Abstract:
Nowadays, more and more users store their important data in the cloud. To ensure the security of the remotely stored data, users need to encrypt important data. From the point of data security which has always been an important aspect of quality of service, cloud computing causes new challenging security threats. Identity-based proxy re-encryption schemes have been proposed to shift the burden of managing numerous files from the owner to a proxy server. The existing solutions suffer from several drawbacks. First, the access permission is determined by the central authority, which makes the scheme impractical. Second, they are insecure against collusion attacks. The existing solutions do not actually solve the motivating scenario, when the scheme is applicable for cloud computing. Hence, it remains an interesting and challenging research problem to design an identity-based data storage scheme which is secure against collusion attacks and supports intra-domain and inter-domain queries. In this paper, we propose an identity-based data storage scheme where both queries from the intra-domain and inter-domain are considered and collusion attacks can be resisted. To achieve security for data transfer in the cloud, we introduce the notion of conditional proxy re-encryption whereby only the cipher text satisfying one or more conditions set by user one can be transferred by proxy and decrypted by user two.

Keywords: - IDS, DAS, PKG, IBDDS etc

I. Introduction:
Cloud computing provides users with a convenient mechanism to manage their personal files with the notion called database-as-a-service (DAS). In DAS schemes, a user can outsource his encrypted files to untrusted proxy servers. Proxy servers can perform some functions on the outsourced cipher text without knowing anything about the original files. Unfortunately, this technique has not been employed extensively. The main reason lies in that users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party. After outsourcing the files to proxy servers, the user will remove them from his local machine. Therefore, how to guarantee the outsourced files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Furthermore, how to guarantee that an authorized user can query the outsourced files from proxy servers only maintains the outsourced cipher text. Consequently, research around these topics grows significantly. In this paper, we propose two identity-based secure distributed data storage (IBSDDS) schemes. Our schemes can capture the following properties: (1) The file owner can decide the access permission independently without the help of the private key generator (PKG); (2) For one query, a receiver can only access one file, instead of all files of the owner; (3) Our schemes are secure against the collusion attacks, namely even if the receiver can compromise the proxy servers, he cannot obtain the owner’s secret key. Although the first scheme is only secure against the chosen plaintext attacks (CPA), the second scheme is secure against the chosen cipher text attacks (CCA). To the best of our knowledge, it is the first IBSDDS schemes where an access permission is made by the owner for an exact file and collusion attacks can be protected in the standard model.

Cloud Computing
Cloud computing is the technology which is the combination of many other technologies such as the utility computing, autonomic computing, virtualization, service-oriented architecture and many. It has got features from all these varied technologies. The goal is to provide scalable, shared resources- software and hardware- and there by providing services over the network. The terminology ‘as a service’ is coupled to it and referred as providing something as a service over the network. It can be Software as a Service (SaaS), Platform as a Service (Paas), Infrastructure as a Service (IaaS) and many.
Based on the hosting of environment and the type of users the architecture can be of four types: 1. Public Cloud, 2. Private Cloud, 3. Hybrid Cloud, 4. Community Cloud. In a public cloud, Cloud services are hosted for public usage and anyone can have their data stored and services get done using this kind of cloud. Data security plays a major role here. A private cloud is the one where the data access and service usage restricted to single authority. A hybrid cloud is shared by a limited set of organizations and has the features of both private and public cloud. Community cloud is much like the private cloud but the data is shared among the same entities of the single organization.

**Traditional System**
Cloud computing provides users with a convenient mechanism to manage their personal files with the notion called database-as-a-service (DAS). In DAS schemes, a user can outsource his encrypted files to untrusted proxy servers. Proxy servers can perform some functions on the outsourced cipher texts without knowing anything about the original files. Unfortunately, this technique has not been employed extensively. The main reason lies in that users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party. After outsourcing the files to proxy servers, the user will remove them from his local machine. Therefore, how to guarantee the outsourced files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Furthermore, how to guarantee that an authorized user can query the outsourced files is another concern as the proxy server only maintains the outsourced cipher texts. Consequently, research around these topics grows significantly.

**II. Proposed System:**
In this paper, we propose two identity-based secure distributed data storage (IBSDDS) schemes in standard model where, for one query, the receiver can only access one of the owner’s files, instead of all files. In other words, an access permission (re-encryption key) is bound not only to the identity of the receiver but also the file. The access permission can be decided by the owner, instead of the trusted party (PKG). Furthermore, our schemes are secure against the collusion attacks.

