An efficient IC-Lock Self-reader Data Security in Cloud Computing

Min-Shiang Hwang\textsuperscript{1,2}, Cheng-Chi Lee\textsuperscript{3,*} and Pei-Shan Chung\textsuperscript{4}

\textsuperscript{1} Department of Computer Science and Information Engineering, Asia University, 500 Liufeng Road, Wufeng, Taichung, Taiwan 402, R.O.C.
\textsuperscript{2} Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan, R.O.C.
\textsuperscript{3} Department of Library and Information Science, Fu Jen Catholic University, No. 510 Zhongzheng Road, Xinzhuang, New Taipei City 24205, Taiwan, R.O.C.
\textsuperscript{4} Department of Management Information Systems, National Chung Hsing University, 250 Kuo Kuang Road, Taichung, Taiwan 402, R.O.C.

Received: 19 Nov. 2014, Revised: 19 Feb. 2015, Accepted: 20 Feb. 2015
Published online: 1 Jul. 2015

Abstract: Recently, the concept of cloud computing is appearing, and the cloud storage service becomes a main issue because of its convenience and low storage cost. The cloud provides storage space service that cloud subscriber (or can be also called cloud user) can upload his/her own data and store it on cloud. However, the cloud is untrustful or semi-trustful. The cloud subscriber doesn’t want the cloud provider to know the information about the data. Therefore, serial researches discuss and propose several schemes to secure the data storage. In 2012, Liao proposed an IC-LOCK approach for the self-reader, which can let cloud subscribers store their data on the cloud and download data by themselves. However, this approach will cost more computation time. Therefore, we present an efficient approach based on RSA signature scheme in this paper. The proposed approach does not only inherits the advantages of Liao’s approach, but also spends less computation cost.

Keywords: Cloud computing, data security, IC-LOCK, self-reader, RSA signature

1 Introduction

As the speed of the bandwidth increases, the requirement of hardware performance and computing capacity are also increased. Through ubiquitous Internet, the cloud computing based on the Internet is appearing. The cloud computing can provide services via the Internet for cloud subscribers, and the cloud subscribers (or can be also called cloud users) can satisfy their requirements by using the cloud computing’s ability \cite{1,2}. In 2010, Armbrust et al. \cite{3} defined the cloud computing as “both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services.” The subscribers can obtain the service which the cloud computing provides through the Internet.

The cloud service and application are various, and one of the cloud service and application is data storage \cite{4,5} where people can store their data from the local server to the cloud. And they can use their data which is stored on the cloud or share it with the other person who is authorized by the data owner. The cloud provides a scalable storage for the cloud subscriber, and the subscriber can use it to obtain the stored data anywhere via the Internet. As the cost of the cloud storage service becomes cheaper \cite{6} or even free at a limit size, people is willing to store data on the cloud, such as the Amazon’s Simple Storage Service (S3) \cite{7}, the Google Cloud Storage \cite{8}, and the Dropbox \cite{9} etc. And like Dropbox, people can upload their data to store and download it to use when he/she needs. Since people will store their data on the cloud, they are afraid that their data will be known by the cloud provider. They may encrypt their data before uploading to the cloud.

A lot of researches \cite{10,11,12,13,14,15,16,17,18,19} discuss how to protect the security of the data storage. Kamara and Lauter \cite{10} uses a secret key to encrypt the data and store it on the cloud. People can retrieve the

\* Corresponding author e-mail: cclee@mail.fju.edu.tw
encrypted data on the cloud. Because the data stored in the cloud is encrypted by the cloud user, the cloud provider can’t retrieve the original data from the encrypted data. This scheme can protect data integrity, data confidentiality, availability, etc. According to above researches, the cloud subscriber stores his/her data on the cloud but he/she doesn’t want the cloud provider to know the stored information. For understanding the data storage security, the following features [10, 12, 14, 15, 17, 18] are described and defined:

– Data confidentiality: the cloud subscriber stores the data on the cloud, and the cloud provider can’t obtain any information of the stored data.

– Data integrity: the cloud subscriber’s data stored on the cloud is the same as his/her original data; and data is not tempered.

– Data availability: the cloud subscriber’s data is downloaded from the cloud and people can recover to the original data and use it at anywhere.

– Data non-repudiation: the cloud subscriber can’t deny that the data stored in the cloud is not sent from him/her.

– Security of Private key: the cloud subscriber’s private key should be protected and nobody can obtain this key except the user.

