Computer Science and Information Systems 11(3):925–941 DOI:10.2298/CSIS130919057C

# A Secure Mobile DRM System Based on Cloud Architecture

Chin-Ling Chen[1], Woei-Jiunn Tsaur[2], Yu-Yi Chen[3] and Yao-Chung Chang[1]

[1]Department of Computer Science and Information Engineering
Chaoyang University of Technology,
Taichung, 41349, Taiwan
{clc@mail.cyut.edu.tw; cyc200@gmail.com}
[2]Department of Information Management
Da-Yeh University,
Changhua, 51591, Taiwan
wjtsaur@mail.dyu.edu.tw
[3]Department of Management Information systems
National Chung Hsing University,
Taichung, 402, Taiwan
chenyuyi@nchu.edu.tw

**Abstract.** Public cloud architecture offers a public access software service. Users can login to access the cloud resources via various devices. The main advantage of the SaaS (Software as a Service) cloud service is that it supports different software and devices, in order to open web browsers, to authenticate the users through the standard format. E-books are protected by digital rights management (DRM), and users can use mobile devices to read them. However, the users' identity need to be authenticated or the communication between the user and the cloud server will be at risk. The processes by which users submit their proof of identity to the cloud needs to be protected. In this paper, information security can be achieved efficiently via cloud server architecture and a cryptography mechanism. The proposed scheme focuses on using a mobile device to access the cloud service. The DRM mechanisms can protect digital content; once the mobile users pass the authentication they can access the cloud services, with authenticated users able to easily use mobile devices to read digital content.

**Keywords:** Cloud, DRM, Authentication, Mobile Devices, Security

## 1. Introduction

First, we introduce cloud architecture, the DRM concept of cloud architecture, and the analysis of DRM implementation using mobile devices.

### 1.1 Cloud Architecture

As long as information is stored in a cloud, users can access the cloud service through the Internet and mobile devices[1,2] anytime and anywhere. The user does not need to

know what kind of cloud architecture is present (such as cluster computing, grid computing, distribution computing, etc). The user need only send the request to the cloud and it will perform the most efficient operations.

The early goals of cloud architecture were to combine many computers of distributed computations via the Internet. The running program was divided into many threads and distributed into many computers for execution, with the result being presented immediately. Cloud architecture was gradually developed into service-oriented applications, with users being able to use the cloud properties: permanently available, fast computing, etc, with simple steps such that users could access the services provided by the cloud [3].

In the early stages, users communicated with different devices provided by the cloud architecture, and the cloud structure communication services needed to be robust. Current cloud structure has adopted a hierarchical structure. The top of the user services request message is forwarded and handled by the internal framework. Users do not directly communicate with the internal structure of clouds, and this ensures internal safety; this is called object-to-object architecture [3], and is distinct from the early host-to-host architecture. The present cloud structure can be divided into the following three modes [4,5]; the structure is shown in Figure 1.

- Public Cloud
- Private Cloud
- Hybrid Cloud

When the user's request message passes through the interface of a public cloud, malicious packets are filtered out by the firewall. The authenticated user's request will be forwarded to the API server. The API server need not be in the same geographical region. For example, when a Google Docs file is stored in a US database, the document can be opened by other people to co-edit it, and other users may edit the same document in different countries; however, all of them use the service through the same API.

On the other hand, a private cloud is an internal self-management systems database which develops and maintains the normal operation of the API. The network is connected through a local network; this incurs greater cost for small and medium enterprises. Thus, the Hybrid Cloud was developed. The Hybrid Cloud structure acts as a proxy server in most enterprises. Its main goal is to identify staff identification. In this way, it allows enterprises to control their own staff permissions, while the database is maintained by the provider. Costs are there by reduced.

### 1.2 Cloud Services Model

If cloud services are provided by a single industry, the cloud may not be able to satisfy all of a user's requirements. Thus, a user may use cloud network services provided by different industries. Cloud size can be divided into the following modes: Domestic clouds and Transborder clouds [6].