**Proposed Work**
For more securely data transfer in cloud we introduce conditional proxy re-encryption. We form the security model of conditional proxy re-encryption. An efficient construction of conditional proxy re-encryption scheme offers several advantages over previous systems including chosen-cipher text security, uni directionality and collusion-resistance. This scheme has better overall efficiency in terms of both computation and communication cost and provides better security.

In cloud, instead of converting all cipher texts, user may only want the proxy to convert the cipher texts with a specific word, such as “Important”. This problem solved by introducing the notion of conditional proxy re-encryption in construction of secure cloud storage. The conditional proxy re-encryption means user1 sends some message to user 2 and he wants that the user2 access some important messages earlier as these messages are very important. So user1 for important messages set condition i.e. set \( w = “important” \) with whom to send and is the reply is needed for this message, in subject and encrypt message. Condition is set in the subject because user 1 wants only the subject of message is visible to the proxy and not the body of the message, so message is encrypted by user 1. This message re-encrypted by proxy server using re-encryption key which formed by user1 and user 2 secret key and conditional key provided by user1 and converts into the cipher text. User1 wants only user 2 read this cipher text that satisfying condition \( w = “important” \) rather than all other user 1’s cipher text. Also user1 and user2 do not want that proxy server able to read this condition set cipher text for security purpose. If User 2 is not receiving or replying for this message then proxy check for another condition from the set. This condition may be tell proxy to send this message to another user i.e. user 3 on behalf of user 2. So here proxy server have functionality of partial decryption, means in difficult circumstances proxy are able to or
having right to decrypt that message on behalf of user which is actually receiver of that message. At that time proxy decrypts message and re-encrypt that message with key of another user called user 3 and send that message to user 3. Proxy server are connected with storage server and key server for accessing multiple conditions which are provided by owner of the message which are actually stored on storage server and keys which are managed by key server. So proxy server having all the information needed for encryption, condition checking and data forwarding to whom and which data to be forward and what to do when difficult situation occurs. If condition is not set properly by user1, then proxy is not able to re-encrypt the message and not able to forward this message towards another user. At that time, proxy replies to the user 1 with error message. So under right condition proxy re-encrypts the message efficiently and able to forward that message. This will be the advantage of this scheme to achieve more security.

III. System Architecture

![Diagram](image)

Fig. 1: The Model of Identity-Based Secure Distributed Data Storage Scheme

| Setup. This algorithm takes as input a security parameter $\lambda$, and outputs a bilinear group $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_t$. Let $g, h, \eta, g$ be the generators of $\mathbb{G}$, $u_0 \in \mathbb{G}$ and $\mathcal{U} = (u_1, u_2, \ldots, u_\lambda)$ where $u_i \in \mathbb{Z}_p$ for $i = 1, 2, \ldots, \lambda$. It chooses $\alpha \in \mathbb{Z}_p$ and sets $g_1 = g^\alpha$ and $g_2 = g^{\alpha \eta}$. The public parameters are $(\ell, G, G_t, g, h, \eta, q, \mathcal{U})$ and the master secret key is $\eta^\alpha$.

| KeyGen. Let $ID$ denote an identity which is an $n$ bit string. $ID_i$ be the $i$th bit of $ID$ and $I$ be the set which consists of all the index $i$ with $ID_i = 1$. This algorithm takes as input the master secret key $\eta^\alpha$ and the user’s identity $ID_i$ and computes:

| $K_{ID,1} = \eta^{\alpha \cdot \sum_{j=1}^{\ell} u_j}$, $K_{ID,2} = g^{\alpha \cdot \sum_{j=1}^{\ell} u_j}$ and $K_{ID,3} = g^{\alpha \cdot \sum_{j=1}^{\ell} u_j}$.

| The secret key for the user is $SK_{ID} = (K_{ID,1}, K_{ID,2}, K_{ID,3})$. This secret key can be verified by $e(K_{ID,1}, g) = e(\eta, g) \cdot e((\sum_{j=1}^{\ell} u_j), K_{ID,2})$ and $e(K_{ID,2}, g) = e(g, K_{ID,3})$.