Recently, Liao [20] proposed a related approach to satisfy the above features of data storage security. Moreover, his approach is based on ElGamal signature scheme [21, 22, 23]. However, we found out that the cost of computation in his approach is high because of more exponent computation. To remedy the disadvantage of the Liao’s approach, we further propose an efficient approach to reduce this computation cost. The proposed approach is based on the RSA signature [24, 25, 26, 27]. The proposed approach not only satisfy the same security features, but also reduces the computation cost. The organization of the paper is organized as follows. In Section 2, we review the Liao’s scheme briefly. In Section 3, we illustrate our proposed approach, and analyze the proposed approach in view of security analysis, key management, and performance analysis in Section 4. In Section 5, Our conclusions are given.

2 A Review of an IC-Lock Self Reader Approach

In this section, we will introduce the Liao’s approach [20]. Liao proposed an IC-lock approach (See Fig. 1), and the proposed approach consists of three flows: lock flow, integrity check flow, and unlock flow. And each flow contains a few modules as follows; the lock flow has three modules: Cryptographic Key Generator (CKG), Encryption/Decryption Module (EDM), and Lock Generating Module (LGM); the integrity-check flow has one module, Integrity Check Module (ICM); the unlock flow has three modules: CKG, EDMs, and Locksmith Modules LSM.

![Fig. 1: The IC-Lock approach.](image-url)
2.2.2 Lock Flow

A data-warrant \( w_d \) of the cloud subscriber can include the cloud subscriber’s identity, several keywords of the data and so on. The cloud subscriber executes the lock flow to generate a one-time-use secret random seed \( k_d \) and \( \gcd(k_d, p - 1) = 1 \). After that, \( CKG \) is executed to generate a secret key \( MK = H(w_d || k_d) \), and \( EDM \) is executed to encrypt \( D \) with the secret key \( MK \). Then, \( LGM \) is executed to compute \( r_d \) and \( s_d \), used to output a lock \( (r_d, s_d) \). Through the Internet, the ciphertext \( C \), data-warrant \( w_d \), and the lock \( (r_d, s_d) \) are sent to the cloud provider.

2.2.3 Integrity-Check Flow

This flow can verify the integrity of all received data. After receiving \( C \), \( w_d \), and \( (r_d, s_d) \), \( ICM \) is executed to verify the data integrity by the cloud provider. If the verifying process is successful, \( (C, w_d, r_d, s_d) \) are valid; otherwise, they are invalid.

2.2.4 Unlock Flow

If the cloud subscriber wants to retrieve the ciphertext, he/she will send a query message including a keyword to the cloud provider. The cloud provider can use this keyword to search data-warrant which contains the keyword. And then the provider returns \( C \), \( w_d \), and \( (r_d, s_d) \) to the cloud subscriber. When the cloud subscriber receive \( C \), \( w_d \), and \( (r_d, s_d) \), he/she can execute \( ICM \) to verify the data integrity. Before \( LSM \) is executed, the cloud subscriber executes unlock flow to open the lock and release the secret random seed. The right cloud subscriber obtains the correct private key to execute \( LSM \). \( LSM \) is executed to recover \( k_d \), and \( CKG \) is executed to rebuild the secret key \( MK \) by the recovered secret random seed, \( k_d \). Finally, \( EDM \) is executed to decrypt the ciphertext \( C \), and obtains the data. The details of the Liao’s approach is depicted in Fig. 2.

In this paper, we propose an efficient approach and show the comparison table in Section 4. Our approach can also achieve the security requirements and is more efficient than the Liao’s approach. We will describe the details in the following section.

### Table 1: Notation table.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Signification</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_d )</td>
<td>The data-warrant created by the cloud subscriber</td>
</tr>
<tr>
<td>( C )</td>
<td>The ciphertext of the data</td>
</tr>
<tr>
<td>( D )</td>
<td>The cloud subscriber’s data</td>
</tr>
<tr>
<td>( k_d )</td>
<td>A one-time-use random number</td>
</tr>
<tr>
<td>( p )</td>
<td>A large prime number based on the Nyberg-Rueppel signature scheme</td>
</tr>
<tr>
<td>( q )</td>
<td>A primitive number is based on the Nyberg-Rueppel signature scheme in ( GF(p) )</td>
</tr>
<tr>
<td>( g )</td>
<td>A primitive number in ( GF(P) )</td>
</tr>
<tr>
<td>( x )</td>
<td>The private key of the cloud subscriber</td>
</tr>
<tr>
<td>( y )</td>
<td>The public key of the cloud subscriber</td>
</tr>
<tr>
<td>( MK )</td>
<td>A secret key</td>
</tr>
<tr>
<td>( {r_d, s_d} )</td>
<td>The lock of data-warrant signed by the cloud subscriber</td>
</tr>
<tr>
<td>( E_{MK}(D) )</td>
<td>Using a symmetric cryptosystem [28, 29] to encrypt the data ( D ) with the secret key ( MK )</td>
</tr>
<tr>
<td>( H() )</td>
<td>A secure one-way hash function [30, 31]</td>
</tr>
<tr>
<td>( \gcd() )</td>
<td>Greatest common divisor function</td>
</tr>
<tr>
<td>( \parallel )</td>
<td>The concatenation operation</td>
</tr>
</tbody>
</table>

