Chapter 7
Accountable Malicious Entity Detection Using Re-Encryption Mechanism to Share Data

S. T. Veena
Mepco Schlenk Engineering College, India

N. R. Somnath Babu
Mepco Schlenk Engineering Collge, India

P. Santosh
Mepco Schlenk Engineering Collge, India

ABSTRACT

Recently data sharing through the cloud is widely used and becoming a trend. But protecting data confidentiality and integrity is essential. Data confidentiality and integrity are maintained using cryptography. However, since it cannot ensure that the data hasn't been altered, there is still a trust issue. The proxy re-encryption (PRE) technique was presented as a solution to this. It includes encrypting the data twice, allowing one to determine whether or not it has been altered. The PRE system is also prone to attacks that forge re-encryption keys. Here the data is encrypted twice so that the data can be checked whether it is modified or not. But the PRE system is prone to the abuse of re-encryption keys. Therefore, accountable proxy re-encryption (APRE) is proposed. Here, if the data is altered by proxy, the system will detect whether proxy is malicious or the delegator is trying to frame proxy as malicious. Also, the authors extend the algorithm and implemented base64 encoding and decoding. This prevents many passive cyber attacks.

DOI: 10.4018/979-8-3693-1431-9.ch007

1. INTRODUCTION

IN today's digital world, Data is a critical resource, and it is increasingly becoming more important in all aspects of our lives.Data sharing is also crucial in today's interconnected world. By sharing data, we can collaborate and work together to solve complex problems and make better decisions. Cloud storage is the greatest choice any time a user wishes to share a data.The confidentiality of the data is the fundamental worry while sharing data on the cloud. The confidentiality and integrity of the data cannot be ensured by the content owner. Encryption of data is a crucial fix for this problem. For this reason, a variety of encryption techniques are available. The data is encrypted by the content owner (or delegator), who also distributes the decryption key to data consumers. For this to work, the content owner must be connected online at all times. To tackle this issue (Blaze et al., 1998) proposed Proxy Re-Encryption (PRE). Applying PRE in cloud, initially the delegator, proxy, and delegatee share all their public key certificates for authentication. The data is encrypted by the content owner (or delegator) and uploaded to the cloud. Using a re-encryption key, the proxy re-encrypts the ciphertext. The shared data is first encrypted, and then the ciphertext is sent to the proxy. The content owner produces a re-encryption key in response to a content receiver's request and provides it to the proxy, who then re-encrypts the ciphertext and forwards it to the receiver.

But the proxy and content owner together has the ability to find the decryption key and can abuse the data. Also the data owner can frame a proxy as malicious.

So to ensure data security and to check if the delegator or proxy is malicious, (Guo et al., 2021) proposed Accountable Proxy Re-encryption(APRE). Here there will be a court module. Anyone can approach the court to verify it. The proxy with a delegatee can modify the data or the delegator can accuse the proxy is accountable. To identify it, the court module is used. The court module uses a judge algorithm. The algorithm gets the initial ciphertext and then it asks the proxy and the delegator to decrypt it. By observing the plaintext the court module will find whether the proxy or delegator is accountable.

The APRE scheme does not capture all the attacks. There will be many attacks over network, which will lead to Chosen Plain Text (CPA)/Chosen Cyber CCA. This can result in serious loss to the data owner. Hence, to extend the security further, our system used base64 encoding and decoding.

2. RELATED WORKS

Other than proxy re-encryption techniques (Sun et al., 2018) presented a revocable identity-based encryption scheme with cloud-aided ciphertext evolution. In order