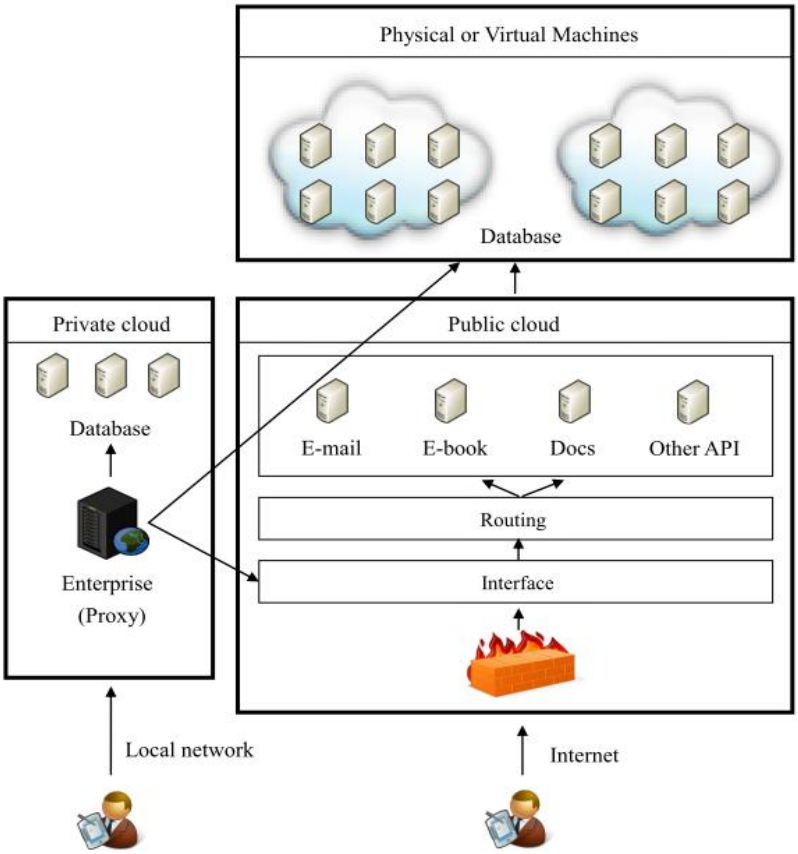


**Fig. 1.** Cloud architecture

(1) Domestic clouds: The entire cloud is physically located within one jurisdiction. The provider provides devices or data exclusively for specific enterprises. The provider need not provide additional service via third party provider to a specific provider, with the resultant advantages of uniform size and high data security.
(2) Transborder clouds: Devices can transmit data to a server (such as Google). Although the Google servers may be located in different countries, users can determine which server stores the data, even if they cannot find some data. Google Docs is a similar concept: someone can open a file and other people can edit the same document in different locations in different countries.

In February 2000, Amazon.com suffered from DDOS attacks which caused serious damage [7]. A new technology was developed to defend against such attacks. Now, packets will be filtered, and it will be determined whether they are normal or not by the firewall before users communicate with the cloud. The private key of the cloud system is not stored in the user's equipment. The user must use a secure encryption method (such as Public Key Infrastructure (PKI) or Secure Socket Layer (SSL) to transmit messages to the cloud, and then the cloud's stored user identity verification table will identify the user.

In 2007, vendors pushed the OpenID [8] verification specification 2.0 and attributes of a standard 1.0. OpenID, aiming to provide different cloud providers with a means to authenticate users' identities. The users only need to register once with OpenID, and they can then log into the authentication pages. However, OpenID alliance should ensure the users' safety and be able to determine if a cloud is illegal or not, otherwise users' privacy will be easily revealed by a malicious attacker masquerading as a cloud service.

Cloud services have been a hot topic in recent years. Despite the lack of a concrete definition of a cloud, there seems to be a common consensus as to what constitutes a cloud [6]. The National Institute of Standards and Technology (NIST) [9] has proposed the following five basic characteristics of current cloud architecture:

(1) On-demand self-service: A consumer can unilaterally provide computing capabilities, such as server time and network storage, as needed automatically, without requiring human interaction with each service's provider.

(2) Broad network access: Capabilities are available over the network and accesses through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops and PDAs).

(3) Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control over or knowledge regarding the exact location of the provided resources, but may be able to specify location at a higher level of abstraction (e.g., country, state or data center). Examples of resources include: storage, processing, memory, network bandwidth, and virtual machines.

(4) Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out, and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

(5) Measured Service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled and reported, providing transparency for both the provider and consumer of the utilized service.

As long as a service is connected to a network and uses the network to achieve a certain goal, it can be called a cloud service (for example: E-mail, E-books, Google Docs, Google TV, Cloud Printer, etc). That is, users rely on the application data stored on a remote server, with no additional devices installed in the personal applications. Users' data and information can be stored anywhere in the cloud.

A prerequisite of connecting to the cloud is network connectivity. Users can send a message, use the service, receive messages etc; all rely on a network connection to communicate with the cloud. The user's information must be protected so there must be communication through a mechanism to protect the user's identity and information. Different suppliers have different protection mechanisms, for example, Google protection mechanism is used for SSL.

The remainder of this paper is organized as follows: Section 2 reviews the DRM related work. In section 3, we introduces the proposed protocol. Section 4, we analyze the security of the proposed scheme, and we provide conclusions in Section 5.

## 2. The DRM Related Work

E-books are currently the main product of cloud services. At present, the main providers are companies like Google, Apple, and Microsoft, although the E-book format has yet to be standardized. However, the main specifications of E-books are DRM, DRM-Free, and Adobe PDF format.