### Table 2: Module table.

<table>
<thead>
<tr>
<th>Modules</th>
<th>Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>( CKG )</td>
<td>A secret key generated</td>
</tr>
<tr>
<td>( EDM )</td>
<td>Data encrypted or the ciphertext decrypted</td>
</tr>
<tr>
<td>( LGM )</td>
<td>A lock generated</td>
</tr>
<tr>
<td>( ICM )</td>
<td>The data integrity (the lock and the ciphertext) verified by using public information</td>
</tr>
<tr>
<td>( LSM )</td>
<td>The lock opened and the secret random seed recovered</td>
</tr>
</tbody>
</table>

### 3 The Proposed Approach

In this section, we present our approach which is more efficient than the Liao’s approach. And our approach spends less computation overhead than the Liao’s approach. The proposed approach is based on RSA signature scheme and contains three flows: (1) lock flow, (2) integrity-check flow, and (3) unlock flow. The notations used in our approach are in Table 3 and the proposed approach is depicted in Fig. 3.
3.1 Notations

The notations of the proposed approach are listed in Table 3. Some notations are the same as Table 1. Here, we just list the notations different from Table 1.

3.2 Our Approach

3.2.1 Parameter Setup

Setup $a$ and $b$ as a large prime number (e.g., 1024 bits each). Compute $N = a \times b$ and $\phi(N) = (a-1)(b-1)$. Choose $e$ that is coprime to $\phi(N)$, and $e$ is less than $N$. Among these parameters, $e$ is the cloud subscriber's private key.

3.2.2 Lock Flow

A cloud subscriber creates a data-warrant which contains the cloud subscriber’s identity, several keywords of the data, and the timestamp generated by the cloud subscriber. First, the cloud subscriber executes CKG to compute $K = H(d || w_d)$ for generating a secret key $K$. And EDM is executed to encrypt the data $D$ with a secret key $K$ to output the ciphertext $C$. And then, LGM is executed to compute $S = H(C || w_d)^d \mod N$ with the cloud subscriber's private key $d$, which output the cloud subscriber’s signature $S$. $(C, S, w_d)$ is sent to the cloud provider. Finally, lock flow is finished.

3.2.3 Integrity-Check Flow

After receiving $(C, S, w_d)$, the cloud provider verifies the validity of the timestamp $T_S$ included in $w_d$ by comparing

---

Fig. 2: Liao’s approach.

Fig. 3: Our approach.

d is chosen to satisfy $e \times d \mod \phi(N) = 1$, where $d$ is the cloud subscriber’s private key; $(d, a, b)$ are kept secret, and $(e, N)$ are public.
### Table 3: Notation used in the proposed approach.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Signification</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>A large prime number</td>
</tr>
<tr>
<td>(b)</td>
<td>A large prime number</td>
</tr>
<tr>
<td>(\phi(N))</td>
<td>Euler’s totient function, it can count number of positive integers less than or equal to (N) that are coprime to (N).</td>
</tr>
<tr>
<td>(d)</td>
<td>The private key of the cloud subscriber</td>
</tr>
<tr>
<td>(e)</td>
<td>The public key of the cloud subscriber</td>
</tr>
<tr>
<td>(K)</td>
<td>A secret key</td>
</tr>
<tr>
<td>(S)</td>
<td>The digital signature signed by the cloud subscriber</td>
</tr>
<tr>
<td>(E_K(D))</td>
<td>Using a symmetric cryptosystem to encrypt the data (D) with the secret key (K)</td>
</tr>
<tr>
<td>(ID_S)</td>
<td>The cloud subscriber’s identity which is in the data-warrant</td>
</tr>
<tr>
<td>(Keyword)</td>
<td>Several keywords of the data in data-warrant</td>
</tr>
<tr>
<td>(T_S)</td>
<td>The cloud subscriber’s timestamp</td>
</tr>
<tr>
<td>(T^*)</td>
<td>The current timestamp of the system</td>
</tr>
<tr>
<td>(\Delta T)</td>
<td>The time threshold predefined by the system</td>
</tr>
</tbody>
</table>

\((T^* - T_S) \leq \Delta T\). Then ICM is executed to compute \(S^*\) by using the cloud subscriber’s public key \(e\), and to check whether \(S^*\) is equal to \(H(C \parallel w_d) \mod N\) or not. If the equation holds, \((C,S,w_d)\) are valid. Otherwise, they are invalid. Through this flow, the data integrity is verified by the cloud provider.

