Novel Approach for improving Security and Confidentiality in Public Clouds using Certificateless Encryption

Chudaman Devidasrao
Sukte
Dept. of Computer Science & I.T.,
Dr. B. A. M. University,
Aurangabad, India

Emmanuel M., PhD
Department of IT
P.I.C.T, Pune, India

Ratnadeep R. Deshmukh, PhD
Dept. of Computer Science & I.T,
Dr. B. A. M. University,
Aurangabad, India

ABSTRACT

In order to assure confidentiality of sensitive information stored in public clouds, a commonly embrace approach is to encrypt the information before uploading it to the cloud. Since the cloud does not know the keys used to encrypt the information, the confidentiality of the information from the cloud is assured. However, as many organizations are required to enforce fine-grained access control to the information, the encrypter mechanism should also be able to provide fine-grained encryption-based access control. Proposed work solves the key escrow problem and revocation problem of the previous systems and performs only a single encryption of each data item and minimize the overhead at the data owner.

Keywords

Cloud Computing, Public key cryptography, Certificate less, Data sharing.

1. INTRODUCTION

Cloud Computing is used in many small, medium and large sized companies and as many cloud users need the services of cloud computing, the major concern is the protection and security of their data in the cloud. Protecting data is always of importance and because of the crucial nature of cloud computing and the large amounts of complex data it support, the need is even more important. Henceforth, concerns regarding data privacy and security are showing to be a barrier to the wider uptake of cloud computing services.

Cloud is an emblematic word for the internet. So cloud computing refers to a type of computing which is based on the internet. Cloud computing [1] is concerned with applications and services that run on a distributed network using resources that are virtualized and accessed through common Internet protocols and networking standards. It creates an illusion that resources are limitless and that details of the physical systems on which software runs are kept hidden from the user. Cloud computing can be defined as a computing environment where computing needs by one party can be outsourced to another party and internet is used to access the computing power and resources like a database.

Cloud computing is a model [2] for enabling everywhere, appropriate, on-demand network access to the computing resources that can be quickly provided and released with minimum effort or service provider interaction. Cloud model consists of three service models that is Software as a Service, Platform as a Service and Infrastructure as a Service.

Security [5,6] within cloud computing is an particularly intrusive issue because the devices used to provide services do not belong to the user’s premises. The users do not have control or knowledge, what could happen to their data. This is a great concern in cases when users have important and personal information stored in a cloud computing service. Cloud computing service facilitator must ensure that the user’s information is safe because users will not compromise their privacy. This, however, is becoming increasingly challenging because as security developments are made, there always seems to be someone to figure out a way to breach the security and take undue advantage of user information.

Now a day, use of cloud has become very common. Due to benefits of public cloud storage, organizations have been adopting public cloud facility such as Microsoft Skydrive and Dropbox to manage their information. However, for the adoption of cloud storage services, the public cloud storage model should solve the critical issue of data confidentiality. That is, shared sensitive information must be strongly secured from unauthorized accesses. In order to ensure confidentiality of sensitive information stored in public clouds, a commonly adopted approach is to encrypt the data before uploading it to the cloud. Since cloud does not know the keys used to encrypt [7] the data confidentiality of data from cloud is assured. So, it is necessary to build the secure encryption system for sharing of data [3, 4, 8, 12] among public cloud. Existing system in the literature making use of symmetric key based mechanism, it has the problem of high costs for key management. Traditional public key cryptosystem [13] requires a trusted certificate authority to issue digital certificates that bind users to their public keys, and hence the overall certificate management is very expensive and complex.

2. METHODOLOGY

Certificate less [10, 11] public key cryptography (CL-PKC), a model for the use of public key cryptography which avoids the inherent escrow of identity-based [9] cryptography and yet which does not require certificates to guarantee the authenticity of public keys. The lack of certificates and the presence of an adversary who has access to a master key necessitate the careful development of a new security model.