Take, for example, DRM-Free with permission [10]; on April 2, 2007, Apple announced that half of the DRM-protected music on iTunes would be sold via DRM-Free. The price would be lower for higher music quality. In this way, DRM-protected MP3 digital products it needed to pay for the license; there was the limitation that only Apple-related products could share this benefit. On the other hand, Google's DRM-Free forbids users to copy or print the digital content [11].
The primary business objectives of DRM are:

- Providers must specify the user's rights
- Digital content cannot be tampered
- The print and copy permissions of the digital content need to be authorized
- Digital content's Copyright notice

The Provider sells DRM-protected digital content that can be used to control the consumer's rights [12, 13, 14, 15]. However, the DRM cloud provider authorizes the users to access the digital content via a one-time sale.

Microsoft for digital content protection [16] must install the RMS software at the user end, and is limited to Windows OSes. The encryption method is the RSA [17] key component. When a user requests to authorize the use of a right, it allows the designated user and is authorized to grant the permission. However, the SP2 version added an offline authorization function, and authors use the RMS application to create file permissions; this specifies the authorization conditions. This is a special license which can be granted by the offline state RMS-protected content permissions.

Our proposed architecture allows consumers to download E-books and enjoy the benefits of the trial period (e.g., DRM-Free). When users buy the products, the users' permission will be changed via License (such as DRM). However, our architecture is such that, via the Internet, it is possible at any time to record a user's E-book page number. The advantages of our approach are that it allows users to read E-books on different devices, and it can be easily modified to record the number of pages. It can also prevent purchased E-book users illegally forwarding documents to other users.

### 2.1 Discussion of Using a Mobile Device to Implement DRM

With the rapid development of smart mobile devices, it is now possible to easily access network resources. Even though it is well-known that mobile devices are undermined by several recent threats [18], these mobile devices (such as PDAs or Tablet PCs) and cloud services can be combined to form an easy to use communication platform.

Users can access the cloud services through different mobile devices, however, the hardware of such mobile devices is limited in the following ways [7, 19]:

- bandwidth limitations
- connection stability
- low computational ability
- limited battery capacity
- small storage capacity

From the mobile user's viewpoint, the user must provide his/her identification before using the cloud service. This is different from using a smart card, since not every device can read smart cards. Moreover, different operating systems have different peripheral limitations (such as iPad). Although users can browse the web, the device does not provide a general standard interface (such as USB) to provide the smart card reading function or other more secure mechanisms (such as a biometric identification mechanism).

On the other hand, the mobile device's computing power is limited. In order to send a protected message from these mobile devices it is necessary to consider other appropriate security mechanisms. Google or Apple, and other providers of these cloud services, do not provide a clear definition for the services model of the cloud. In this paper, we present a mobile device-based DRM system to achieve the following objectives:

(1) Provide a process for clearer communication enabling a unified authentication.
(2) Reduce the computation of communication for mobile devices.
(3) The suppliers can use their encryption method to protect the security of E-books.
(4) The E-book providers for DRM purpose of sale and limitation are not the same.

In order to achieve the required level of security, we have integrated the mobile devices and the cloud services model to allow users to access an E-book resource under secure authentication.

## 3. Proposed Authentication Protocol

Because Linux is outstanding for parallel computing and executing efficiency [4], the proposed cloud service for mobile DRM systems is based on Linux. Linux is open source, so users can develop various APIs to meet their requirements.

The user's message will first pass through the firewall to confirm whether or not the packets are normal. For authenticating users, the cloud server aims to produce the session key between user and cloud. The cloud confirms the identity of the user's mobile device. The user need not worry about the messages sent to the cloud end or the internal processing. Our proposed architecture is shown in Figure 2:

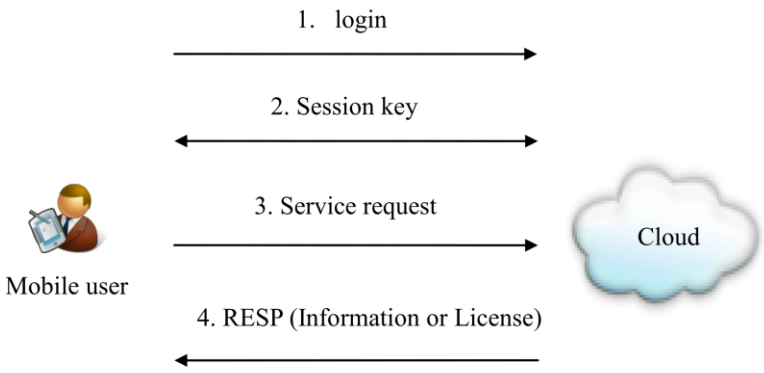


**Fig. 2.** Our proposed architecture

Step 1: User logs into cloud for authentication via mobile device.
Step 2: Cloud confirms the user's identification and generates the session key.
Step 3: The user's message is protected by the session key and a service request is made to the cloud.
Step 4: Cloud responds to user's request (such as the E-book pages or E-book usage rights).