| Encryption. Suppose that there are $k$ message $\{M_1, M_2, \ldots, M_k\}$. To encrypt $M_i$, the owner O chooses $s_i \in \mathbb{Z}_p$ and computes:

| $C_{i,1} = M_i \cdot e(g_1, \eta)^{s_i}$, $C_{i,2} = g^{s_i}$ and $C_{i,3} = (\sum_{j=1}^{\ell} u_j)^{s_i}$

| for $i = 1, 2, \ldots, k$. The ciphertext for the message $M_i$ is $CT_i = (C_{i,1}, C_{i,2}, C_{i,3})$. The owner sends $\{CT_1, CT_2, \ldots, CT_k\}$ to the proxy servers. The proxy servers (PSs) validate the ciphertexts by checking:

| $e(\sum_{j=1}^{\ell} u_j, C_{i,2}) = e(C_{i,1}, g)$

| for $i = 1, 2, \ldots, k$. If the equation holds, the proxy servers store the ciphertext $CT_i = (C_{i,1}, C_{i,2}, C_{i,3})$ for the owner. Otherwise, the proxy servers reject the ciphertexts.

| Query. If a receiver $R$ with identity $ID’$ wants to access $CT_i$, he chooses $t \in \mathbb{Z}_p$ and computes $K_{ID’,1} = K_{ID,1}^{t \cdot \eta}$ and $\Gamma = g^t$. He sends $(ID’, K_{ID’,1}, K_{ID,3}, \Gamma)$ to the proxy server. Then, the proxy server redirect $(ID’, K_{ID’,1}, K_{ID,3}, \Gamma, C_{i,2})$ to the owner.

| Permission. The owner checks whether the receiver has been verified by the PKG by $e(K_{ID’,1}^{t \cdot \eta}, g) = e(\sum_{j=1}^{\ell} u_j, K_{ID,2}) \cdot e(h, \Gamma)$. If it holds, the owner chooses $\beta \in \mathbb{Z}_p$ and computes:

| $D_1 = \frac{K_{ID,1}^{t \cdot \eta}}{K_{ID’,1}^{t \cdot \eta}} = (\sum_{j=1}^{\ell} u_j)^{t \cdot \eta}$, $D_2 = e(C_{i,3}, (\sum_{j=1}^{\ell} u_j)^{t \cdot \eta})$ and $D_3 = g^{t \cdot \eta}$.

| The owner sends $(D_1, D_2, D_3, K_{ID,2})$ to the proxy server.

| Re-encryption. Receiving $(D_1, D_2, D_3, K_{ID,2})$ from the owner, the proxy server computes the re-encrypted ciphertext as $C’_{i,1} = D_1 \cdot C_{i,1}$. $C’_{i,2} = C_{i,2}$. $C’_{i,3}$ = $C_{i,3}$. $C’_{i,4}$ = $D_1$. $C’_{i,5}$ = $D_3$ and $C’_{i,6}$ = $K_{ID,2}$.

| The proxy server responds the receiver with $CT’_i = (C’_{i,1}, C’_{i,2}, C’_{i,3}, C’_{i,4}, C’_{i,5}, C’_{i,6})$.

| Decryption. 1) To decrypt the ciphertext $CT_i = (C_{i,1}, C_{i,2}, C_{i,3})$, the owner O computes $M_i = C_{i,1} \cdot \frac{e(K_{ID,1}, C_{i,2})}{e(K_{ID’,1}, C_{i,2})}$.

| 2) To decrypt the ciphertext $CT’_i = (C’_{i,1}, C’_{i,2}, C’_{i,3}, C’_{i,4}, C’_{i,5}, C’_{i,6})$, the receiver R computes $K_{i} = K_{ID’,1}^{t \cdot \eta}$.

| $M_i = \frac{e(C’_{i,3}, C’_{i,5})}{e(K_{i}, C’_{i,5})}$.  

Fig. 2: IBSDDS I: Identity-Based Secure Distributed Data Storage Scheme
IV. Conclusion:
Cloud computing is a distributed system where users in different domains can share data among each other. Identity-based proxy re-encryption schemes have been proposed to outsource sensitive data from the owner to an external party. Nevertheless, they cannot be employed in cloud computing. As security of data storage is important, also the security of data transfer is important. We achieve the security of data transfer by introducing the conditional proxy re-encryption. It provides many advantages like chosen cipher text attack, unidirectionality and collusion-resistance over the previous schemes. This scheme provides secure model of cloud storage with safe data forwarding.
For example, they can only support the intra-domain query and the access key is computed with the help of the private key generator (PKG). Additionally, the proxy server must be trusted. In this paper, we proposed an identity-based data storage scheme which is suitable to the cloud computing scenario as it supports both intra-domain and inter-domain queries. In our scheme, the access key is bound to not only the requester’s identity but also the requested cipher text, and can be computed by the owner independently without the help of the PKG. For one query, the requester can only access one file of the owner, instead of all files. Furthermore, our scheme is secure against the collusion attacks. We proved the security of the proposed scheme in the selective-identity model.

References