A CL-PKC enabled systems [14] still makes use of TTP which is name as the key generating centre (KGC). By way of contrast to the PKG in ID-PKC, this KGC does not have access to entities’ private keys. Instead, the KGC supplies an entity A with a partial private key D_A which the KGC computes from an identifier ID_A for the entity and a master key. Note that we will often equate A with its identifier ID_A. The process of supplying partial private keys should take place confidentially and authentically: the KGC must ensure that the partial private keys are delivered securely to the correct entities. Identifiers can
be arbitrary strings. The entity A then combines its partial private key $D_A$ with some secret information to generate its actual private key $S_A$. In this way, A’s private key is not available to the KGC. The entity A also combines its secret information with the KGC’s public parameters to compute its public key $P_A$. Note that A need not be in possession of $S_A$ before generating $P_A$; all that is needed to generate both is the same secret information. The system is not identity-based, because the public key is no longer computable from an identity (or identifier) alone. Entity A’s public key might be made available to other entities by transmitting it along with messages (for example, in a signing application) or by placing it in a public directory (this would be more appropriate for an encryption setting). But no further security is applied to the protection of A’s public key. In particular, there is no certificate for A’s key. To encrypt a message to A or verify a signature from A, entity B makes use of $P_A$ and $ID_A$.

The cloud is trusted to perform the security mediation service and key generation correctly, but it is not trusted for the confidentiality of the content and key escrowing. Attribute Based Encryption (ABE) has been proposed that allows one to have most of the key generation and management functionality deployed in the untrusted cloud as our mCIL-PKE scheme does not have the problem of key escrowing and thus the KGC is unable to learn the full private keys of users.

The proposed system architecture of mediated certificateless encryption system is shown in Figure 1. The architecture consists of three entities: data owner, cloud, and users. The data owner possesses sensitive content that it wants to share with authorized users by storing it in the public cloud and partially decrypting encrypted data when users request the data. The cloud consists of three main services: an encrypted content storage; a key generation center (KGC), which generates public/private key pairs for each user; and a security mediation server (SEM), which acts as a security mediator for each data request and partially decrypts encrypted data for authorized users.

**Figure 1. System Architecture**

### 2.1 Proposed Algorithm

The algorithm for basic mediated certificateless public key encryption scheme is given as follows. It is made up of seven steps described below.

**Step 1:** Set-Up:

1. KGC takes as input a security parameter $k$ to generate two prime numbers $p$ and $q$. It then performs the following steps:
   1. A generator $g$ of $Z_p^*$ with order $q$ is picked.
   2. Select $x \in Z_q^*$ uniformly at random and using that value of $x$ calculate $y = g^x$.
   3. From the following cryptographic hash functions $H_1 : \{0,1\}^* \times Z_p^* \rightarrow Z_q^*$; $H_2 : \{0,1\}^* \times Z_p^* \rightarrow Z_q^*$; $H_3 : Z_p^* \rightarrow \{0,1\}_{n+k}$; $H_4 : Z_p^* \rightarrow \{0,1\}_{n+k}$; $H_5 : Z_q^* \rightarrow \{0,1\}_{n+k}$.

**Input:** Security parameter $k$.

**Output:** System Parameter and Secret Master Key.

**Step 2:** Set-Private-Key

- The entity A sets its private key by selecting $z_A \in Z_q^*$ uniformly at random.

**Input:** System Parameter and ID.

**Output:** User Private Key.

**Step 3:** Set-Public-Key

- The entity A computes its public key by $U_A = g^{z_A}$.

**Input:** User Private Key.

**Output:** User Public Key.

**Step 4:** SEM-Key-Extract

- KGC picks $s_0, s_1 \in Z_q^*$ and calculates $w_0 = g^{s_0}, w_1 = g^{s_1}, d_0 = s_0 + xH_1(ID_A, w_0), d_1 = s_1 + xH_2(ID_A, w_0, w_1)$. SEM-key for A is assigned as $d_0$ by KGC. Finally KGC sets $(U_A, w_0, w_1, d_1)$.

**Input:** User Registration to KGC and System Parameter, Master key and User Identity.

**Output:** SEM Key to Each ID.

**Step 5:** Encrypt

It executes the following steps to perform encryption for a plaintext $M \in \{0,1\}^n$ for the entity A with identity $ID_A$ and public keys $(U_A, w_0, w_1, d_1)$:

1. First it check’s whether $g^d = w_1 y^{H_A(ID_A, w_0, w_1)}$. If the consequent result is not valid, encryption algorithm must be aborted.
2. Calculate $r \in \{0,1\}_{n+k}$ and compute $r = H_3(M, \sigma, d_0, U_A)$.
3. Calculate $C_1 = g^{r'}$.
4. Calculate $C_2 = (M \parallel \sigma) \oplus H_4(U_A) \oplus H_5(w_0^{r'} y^{H_A(ID_A, w_0, w_1)})$.
5. Calculate $C_3 = H_d(U_A, (M \parallel \sigma) \oplus H_4(U_A), C_1, C_2)$.

Final ciphertext is $C = (C_1, C_2, C_3)$. 