Our scheme is to record the page number of the user's last review. We limit the user's communication time with the cloud to negotiate the session key by changing the license permission. The advantages are that users can read E-books on different devices, and we can prevent access to the E-books.

The following notation is used in this paper:

| | |
|---|---|
| $\oplus$ | exclusive -or operation |
| $\|\|$ : | concatenation operation |
| $ID$ : | user identification |
| $PW$ : | user password |
| $IMEI$ : | identity of the mobile device, International Mobile Equipment Identification |
| $N_u, N_s$ : | nonces |
| $SK$ : | session key between user and cloud |
| $E-book_{req}$ : | mobile user's first request of the E-book |
| $M_{req}$ : | E-book page number request after last view |
| $RESP$ : | response message of the cloud to user's request |
| $E_{SK}(m)$ : | use the symmetrical key $SK$ to encrypt a message $m$ |
| $D_{SK}(m)$ : | use the symmetrical key $SK$ to decrypt a message $m$ |
| $A \stackrel{?}{=} B$ : | determine whether or not A and B are equal |
| $h(.)$ : | one way hash function |

### 3.1 Registration Phase

The user proposes an identification *ID* and password *pw* to the cloud through a secure channel. The cloud stores the user's authentication information in the verification table.

### 3.2 Authentication Phase

The user authenticates with the cloud, and generates a session key. Figure 3 shows our proposed authentication protocol.

**Step 1**: User enters *ID* and *pw*, and generates a nonce $N_u$ and computes $C_1$ and $C_2$ as follows:

$$C_1 = h(h(pw) \oplus IMEI) \oplus N_u \tag{1}$$

$$C_2 = h(ID \| h(pw) \| h(N_u \oplus IMEI)) \tag{2}$$

Afterward, the user sends (*ID*, *IMEI*, $C_1$ and $C_2$) to the cloud.

**Step 2**: The cloud first checks *ID* and uses the *ID* to identify the corresponding *pw* on the verification table. Then the cloud computes $N_u'$ and performs the authentication as Eq. (4)

$$N_u' = C_1 \oplus h(h(pw') \oplus IMEI) \tag{3}$$

$$h(ID \| h(pw') \| h(N_u' \oplus IMEI)) \stackrel{?}{=} C_2 \tag{4}$$

If Eq. (4) holds, then the cloud completes the user's authentication. The cloud generates $N_s$ and computes the communication session key for the next communication as follows:

$$SK = h(N_u \| N_s) \oplus IMEI \tag{5}$$

Afterward, the cloud sends $h(h(pw) \| IMEI \| N_u) \oplus N_s$ and $h(IMEI \| N_u \| N_s)$ to the user.

**Step 3**: The user computes $N_s'$ and checks $N_s'$

$$N_s' = h(h(pw) \| IMEI \| N_u) \oplus h(h(pw) \| IMEI \| N_u) \oplus N_s \tag{6}$$

