultaneously uses symmetric and asymmetric encryption. They use a deniably encrypted plan-ahead symmetric data encryption key, while real data are encrypted by a symmetric key encryption mechanism. This reduces the repeating number from the block size to the key size. Though bitwise deniable encryption is more flexible than blockwise deniable encryption in "cooking" fake data, when considering cloud storage services, blockwise encryption is much more efficient in use.

Unlike those techniques used in previous deniable encryption schemes, we build two encryption environments at the same time, much like the idea proposed in [17].We build our scheme with multiple dimensions while claiming there is only one dimension. This approach removes obvious redundant parts in [17]. We apply this idea to an existing AB ENCRYPTION scheme by replacing prime order groups with composite order groups. Since the base AB ENCRYPTION scheme can encrypt one block each time, our deniable CPAB ENCRYPTION is certainly a blockwise deniable encryption scheme. Though the bilinear operation for the composite order group is slower than the prime order group, there are some techniques that can convert an encryption scheme from composite order groups to prime order groups for better computational performance, such as those described in and . We use composite order groups to describe our idea in Section 4 and transform it to prime order groups

Consistent Environment. Most of the previous deniable encryption schemes are inter-encryptionindependent. That is, the encryption parameters should be totally different for each encryption operation. If two deniable encryptions are performed in the same environment, the latter encryption will lose deniability after the first encryption is coerced, because each coercion will reduce flexibility. For example, once coercers get private keys, which are the most common receiver proofs, these keys should be convincing not only under some particular files, but also under all related stored data. Otherwise, the coercers will know that these keys are fake; however, all proposed schemes only provide convincing proofs for particular transmissions. In the secure cloud storage service, this is not practical. It is impossible for a cloud storage service provider to prepare a unique encryption environment for each file, much less to maintain the access control mechanism at the same time.

In this work, we build a consistent environment for our deniable encryption scheme. By consistent environment, we means that one encryption environment can be used for multiple encryption times without system updates. The opened receiver proof should look convincing for all

ciphertexts under this environment3, regardless of whether a ciphertext is normally encrypted or deniably encrypted. The deniability of our scheme comes from the secret of the subgroup assignment, which is determined only once in the system setup phase. By the canceling property and the proper subgroup assignment, we can construct the released fake key to decrypt normal ciphertexts correctly.

• **Deterministic Decryption**. Most deniable encryption schemes have decryption error problems. These errors come from the designed decryption mechanisms.For example, in [12], Canetti et al. uses thesubset decision mechanism for decryption. The receiverdetermines the decrypted message accordingto the subset decision result. If the sender chooses anelement from the universal set but unfortunately theelement is located in the specific subset, then an error occurs. The same error occurs in all translucentset-based deniable encryption schemes. , which uses a voting mechanism fordecryption. Decryption is correct if and only if thecorrect part overwhelms the false part. Otherwise,the receiver will get the error result.

The concept of our deniable scheme is different than these schemes described above. Our scheme extends a pairing AB ENCRYPTION, which has a deterministic decryption algorithm, from the prime order group to the com-posite order group. The decryption algorithm in our scheme is still deterministic; therefore, there is no decryption errors using our scheme.

PERFORMANCE EVALUATION

In this section, we evaluate the performance of our idea by implementing two deniable schemes: the composite order scheme and the prime order simulation scheme. We compare them with the Waters scheme [4]. We use the Pairing Based Cryptography (PBC) library for cryptographic operations. We use type A1 pairing because this type of pairing can support both prime order and composite order groups. In our experiment, we set the size of each prime to 512 bits, which is equal to 256 bits of security. Under this setting, the composite group order size is 1536 bits. However, when considering security, the composite order scheme with a group size of 1536 bits is equal to the prime order scheme with a group size of 512 bits. This is because a message is encrypted in one subgroup whose group size is 512 bits. Our experiments focus on encryption and decryption performance. The Setup and KeyGen performance are skipped because these two algorithms are not time critical. The four Open algorithms are low-cost algorithms

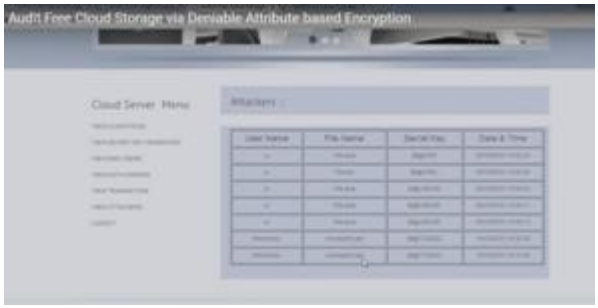

Fig1: performance

because these algorithms only return existing information. The cost of Verify algorithm is equal to that of Dec. Note that we do not distinguish deniable encryption from normal encryption; their numbers of arithmetic operations and pairing operations are equal, and therefore the normal one and the deniable one will have similar performance. In our design, the encryption cost and the decryption cost depend on required attribute numbers. For convenience, we make all attributes mandatory as our cryptographic policy. We run the experiments with different attribute numbers, from 10 to 1000. Our experiments focus on one block encryption/decryption. Each block is set to 128 bytes because PBC reads around 130 bytes to generate a GT element when the group size is 512 bits5. A large file can be divided into multiple blocks, and all blocks can be protected by one secret s. Because GT multiplication and H