---

**International Journal of Computer Applications (0975 – 8887)**

**International Conference on Cognitive Knowledge Engineering 2016**
The proposed algorithm is able to provide secure key generation in Certificateless cryptography. The basic approach in Certificateless cryptography allows the user to generate a secret key which involves both the KGC’s partial key, and user’s random secret value. The proposed algorithm is implemented using three different hash functions. Below are the tables showing the different values obtained when the algorithm is implemented using different hash function. Each of these tables show the time taken by algorithm when implemented with different hash function. Below table 1 consists of four entries showing the encryption time for both basic and improved approach and also the decryption time for both the schemes for different file sizes. Here six different file sizes are taken all in KB for getting encryption and decryption results when SHA-1 is used as hashing function.

**TABLE 1. COMPARISON TABLE USING SHA-1 HASH FUNCTION**

<table>
<thead>
<tr>
<th>File Size (in KB)</th>
<th>Basic_enc_time (in sec)</th>
<th>Basic_dec_time (in sec)</th>
<th>Improv_enc_time (in sec)</th>
<th>Improv_dec_time (in sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.46</td>
<td>7</td>
<td>1.7</td>
<td>3</td>
<td>1.6</td>
</tr>
<tr>
<td>2.27</td>
<td>8.6</td>
<td>2.5</td>
<td>3.9</td>
<td>2.4</td>
</tr>
<tr>
<td>4.06</td>
<td>15.3</td>
<td>4.7</td>
<td>7</td>
<td>4.1</td>
</tr>
<tr>
<td>5.64</td>
<td>21</td>
<td>8.5</td>
<td>11</td>
<td>7.2</td>
</tr>
<tr>
<td>7.24</td>
<td>31</td>
<td>9.4</td>
<td>15</td>
<td>8.5</td>
</tr>
<tr>
<td>9.32</td>
<td>37</td>
<td>13.5</td>
<td>17</td>
<td>11.5</td>
</tr>
</tbody>
</table>

From above table 1 it can be observed that the time taken for decryption is half of the encryption. And also the improved scheme takes less time than the basic approach for different file sizes. Below table 2 gives the data when SHA-256 is used, instead of SHA-1. The purpose of using SHA-256 is quite clear as it is considered to be secure than SHA-1 and after 2010, the security policies have made it clear to use SHA-256 over SHA-1, because of its unbrokenness till now. After one or two year may be 2017, browser like chrome will not be supporting SHA-1 so all the system implemented with SHA-1 needs to update accordingly in order to persist with the policies.

**TABLE 2. COMPARISON TABLE USING SHA-256 HASH FUNCTION**

<table>
<thead>
<tr>
<th>File Size (in KB)</th>
<th>Basic_enc_time (in sec)</th>
<th>Basic_dec_time (in sec)</th>
<th>Improv_enc_time (in sec)</th>
<th>Improv_dec_time (in sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.46</td>
<td>2.3</td>
<td>1.7</td>
<td>2.1</td>
<td>1.6</td>
</tr>
<tr>
<td>2.27</td>
<td>3.5</td>
<td>2.4</td>
<td>3.2</td>
<td>2.2</td>
</tr>
<tr>
<td>4.06</td>
<td>7.8</td>
<td>4.3</td>
<td>7.2</td>
<td>4.1</td>
</tr>
<tr>
<td>5.64</td>
<td>11.5</td>
<td>5.9</td>
<td>10.5</td>
<td>5.1</td>
</tr>
<tr>
<td>7.24</td>
<td>21.8</td>
<td>7.4</td>
<td>13.7</td>
<td>6.8</td>
</tr>
<tr>
<td>9.32</td>
<td>30.1</td>
<td>9.6</td>
<td>27.8</td>
<td>8.6</td>
</tr>
</tbody>
</table>

3. EXPERIMENTAL SETUP

We run our experiments on following system configurations:

Certificateless cryptography is a variant of identity based cryptography intended to prevent the key escrow problem. Ordinarily, keys are generated by certification authority or a key generation center that is given complete power and is implicitly trusted. To prevent the complete breakdown of the system in the case of a compromised KGC, the key generation process is split between the KGC and the user. The KGC first generates a key pair, where the private key is now the partial private key of the system. The remainder of the key is a random value generated by the user, and is never revealed to anyone, not even the KGC. All cryptographic operation by the user are performed by using a complete private key which involves both the KGC’s partial key, and user’s random secret value.
From the above table 2, it is quite clear that SHA-256 performs almost similar to SHA-1 despite having more number of iterations and complication, this performs even better usually if proper hardware resources and computing ability is met.