$$h(IMEI \| N_u \| N_s) \stackrel{?}{=} h(IMEI \| N_u \| N_s') \tag{7}$$

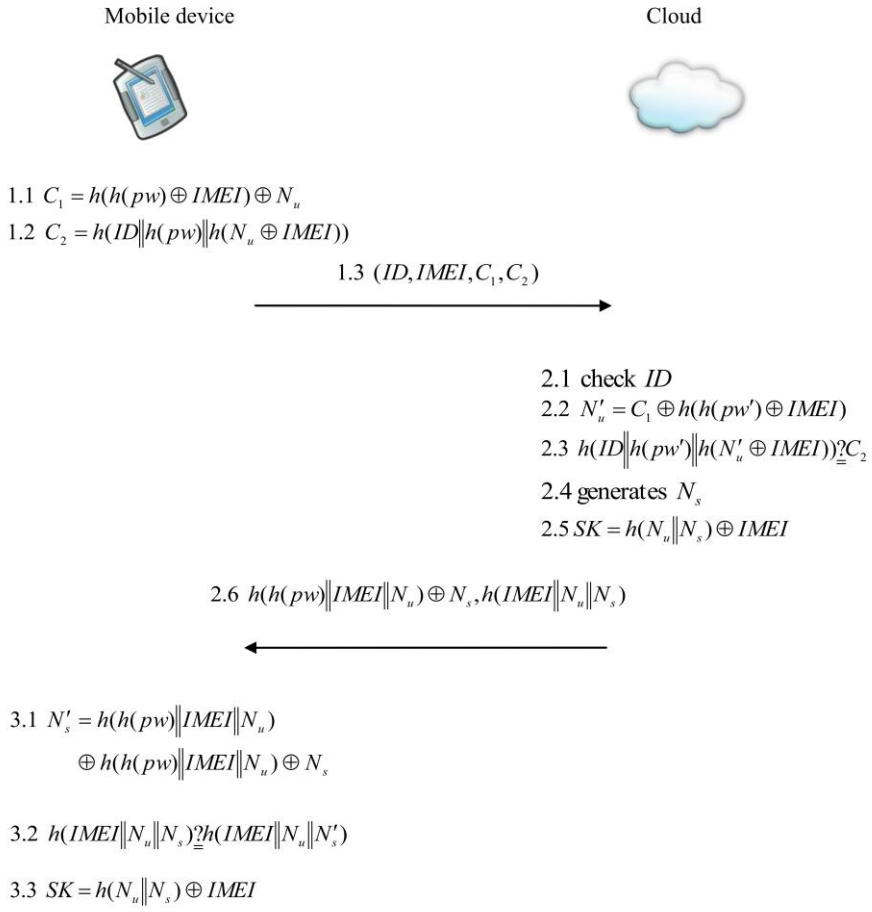


**Fig. 3.** The overview of our proposed authentication phase

If Eq. (7) holds, the cloud completes the mutual authentication with the user, and then the user can communicate the service message with the cloud. The user also generates a session key *SK* for the next communication.

$$SK = h(N_u \| N_s) \oplus IMEI \tag{8}$$

### 3.3 Service Response Phase

The user presents the service request by using the previous generated session key, and the cloud responds to the user's request. Figure 4 shows our proposed service response process.

**Step 1**: The user chooses the cloud service API license or asks to respond to the request; the cloud authenticates the user identity, generating a symmetric encryption message as follows:

$$C_3 = E_{SK}(E-book_{req}) \tag{9}$$

$$\text{or} \quad C_3' = E_{SK}(M_{req}) \tag{10}$$

A new nonce $N_{u+1}$ is generated and an authentication message is computed as follows:

$$C_4 = h(h(pw) \oplus IMEI) \oplus N_{u+1} \tag{11}$$

$$C_5 = h(ID \| h(pw) \| h(N_{u+1} \oplus IMEI)) \tag{12}$$

Afterward, the user sends $(ID, C_3, C_4, C_5)$ to the cloud.

**Step 2**: The cloud first checks *ID*, and uses the corresponding session key $SK$ to decrypt the service request.

$$E-book_{req} = D_{SK}(C_3) \tag{13}$$

$$\text{or} \quad M_{req} = D_{SK}(C_3') \tag{14}$$

The cloud computes $N'_{u+1}$

$$N'_{u+1} = C_4 \oplus h(h(pw') \oplus IMEI) \tag{15}$$

$$h(ID \| h(pw') \| h(N'_{u+1} \oplus IMEI)) \stackrel{?}{=} C_5 \tag{16}$$

If Eq. (16) holds, then the cloud generates the next nonce $N_{s+1}$, and calculates the new session key $SK_{new}$ as follows:

$$SK_{new} = h(N_{u+1} \| N_{s+1}) \oplus IMEI \tag{17}$$

User computes $C_6$ as follows:

$$C_6 = E_{SK_{new}}(RESP) \tag{18}$$

Afterward, the cloud sends $C_6'$, $h(h(pw) \| IMEI \| N_{u+1}) \oplus N_{s+1}$ and $h(IMEI \| N_{u+1} \| N_{s+1})$ to the user.