are lightweight operations, we use one-block encryption/decryption to evaluate the performance. The experiments are tested on a virtual machine with 3.47 GHz CPU and 8 GB memory. Figures 1 and 2 show the experiment results. As we can see, encryption time and decryption time grow linearly over the attribute number in all three schemes. The composite order scheme is undoubtedly the most timeconsuming scheme; its performance is almost unacceptable for practical applications. The reason for this poor performance is that all arithmetic and pairing operations are executed in a group much larger than those for the other two schemes. As for the prime order simulation scheme, it takes little time to get the deniability feature from the Waters scheme and therefore, the prime order simulation scheme is suitable to be distributed to cloud storage services for the deniability feature.

CONCLUSIONS

In this , we proposed a deniable CP-AB ENCRYPTION scheme to build an audit-free cloud storage service. The deniability feature makes coercion invalid, and the AB ENCRYPTION property ensures secure cloud data sharing with a fine-grained access control mechanism. Our proposed scheme provides a possible way to fight against immoral interference with the right of privacy. We hope more schemes can be created to protect cloud user privacy.

REFERENCES

[1] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in
Eurocrypt, 2005, pp. 457–473.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based
encryption for fine-grained access control of encrypted data,” in *ACM Conference on Computer and Communications Security*, 2006,
pp. 89–98.

[3] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy
attribute-based encryption,” in *IEEE Symposium on Security and
Privacy*, 2007, pp. 321–334.

[4] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive,
efficient, and provably secure realization,” in *Public Key Cryptography*, 2011, pp. 53–70.

[5] A. Sahai, H. Seyalioglu, and B. Waters, “Dynamic credentials and
ciphertext delegation for attribute-based encryption,” in *Crypto*,
2012, pp. 199–217.

[6] S. Hohenberger and B. Waters, “Attribute-based encryption with
fast decryption,” in *Public Key Cryptography*, 2013, pp. 162–179.

[7] P. K. Tysowski and M. A. Hasan, “Hybrid attribute- and reencryption-
based key management for secure and scalable mobile
applications in clouds.” *IEEE T. Cloud Computing*, pp. 172–186,
2013.

[8] Wired. (2014) Spam suspect uses google docs; fbi happy. [Online].
Available: http://www.wired.com/2010/04/cloud-warrant/

[9] Wikipedia. (2014) Global surveillance disclosures (2013present).
[Online]. Available: http://en.wikipedia.org/wiki/Global surveillance disclosures (2013-present)

[10] ——. (2014) Edward snowden. [Online]. Available: http://en.
wikipedia.org/wiki/Edward Snowden

[11] ——. (2014) Lavabit. [Online]. Available: http://en.wikipedia.
org/wiki/Lavabit

[12] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable
encryption," in *Crypto*, 1997, pp. 90–104.

[13] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters,
"Fully secure functional encryption: Attribute-based encryption
and (hierarchical) inner product encryption," in *Eurocrypt*, 2010,
pp. 62–91.

[14] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert,
E. de Panafieu, and C. R`afols, "Attribute-based encryption
schemes with constant-size ciphertexts," *Theor. Comput. Sci.*, vol.
422, pp. 15–38, 2012.

[15] M. D¨urmuth and D. M. Freeman, "Deniable encryption with
negligible detection probability: An interactive construction," in
Eurocrypt, 2011, pp. 610–626.

[16] A. O'Neill, C. Peikert, and B. Waters, "Bi-deniable public-key
encryption," in *Crypto*, 2011, pp. 525–542.

[17] P. Gasti, G. Ateniese, and M. Blanton, "Deniable cloud storage:
sharing files via public-key deniability," in *WPES*, 2010, pp. 31–
42.

[18] M. Klonowski, P. Kubiak, and M. Kutylowski, "Practical deniable
encryption," in *SOFSEM*, 2008, pp. 599–609.

T. Nagaraju, received B Tech. [CSE] from JNTU-H. He is completed Master of Technology (Information Technology from JNTU-H) He is working as assistant professor in CMRIT Hyderabad. His research interest is Cloud computing ,Network security.

M.S.R.LakshmiReddy received Btech from Annauniversity. He is completed Master of Technology (Software engg from JNTU-H) He is working as Assistant professor in CMRIT Hyderabad. His research interest is Cloud computing.

S.Vidyullatha received BTech from JNTU ,Kakinada. She completed Master of Technology (CSE from JNTU-H).She is working as Assistant professor in CMRIT Hyderabad. Her research interest is Cloud computing.