Below Table 3 shows the values using MD5 algorithm. The reason for picking this algorithm for showing the result is that this is by far considered as the fastest algorithm and this will be clearer by the table below as the entries in it depict the same. But when the security is taken into consideration MD5 is taken as the best algorithm. So, despite having the better result we will consider SHA-1 or SHA-256 over MD5.

TABLE 3. COMPARISION TABLE USING MD5 HASH FUNCTION

<table>
<thead>
<tr>
<th>File Size (in KB)</th>
<th>Basic_enc time (in sec)</th>
<th>Basic_dec time (in sec)</th>
<th>Improv_enc time (in sec)</th>
<th>Improv_dec time (in sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.46</td>
<td>2.4</td>
<td>1.5</td>
<td>2.2</td>
<td>1.3</td>
</tr>
<tr>
<td>2.27</td>
<td>3.4</td>
<td>2.1</td>
<td>3.1</td>
<td>2.0</td>
</tr>
<tr>
<td>4.06</td>
<td>7.2</td>
<td>3.6</td>
<td>6.8</td>
<td>3.3</td>
</tr>
<tr>
<td>5.64</td>
<td>10.6</td>
<td>5.2</td>
<td>10</td>
<td>4.7</td>
</tr>
<tr>
<td>7.24</td>
<td>14.8</td>
<td>6.6</td>
<td>13.8</td>
<td>6.3</td>
</tr>
<tr>
<td>9.32</td>
<td>19.7</td>
<td>8.4</td>
<td>18.1</td>
<td>7.8</td>
</tr>
</tbody>
</table>

From all the three tables above, it is quite clear that the improved scheme performs better than the basic scheme. For the hash function, we use SHA1 as the elementary operation. The basic idea of the hash function construction is as follows. Based on the field of the hash function output, we break the input into multiple blocks, and the block number is dynamically adjusted. For each block, we execute SHA1, convert the output into decimal numbers, say \(a_1, a_2, \ldots, a_n\), if the output field is \(Z^*_{\alpha q}\), then we compute \((a_1 \parallel a_2 \parallel a_3 \parallel \ldots \parallel a_n) \mod q\), where \(\parallel\) denotes the concatenation operation, to get the final result of the hash function.

Figure 2. shows the time required to perform the encryption operation in both the schemes i.e. in basic mCL-PKE scheme and improved mCL-PKE scheme for different message sizes. It can be seen from the graph, the encryption time increases linearly as the message size increases. Also it is clear that the time required for encryption using improved scheme is lesser than the basic approach. In terms of percentage it is decreased by 50%. As the bit length of \(q\) increases, the cost increases non-linearly since the encryption algorithm performs exponentiation operations. The time complexity of the encryption algorithm is \(O(n)\) where \(n\) is the number of user, as the user increases the time needed also increases. Space complexity of basic scheme is \(O(ns)\) where \(s\) is the size of the file and the number of user and improved scheme is \(O(n)\) since only one time encryption is needed.

In the proposed work improved scheme was implemented, where the data owner performs only one encryption per data item and creates a set of intermediate keys that allows authorized users to decrypt the data. Figure 3. compare the time to decrypt the data for both the basic and improved approach. It can be seen from the graph as more users are allowed to access the same data item, the better the improved scheme performs compared to the basic scheme. The improved scheme takes 2% less time as compared to basic scheme.

As the SHA-1 is constructed, in the same way SHA-256 is used with this system. Figure 4. and figure 5. shows the comparison of basic and improved scheme on both the parameter that is encryption and decryption time. SHA-256 is secure, so by using this the main feature of the system remains intact that is improved scheme performs better over the basic one.

MD5 takes on average 35% less time than SHA-1 and SHA-256 for encryption and 40% less time for decryption which is definitely faster compare to both. But from security point of view SHA-256 is preferred and used.
4. CONCLUSION AND FURTHER WORK

For secure data sharing in public cloud the proposed approach is called Mediated Certificateless encryption. This approach not only increases the level of security but also removes the drawback like key escrow problem. The proposed approach supports immediate revocation and assures the confidentiality of the data stored in an untrusted public cloud while enforcing the access control policies of the data owner. If the basic mCL-PKE scheme is applied directly to cloud computing and if many users are authorized to access the same data, because in real time the number of users will be high so the encryption costs at the data owner can become quite high. In such case, the data owner has to encrypt the same data encryption key multiple times, once for each user, using the user’s public keys. This drawback was eliminated in the proposed approach.

5. REFERENCES