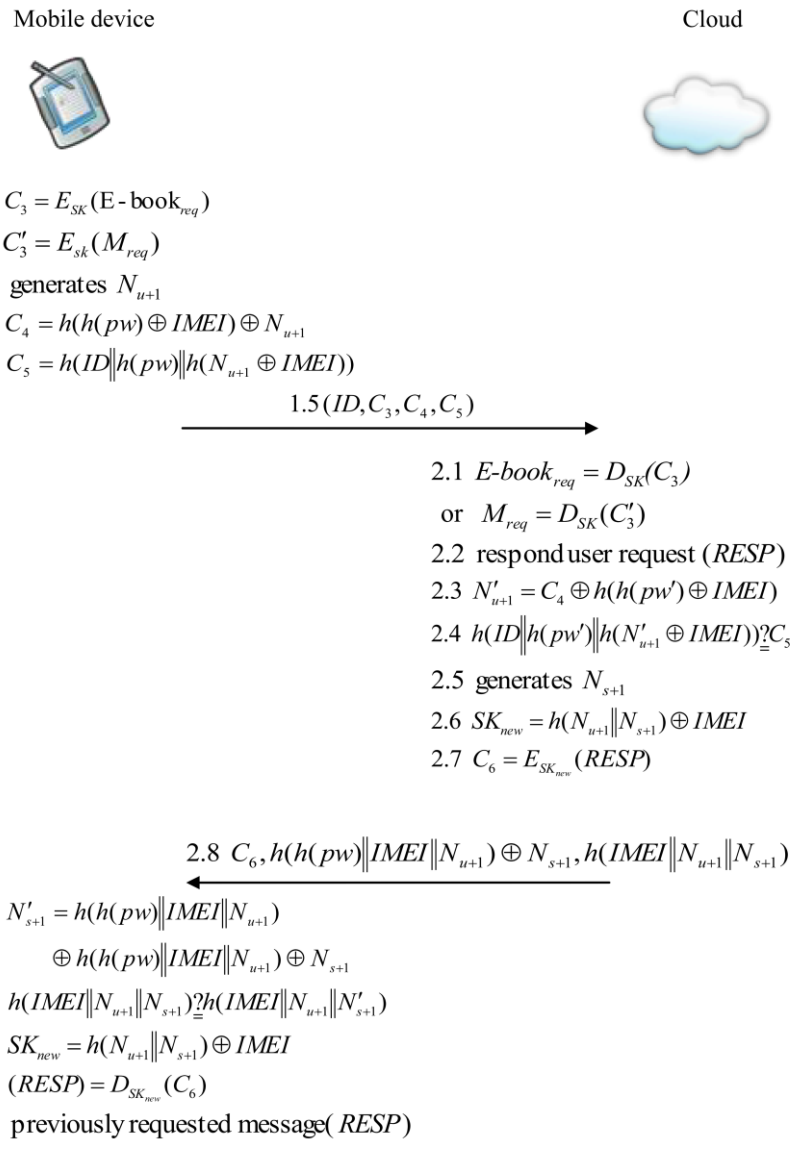


**Fig. 4.** The overview of our proposed service response phase

**Step 3**: The user computes $N'_{s+1}$ as follows:

$$N'_{s+1} = h(h(pw) \| IMEI \| N_{u+1}) \oplus h(h(pw) \| IMEI \| N_{u+1}) \oplus N_{s+1} \tag{19}$$

And authenticates the $N'_{s+1}$

$$h(IMEI \| N_{u+1} \| N_{s+1}) \stackrel{?}{=} h(IMEI \| N_{u+1} \| N'_{s+1}) \tag{20}$$

If Eq. (20) holds, then the user uses the previously generated $N_{u+1}$ to compute the $SK_{new}$ as follows:

$$SK_{new} = h(N_{u+1} \| N_{s+1}) \oplus IMEI \tag{21}$$

and decrypts $C_6$ to obtain the response message $RESP$

$$RESP = D_{SK_{new}}(C_6) \tag{22}$$

Thus, the user can access the previously requested message $RESP$.

In our proposed protocol, the cloud service user can continue to maintain a secure communication with the cloud.

## 4. Security Analysis

The following analysis is to show how our proposed scheme can prevent various attacks.

### 4.1 DOS Attack Prevention

As with the Amazon cloud infrastructure sites [5], the user's communication messages are the first through the firewall filters on the server. The user can synchronize with cookies to reduce abnormal malicious attacks, and users also must be limited to connect with the clouds. If users use the browser to perform malicious attacks, the server automatically locks the user's behavior. For example, if the same IP requests 1000 messages in one second, the user is regarded as a malicious attacker, and the server will block the IP services.

### 4.2 Password Guessing Attack Prevention

As the users do not store any user data in their mobile device, an attacker cannot achieve offline password guessing attacks via the mobile device. The cloud protects the users' accounts on the cloud end. If an attacker or a legitimate user enters consecutive incorrect passwords, the server will block the account, and the user will be requested to change the password and to send the registration information via email. Thus, there is no way to use online password guessing attacks since the attacker or the user does not know the previous password set.

### 4.3 Insider Attack Prevention

High value assets of the cloud system [5] request the user to change their password regularly, and the private key of the server will also be regularly changed. While the cloud stores the password, it does not directly store user passwords, and it is protected

by a one way hash function in order for users to store their passwords. For example, a user's password $pw$ and Linux's private key $x$ are protected by an MD5 hash function $h(h(pw) \oplus x))$. So, even if an insider attacker (root) steals the verification table, the attacker cannot use brute-force attacks to guess the user's password and identify the server's private key.

### 4.4 Reply Attack Prevention

Because the nonces $N_u$ and $N_s$ are not the same, even if the attacker were to intercept the messages ($C_1 = h(h(pw) \oplus IMEI) \oplus N_u$ and $h(h(pw) \| IMEI \| N_u) \oplus N_s$), in order to make a forged message $C_4 = h(h(pw) \oplus IMEI) \oplus N_{u+1}$ and $h(h(pw) \| IMEI \| N_{u+1}) \oplus N_{s+1})$, an attacker cannot use the intercepted messages to communicate with a user on the cloud during the authentication phase.

