Multi-Authority Attribute Base Encyrption Scheme for Bdrive

Marvin Petzolt*

TU Berlin
marvin.petzolt@protonmail.com

July 10, 2018

Abstract

TODO: Abstract

I. Introduction

Bdrive is a secure cloud storage where files get split up in smaller chunks that are saved speratly on different storage provider. To ensure end-to-end encryption a Bdrive client encrypts each of his chunks with the a one-time symmetric key that is then encrypted under his own public key. This encrypted key is called a file key and it is uploaded to the Bdrive server where it is stored securely.

Since each device of the same user has a different private-public key pair, the client is in charge of making the file keys available for the new client. This is done by downloading each file key for the receptive file, downloading the public key of the new client, decrypting the file key with his own private key, encrypting it again with the public key of the new device and finally, uploading the new file key to the Bdrive server.

The number of file keys that need to be maintained raises with the number of clients. In addition Bdrive allows to share files between different users. The formula XXXX describes the number of file keys Bdrive need to store for each shared files between u users, where each user u_i has d_i devices.

$$n = \sum_{i \leq u} d_i$$

Lets construct an example where the manager of a company wants to create a shared folder with all company employees. It is a medium sized company with 50 employees. At least haft of them have two Bdrive clients running. The manager wants to upload the 250 photos of the last company trip. We end up by computing $3/2 * 50 * 250 = 18750$ file keys and for every new file uploaded 75 new file keys need to be uploaded.

We reach the point where the classical public-key end-to-end encryption scheme does not scale anymore.

II. Related Work

Attribute Based Encryption (ABE), first introduced by Sahai and Waters [1], is a cryptographic encryption scheme which encryptes under attributed that describe a user, rather than common known keys. This enables the encryptor to craft a cypher text over chosen attributes that can only be decrypted by any entity that holds a super set of matching attributes. Further, it is possible to embed the access policy in a tree inside the cipher text, where each node is contains âĂIJANDâĂİ or âĂIJORâĂİ gates. If the attribute values are stored in the leafs, the decryptor can evaluate the tree in a post-order fashion and evaluate whether the root node yields âĂIJtrueâĂİ