#### 3.2.4 Unlock Flow

The cloud subscriber will send a query message including a keyword to the cloud provider if he/she wants to retrieve the ciphertext. The cloud provider can search the data-warrant \(w_d\) by using this keyword. Then the cloud provider returns \((C,S,w_d)\) to the cloud subscriber. When the cloud subscriber receives the message, ICM is executed to verify if the equation holds or not by him/her. If \(S^*\) is equal to \(H(C \parallel w_d) \mod N\), \((C,S,w_d)\) are valid. Next, CKG is executed to recovery a secret key \(K\) by computing the equation \(H(d \parallel w_d)\) with the cloud subscriber’s private key. Finally, EDM is executed to decrypt the ciphertext \(C\) with a secret key \(K\) so to obtain the data \(D\).

### 4 Analysis of the Proposed Approach

In this section, we will analyze the proposed approach in view of security analysis, key management analysis, and performance analysis as follow.

#### 4.1 Security Analysis

We describes the security of our proposed approach with the above properties of data storage security: data confidentiality, data integrity, data availability, data non-repudiation, and security of private key. Through these properties, we can achieve data storage security. In addition, we add to the other feature in our approach.

**Theorem 1.** Our approach can achieve property of data confidentiality.

**Proof.** When the cloud subscriber wants to store his/her data on the cloud, he/she encrypts the data with a secret key \(K\) before uploading to the cloud. Then the encrypted data will be transferred from the subscriber to the cloud. A secret key \(K\) is generated with the cloud subscriber’s private key \(d\) by executing EDM: \(K = H(d \parallel w_d)\). Even if the attacker interrupts the transmitted message \((C,S,w_d)\), he cannot get information of the cloud subscriber’s private key \(d\). Therefore, the attacker cannot calculate a secret key \(K\) to decrypt this saved data.

**Theorem 2.** Our approach can achieve property of data integrity.

**Proof.** The data is encrypted with a secret key \(K\) before uploading to the cloud. The saved data on the cloud owned by the provider cannot be decrypted or tampered by the attackers because the attackers don’t have a secret key \(K\), and the transmitted message doesn’t include the information of a secret key \(K\). If they want to compute a secret key \(K\), they will get the cloud subscriber’s private key \(d\). However, this private key \(d\) is protected by the cloud subscriber. So the attacker and the cloud can’t obtain a secret key \(K\) so that the data integrity is guaranteed. In addition, the module, ICM can check the integrity of message by executing \(S^* = H(C \parallel w_d) \mod N\), so the attacker can’t interpret the transmitted message in order to tamper with it.

**Theorem 3.** Our approach can achieve property of data availability.

**Proof.** Through ICM and CKG, the cloud subscriber can check if the received message is right or not, and then he/she rebuilds the secret key \(K\) with his/her private key \(d\). And the cloud subscriber can decrypt the ciphertext \(C\)
of the data with the secret key $K$ to obtain the original data $D$. Therefore, the cloud subscriber can recover the original data. Since the secret key $K$ is generated by the private key $d$, the cloud subscriber only manages his/her private key used to obtain the decrypted data via the cloud at anywhere.

**Theorem 4.** Our approach can achieve property of data non-repudiation.

**Proof.** Data is encrypted by a secret key $K$, and the secret key is generated by the cloud subscriber’s private key $d$. Besides, $LGM$ computes $H(C \parallel w_d)^d \mod N$ with the cloud subscriber’s private key $d$ that is based on RSA signature. Only the cloud subscriber owns the private key $d$. There, the cloud subscriber can’t deny the encrypted data $C$ is not sent from him/her.

**Theorem 5.** Our approach can achieve property of security of private key.

**Proof.** In our approach, the attacker can’t obtain the private key from the transmitted message $(C, S, w_d)$ because the attacker can’t get any information about the cloud subscriber’s private key $d$ from these messages. Even the attacker knows the public key $e$, he/she also can’t obtain the cloud subscriber’s private key. The cloud subscriber’s private key of our approach is based on the security of the RSA algorithm. Therefore, the attacker can’t get $d$ from any manner.