### 4.5 Impersonation Attack Prevention

Since each communication is recorded for a user's *ID* and *IMEI*, the user's password $pw$ is protected by a one way hash function ($C_1 = h(h(pw) \oplus IMEI) \oplus N_u$), so the attacker cannot successfully fake being the user during the communication process. Neither can an attacker fake being the server. Moreover, the user's password is difficult to work out. Only the legal cloud can compute the correct $N'_{u+1} = C_4 \oplus h(h(pw') \oplus IMEI)$, so the attacker cannot fake being the cloud.

### 4.6 Man-in-the-Middle Attack Prevention

Each message is protected by two unknown nonces $N_X$ and $h(pw)$, so even if an attacker intercepts the messages $C_1 = h(h(pw) \oplus IMEI) \oplus N_u$ and $h(h(pw) \| IMEI \| N_u) \oplus N_s$, the attacker cannot pass the authentication by the following equations: $h(ID \| h(pw') \| h(N'_{u+1} \oplus IMEI)) \stackrel{?}{=} C_2$ and $h(IMEI \| N_u \| N_s) \stackrel{?}{=} h(IMEI \| N_u \| N'_s)$. Thus, the Man-in-the-Middle attack will be prevented.

### 4.7 Parallel Sessions Attack Prevention

The user transmits the communication messages $(C_1, C_2, C_4, C_5)$ to the cloud, and the cloud responds with the messages $h(h(pw) \| IMEI \| N_u) \oplus N_s$ and $h(IMEI \| N_u \| N_s)$. Both

of the communication messages of the hash value are different; thus, the proposed scheme prevents parallel session attacks.

### 4.8 Session Key Error or Tampering

Our protocol aims at reducing the computation cost on the mobile device. Once the session key is checked, if an error occurs or the key is tampered with during the authentication, the user just needs to be authenticated again and log into the cloud to access the cloud services.

### 4.9 Comparison

From Figure 5, it can be seen that we combine the charging mechanisms and replace usage rights with licenses in order to change the method of E-book usage rights via purchase. The proposed scheme enables the cloud to easily record a user's reading information, and the last viewed page immediately, despite interface and device limitations. Users can read E-books free from the various devices and paid software (such as office series) limitations anytime and anywhere. We use symmetric encryption for the device to reduce the computation and communication cost, which is different from other suppliers' encryption mechanisms.

| | DRM model | Interface model | Limited device or software | Protected mechanism | Install related API |
|---|---|---|---|---|---|
| Apple | B | Cloud | Limited to a single product brand equipment | N/A | iTunes |
| Google | A, B | Cloud | free | SSL | adobe reader |
| Microsoft | A, C | Client-Server | Office Series | RSA | RMS |
| Our scheme | A, B, C | Cloud | free | Symmetric encryption | adobe reader |

A：DRM, B：DRM-Free, C：Exchange License

**Fig. 5.** The comparisons of the related works

## 5. Conclusions

The proposed cloud scheme not only provides more convenient E-book services, but allows users to apply to other cloud services, with the digital content stored in the cloud. Users can access E-books using different devices, anytime and anywhere. The digital content is protected by DRM, which is flexible via changing the license usage mechanism such that the cloud can record the user's information.

Our proposed protocol allows users to use different mobile devices to access the cloud services. The mobile devices do not need to store the user's privacy and cloud's related messages. In the communication process, we use low complexity functions (such

as hash function, exclusive-OR and lightweight operations [20, 21]) to reduce the computing cost of the mobile device, and we also address mutual authentication issues. This study realizes the following goals:
(1) Propose a cross-vendor authentication of the cloud.
(2) Resist known attacks.
(3) Provide a low computing cost for mobile user.
(4) Provide a user friendly use for the digital content.
(5) Provide a device-independent management for DRM.

Considering the distributed nature of protected DRM contents and also that the proposed protocol allows to use different mobile devices, some possible future work could be to extend the work in a way to be also applicable to interconnected federated cloud, such as proposed in [22].