fix encoding

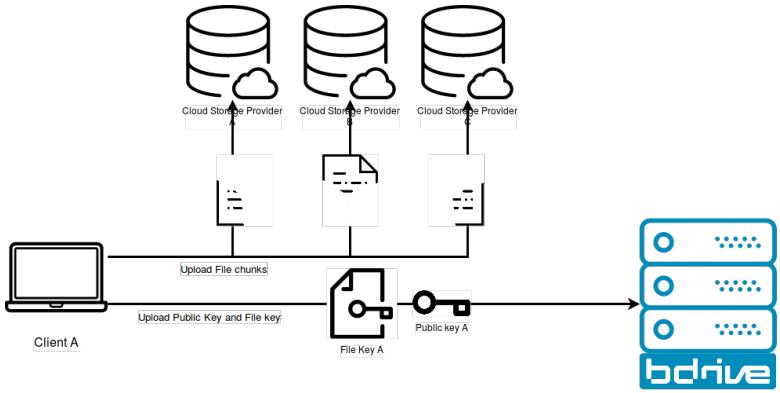

Figure 1: *caption here*

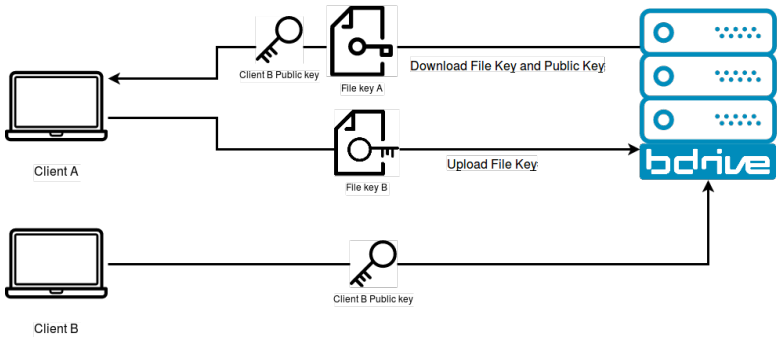

Figure 2: *caption here*

or âĂIJfalseâĂİ. This approach was first introduced by Bethencort, Sahai and Waters in Ciphertext-policy Attribute-Based Encryption (CP-ABE). [2]. It is also possible to do it the other way arround: Associate the userâĂŹs key with an access policy. Now, the encryptor needs only to encrypt the given plain text with the public key of specific attributes so that only user who hold the right keys are able to decrypt the cipher text. This approach is called Key-Policy Attribute Based Encryption cite(KP-ABE).

extend:

Multi-ABE

III. Security Requirements

The core security requirements regarding Bdrive in the context of a multi-authority ABE scheme are the following:

- **Collusion resistance:** For two users it should not be possible to combine their attributes to archive a higher level
- **Inter-Company Sharing:** Since Bdrive would need to consider a multi-authority ABE scheme, it should not be possible for a company to decrypt or issue files of other companies if no explicit exception is given for certain files by a trusted company relationship. A companies attribute authority (AA) should be responsible for its domain. In the case of an inter company relationship, attributes needs to be issued across different companies.
- **Central Authority:** In a multi-authority setting usually a central authority is coordinating the different attribute authorities to identify users and prevent collusion. However, it should not be possible that the central authority has a global decryption power.
- **Secret Master key (if any):** Key recovery requires a secret and securely stored mas-

ter key. It should solely function in the company domain and not globally.

- **Large Key Universe:** The key universe should be large so that Bdrive can act dynamically regarding attribution issuing and each company could define their own set of attributes. Further, a finite field of keys would restrict the number of possible users in the system, which is not intended.
- **Adding new Attribute Authorities:** It should be possible to add new attribute authorities while runtime. Without either shutting down the system or recreating each key.
- **Key and Attribute revocation:** Either a period defining the lifetime of valid keys or a direct revocation mechanism ensures key and attribute revocation. In both cases the keys have a limited lifetime of two months. After this period the keys become invalid and have to be reissued. Key attribute or whole key revocation should be possible to handle user management in terms of attribute promotion, attribute demotion and key revocation. Revoked keys are no longer able to decrypt the cypher text.

On the other hand we are forced to loosen some security requirements that have hold in a RSA based environment. One Attribute Authority (AA) needs to issue the attributes for a group of users. This users are assigned to a designated company. Since the AA issues also the private keys for each attribute it holds also a master secret for this company domain and is in turn able to decrypt all users file of this company. While in the current cryptographic scheme enrolled in Bdrive an artificial master-key was implemented, it would still be possible to easily remove it from the system. This is not possible with the an ABE scheme.

IV. Targets

The target of this work will be to implement and define a multi-authority ABE scheme for Bdrive that fits the security requirements of the previous section. This approach will decrease the number of file keys stored on the Bdrive server dramatically and will make Bdrive scale better for a higher user workload. To evaluate this a prototype should be implemented that is able to up- and download files, to encrypt them symmetrically and encrypt the file key under a multi-authority ABE scheme.

The minimal requirement is to implement a multi-authority ABE scheme where at least two AAs issue attributes to different users and the user are able to share a file between them. This file should only be encrypted using one file key. This implementation and system design summarized the security requirements of: Collusion resistance, inter-company Sharing, central authority, adding new authorities and Secret Master key. To implement and define this prototype ChaseâĂŹs proposal of an multi-authority ABE scheme [3] in combination with the extension of Chase to deescalate the decryption power of the central authority [4] is used.

The next step would be to implement the key-revocation, which is not trivial in an ABE setting. Optional is to extend the defined scheme with a large key universe. It is not clear yet whether this is even possible as it is proposed by Chase or if a complete different approach need to be taken to take this issue.

V. Concept

VI. Method of Evalution

References

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, 2005.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pp. 321–334, IEEE, 2007.

[3] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography Conference*, pp. 515–534, Springer, 2007.

[4] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 121–130, ACM, 2009.