INTRODUCTION

- Cloud based systems have revolutionized software and services.
- Certificate Authority:

Most solutions for protecting user privacy either obfuscate data from the cloud or rely on ABE performance overhead is not a concern.
- If you do not pay for a product, you are the product.
- Example solutions for personal information include emails, chat messages, pictures, personal files, documents.
- Cloud service providers offer free or subsidized services in exchange for user data.
- Examples of other personal information include emails, chat messages, pictures, personal files, documents.
- Our solution encompasses around letting users attach terms of service (ToS) to data before uploading it to the cloud environment.

This is achieved using Ciphertext Policy Attribute Based Encryption[11]

BACKGROUND

- Most solutions for protecting user privacy either obfuscate data from the cloud or rely on not using cloud services.
- Obfuscating data from the cloud implies that cloud cannot process data.
- Not relying on the cloud often implies setting up private servers which is expensive, difficult and requires skill.
- Our solution ensures that cloud services operate, but data owner dictates the Terms of Use.

SOLUTION

- Our proposed solution revolves around enabling the user to attach ToS to the data before uploading it to the cloud.
- Similar to ToS on websites, DVDs, Books etc.
- Policies are key value pairs combined with conjunction and disjunction
- Eg. shareWithThirdParty: true and allow_machine_learning: false
- Policies are legally binding, all violations support non repudiation
- Policy can be explicitly specified in the encryption tool or can be embedded into JPEG EX IF meta data.
- Policies are attached to the data using Attribute Based Encryption, the resulting Ciphertext is uploaded to the cloud environment.
- If cloud service provider does not comply with the policy, it never gets access to the plain text data.
- Infrastructure employs a Certificate Authority to ensure that Cloud Service Provider complies with policy.
- CA is similar to SSL CA, ensures CSP and users are who they claim they are.

ARCHITECTURE

- System consists of the following components:
  - Certificate Authority: Certifies authority ensures all parties are who they claim they are
    - In Step 1, CA shares its public key with Client & CSP
    - In Step 2, CSP authenticates itself presenting a set of policies
    - In Step 3, CA issues a K_{User} to CSP that can be used to decrypt data that complies with policy in Step 2
  - Client: ABE Encryptor app helps user encrypt data before uploading it to the cloud
  - Policy: Can be either embedded in JPEG meta data or be explicitly specified
  - Data is encrypted in Step 4 and plain text never leaves the client machine
  - Cloud Service Provider: CSP in our implementation was an image sharing service
  - Only Ciphertext is stored on the cloud environment
  - Each access request is decrypted using the K_{User} if the policy in the data matches the one in the key.

RESULTS

- ABE needs a reference standard and implementation
- Policies need to be standardized similar to software licenses like GNU GPL, Apache
- ABE performance overhead is not a concern
- PCD CA infrastructure needs to be established
- Cloud Service Providers should be incentivized to onboard before users.

REFERENCES