## References


1. Albano, P., Bruno, A., Carpentieri, B., Castiglione, A., Castiglione, A., Palmieri, F., Pizzolante, R. and You, I.: A Secure Distributed Video Surveillance System Based on Portable Devices, Lecture Notes in Computer Science, Vol. 7465, pp 403-415, (2012).
2. Pizzolante, R., Carpentieri, B. and Castiglione, A.: Text Compression and Encryption through Smart Devices for Mobile Communication, Proceeding of 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2013), July 3-rd to July 5-th, 2013, Asia University, Taichung, Taiwan, pp. 672 - 677.
3. Ohlman, B., Eriksson, A., Rembarz, R.: What Networking of information Can Do for Cloud Computing. the 18th IEEE International Workshops on Enabling Technologies : Infrastructures for Collaborative Enterprises, 78-83, (2009).
4. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud Computing and Emerging IT Platfoms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems, Vol. 25, No. 6, 599-616, (2009).
5. Subashini, S., Kavitha, V.: A Survey on Security Issues in Service Delivery Models of Cloud Computing. Journal of Network and Computer Applications, Vol. 34, No. 1, 1-11, (2011).
6. Svantesson, D., Clarke, R.: Privacy and Consumer Risks in Cloud Computing. Computer Law & Security Review, Vol. 26, 391-397, (2010).
7. Chen, C.L.: A Secure and Traceable E-DRM System Based on Mobile Device. Expert Systems With Applications, Vol. 35, No. 3, 878-886, (2008)
8. OpenID. http://openid.net/government/, Access available 13/8/2013.
9. Mell, P., Grance, T.: The NIST Definition of Cloud Computing (Draft). http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf, Access available 4/8/2011, (2011).
10. Apple DRM-Free. http://www.apple.com/pr/library/2007/04/02itunes.html, Access available 13/8/2013.
11. Google DRM-Free. http://books.google.com/support/partner/bin/answer.py?hl=en&answer=170424, Access available 13/8/2013.
12. Google DRM. http://books.google.com/help/ebooks/content.html, Access available 13/8/2013.
13. Google adopts Adobe ebook DRM. http://blogs.adobe.com/digitalpublishing/2010/12/google-ebooks.html, Access available 13/8/2013.

14. Chen, Y.Y., Wang, Y.J. and Chen, J.C.: A Fair-use DRM System Based on Web Service. Eighth International Conference on Intelligent Systems Design and Applications, Vol. 3, No. 11, 11-16, (2008).
15. Lee, W. B., Wu, W. J., Chang C. Y.: A Portable DRM Scheme Using Smart Cards. Journal of Organizational Computing and Electronic Commerce, Vol. 17, No. 3, 247-258, (2007).
16. Windows Rights Management Services http://technet.microsoft.com/zh-tw/library/cc706990(WS.10).aspx, Access available 13/8/2013.
17. Rivest, R., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of the ACM, Vol. 21, No. 2, 120-126, (1978).
18. Castiglione, A., De Prisco, R. and De Santis, A.: Do You Trust Your Phone?, Lecture Notes in Computer Science, Vol. 5692, pp 50-61, (2009).
19. Chen, C.L.: All-In-One Mobile DRM System Design. International Journal of Innovative Computing, Vol. 6, No. 3A, 897-911, (2010).
20. Chen, C. L., Tsai, Y. T.: Aniello Castiglione and Francesco Palmieri, Using Bivariate Polynomial to Design a Dynamic Key Management Scheme for Wireless Sensor Networks. Computer Science and Information Systems, Vol. 10, No. 2, 589-609, (2013).
21. Chen, C. L., Tsai, W. C.: Using a Stored-value Card to Provide an Added-value Service of Payment Protocol in VANET. 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2013), July 3-rd to July 5-th, 2013, Asia University, Taichung, Taiwan.
22. Esposito, C., Ficco, M., Palmieri, F. and Castiglione A.: Interconnecting Federated Clouds by Using Publish-Subscribe Service, Cluster Computing, In press, DOI 10.1007/s10586-013-0261-z, (2013).

**Chin-Ling Chen**, PhD, is a member of the Chinese Association for Information Security. From 1979 to 2005, he was a senior engineer at the Chunghwa Telecom Co., Ltd. He is currently a professor of the Department of Computer Science and Information Engineering at Chaoyang University of Technology, Taiwan. His research interests include cryptography, network security and electronic commerce. Dr. Chen had published over 50 SCI/SSCI articles on the above research fields in international journals.

**Woei-Jiunn Tsaur**, PhD, worked as a project manager and technology consultant from 1994 to 2003 in R&D Division of Syscom Computer Engineering Co., a research center of software development in Taiwan. Since 1999, he has been with the Department of Information Management at Da-Yeh University, Taiwan, where he is currently a full professor. His research interests include network security, security topics in operating systems, applied cryptography, information security management and computer networks. He has directed many research projects in the areas of network security and cloud computing security. Dr. Tsaur is also a member of the IEEE and the Chinese Cryptology and Information Security Association.

**Yu-Yi Chen**, PhD, is presently an associate professor of the Department of Management Information systems, National Chung Hsing University, Taiwan. His research interests include computer cryptography, network security, and e-commerce.

**Yao-Chang Chung** was born in 1987. He received the B.S degree in Department of Computer Science and Information Engineering from St. John's University, Taipei Taiwan in 2010. He received his Master degree at the Department of Computer Science and Information Engineering, Chaoyang University of Technology in 2012. His research interests include information security and cloud security.

*Received: September 19, 2013; Accepted: January 6, 2014.*