**Theorem 6.** Our approach can achieve property of new division assurance.

**Proof.** In our approach, the timestamp $T_3$ is added to the data-warrant $w_d$. When the cloud subscriber downloads the saved data, he/she can know if the generated time of the data is new or not. Through this, the cloud subscriber can get the data which he/she wants.

### 4.2 Key Management

In addition, the proposed approach can achieve these features, and it is efficient because it just needs to protect one key, subscriber’s private key. In our approach, the secret key $K$ is derived from $CKG: K = H(d \parallel w_d)$ and $w_d$ is sent to the cloud provider. Besides, if the cloud subscriber wants to recover the secret key $K$, he just needs his private key $d$ and the data-warrant $w_d$ sent from cloud provider to compute by executing $CKG$. Thus, only the private key $d$ has ability to recover the secret key $K$ used to decrypt the encrypted data. And only the private key $d$ of the cloud subscriber needs to be kept. Therefore, the proposed approach can achieve efficient key management.

### 4.3 Performance Analysis

The proposed approach not only achieves the security of the data storage, but also reduces the cost of the computation. We compare the computation cost of our approach with the Liao’s approach. In Table 4, it can be found out that our approach only needs less modular exponential operations, compared to Liao’s approach. We don’t need the module $LSM$ for recovering the secret key $MK$ to compute $k_d$ in unlock flow. We can reduce two exponential operations because our $ICM$ module just needs one exponential operation and one hash operation to verify. This performance comparison is listed as follows.

<table>
<thead>
<tr>
<th>Item</th>
<th>Liao’s approach</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CKG$</td>
<td>$2T_h$</td>
<td>$2T_h$</td>
</tr>
<tr>
<td>$EDM$</td>
<td>$2T_{sym}$</td>
<td>$2T_{sym}$</td>
</tr>
<tr>
<td>$LGM$</td>
<td>$1T_{exp} + 1T_h$</td>
<td>$1T_{exp} + 1T_h$</td>
</tr>
<tr>
<td>$LSM$</td>
<td>$1T_h$</td>
<td>No</td>
</tr>
<tr>
<td>$ICM$</td>
<td>$(2T_{exp} \times 2)$</td>
<td>$((1T_{exp} + 1T_h) \times 2)$</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>$4T_h + 2T_{sym} + 5T_{exp}$</td>
<td>$5T_h + 2T_{sym} + 3T_{exp}$</td>
</tr>
</tbody>
</table>

$T_h$: the computation time of one-way hash function;
$T_{sym}$: the computation time of symmetric key operation;
$T_{exp}$: the computation time of modular exponential operation.

### 5 Conclusion

In this paper, the proposed approach not only achieves the data storage security properties: data confidentiality, data integral, data availability, data non-reputation, security of private key, and new division assurance, but also reduces the computation time. In addition, the proposed approach also retains the advantages of the Liao’s approach. The cloud subscriber just keeps their private key to recover a secret key of decrypted data to obtain the original data. The cloud subscriber only needs to manage one key, his private key, so he/she doesn’t need many keys to encrypt or decrypt the data. According to our analysis, the proposed approach is a secure and efficient approach for cloud computing.

### Acknowledgement

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

### References


Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China (ROC), in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field “Electronic Engineer” in 1988. He also passed the National Telecommunication Special Examination in field “Information Engineering”, qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999–2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002-2003. He is currently a professor of the department of Management Information System, National Chung Hsing University, Taiwan, ROC. He obtained 1997, 1998, 1999, 2000, and 2001 Outstanding Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang had published 170+ articles on the above research fields in international journals.

Cheng-Chi Lee received the Ph.D. degree in Computer Science from National Chung Hsing University (NCHU), Taiwan, in 2007. He is currently an Associate Professor with the Department of Library and Information Science at Fu Jen Catholic University. Dr. Lee is currently an editorial board member of International Journal of Network Security and Journal of Computer Science. He also served as a reviewer in many SCI-index journals, other journals, other conferences. His current research interests include data security, cryptography, network security, mobile communications and computing, wireless communications. Dr. Lee had published over 100+ articles on the above research fields in international journals.

Pei-Shan Chung received her B. M. in information Management from Chung Yuan Christian University, Jungli, Taiwan, ROC, in 2010. She is currently pursuing the M.S. degree with the Department of Management Information Systems from National Chung Hsing University. Her research interests include information security, cloud computing, and cryptography.