www.ietdl.org

Published in IET Information Security
Received on 18th June 2012
Revised on 27th December 2012
Accepted on 28th January 2013
doi: 10.1049/iet-ifs.2012.0198

ISSN 1751-8709

uCloud: a user-centric key management scheme for cloud data protection

Yung-Wei Kao, Kuan-Ying Huang, Hui-Zhen Gu, Shyan-Ming Yuan

Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan
E-mail: ej3muse@gmail.com

Abstract: One of the most challenging problems of cloud service solicitation is to persuade users to trust the security of cloud service and upload their sensitive data. Although cloud service providers can claim that their services are well-protected by elaborate encryption mechanisms, traditional cloud systems still cannot persuade the users that even if the cloud servers are compromised, the data are still securely protected. This study proposes uCloud, a user-centric key management scheme for cloud data protection, to solve this problem. uCloud utilises RSA and indirectly encrypts users' data by users' public keys, but stores the users' private keys on neither servers nor users' PCs; instead, the private keys are stored on users' mobile devices and presented via two-dimensional (2D) barcode images when they are utilised to decrypt users' sensitive data. In this manner, users' data are safely protected even if the cloud servers are compromised. Also, uCloud provides users with the experience of managing visible private keys by storing the keys into mobile phones and displaying them via 2D barcodes. Moreover, three scenarios: personal storage, home surveillance and enterprise storage scenarios are proposed to present the practicability of uCloud. In addition, a hierarchical structure is designed for basic key backup and data sharing in the proposed scheme.

1 Introduction

Benefit from the idea of cloud and related technologies [1–3], cloud services are popular in recent years. More and more users and enterprises tend to upload their data onto cloud servers so that the data can be maintained properly with the scalability, ubiquity and accessibility properties. However, since users usually do not know or trust the security level of external cloud services, risk management of out-sourcing data storage service is a critical issue [4–6].

In addition to normal files, video recorded by home surveillance system [7–9] is also a kind of sensitive data. To satisfy the needs of home healthcare and home security, numerous people install surveillance systems into their houses. Since users usually do not maintain media servers by themselves, most of them tend to upload the videos onto online servers. However, several sensitive behaviours such as changing clothes may also be recorded without users' awareness; if the online servers are compromised, these sensitive videos may be accessed by attackers.

Another example of sensitive data are trade secret. Traditionally, enterprises manage their trade secrets by themselves. Although uploading these trade secrets to cloud servers can reduce the cost of data management, the risk of secret leakage is greater. Business rivals, or even governments, may try to steal the trade secrets by employing hackers or sending insiders to the cloud service operator. Moreover, since the ownership of enterprise data usually belongs to the enterprises, not creators, the data must be accessible to the creator's direct or indirect bosses according to the enterprise hierarchy structure but not to other unauthorised employees.

Traditional cloud systems usually utilise accounts and user-selected passwords for authentication; only the users who provide the correct passwords can access the protected data. However, since users must remember their passwords, users tend to chose simple and meaningful passwords; this kind of passwords is easy to be uncovered by dictionary attack. Another solution of authentication is adopting one-time password devices. In this manner, users must bring one more hardware device with them for each service; it is very inconvenient. The most serious problem of using authentication mechanism to protect user's data are that the right of accessibility is controlled by the authentication system; if the authentication system is compromised, the data can be accessed by attackers.

The most common and trusted solution to protect data are to encrypt and decrypt data by encrypting and decrypting keys. However, the design of key management of these keys is the greatest challenge of this solution if both the data confidentiality and sharing requirements should be satisfied. Traditionally, the keys are designed to be stored on PCs where the data are encrypted and decrypted. In this manner, although the data can be safely protected, it is difficult to be shared with other users who are authorised to

access it. Also, the keys are not portable and not convenient to be used on other PCs.

Since current cloud systems still cannot persuade users that the sensitive data can be safely protected, a secure and convenient mechanism is necessary to be developed and make users feel safe. This paper proposes uCloud, a user-centric key management scheme for cloud data protection, to solve this problem. uCloud utilises RSA and indirectly encrypts users' data by users' public keys, but neither stores the users' private keys on servers nor on PCs; instead, the private keys are stored on users' mobile devices and presented via two-dimensional (2D) barcode images when they are utilised to decrypt users' data. In this manner, users can see the unique 2D barcode images of private keys and believe that their data cannot be accessed without the presentation of these images on their mobile phones even if the public cloud servers are compromised. Moreover, uCloud also includes a hierarchical structure for basic key backup and data sharing. For enterprises which have complex access control rules, a trusted server can be maintained to provide complete access control and key backup.

The research contains seven sections. In Section 2, the backgrounds and related works of the proposed system are introduced. Section 3 discusses the overview and architecture of uCloud. Section 4 defines system assumptions. We present the detailed system design of uCloud in Section 5. System demonstrations and evaluations of scenarios are presented in Section 6. Finally, the paper ends up with a conclusion and discusses the future works of uCloud in Section 7.

2 Backgrounds and related works

2.1 Cloud data protection

In traditional cloud services such as Google Gmail, Facebook and home surveillance services [10], users' data are managed and protected by service providers. If the cloud servers are compromised, users' data may suffer from the leakage problem. Several researches [11] include the idea of client application to encrypt data before it is transferred to cloud and decrypt data after it is downloaded. However, since the client application is coupled with user's computer, the decryption keys of data are difficult to be shared with other users or another user's client application. Kamara and Lauter [12] proposed a data sharing and searching architecture between the data owners and other users. However, the attribute-based encryption is adopted to generate the decryption key according to the policy used to encrypt the data. In other words, the access policy is fixed and not flexible unless the data are re-encrypted by a new policy. Dai and Zhou [13] proposed another approach similar with the Kamara's. In this approach, the access control matrix (ACM) is sent to cloud storage provider so that the provider can check this matrix for each request. However, in this manner, each user has to define his (her) ACM; in the enterprise scenario, it is very difficult to maintain such a huge matrix for each user. Sanka *et al.*[14] proposed a more advanced architecture to maintain an access control list in the owner server. However, in this approach, the owner has to maintain a server and manage the access control list (ACL); it is not applicable in the personal storage scenario.

2.2 Device paring and constrained channel

In systems such as zero-interaction authentication (ZIA) [15], the computer is usually required to be paired, or bound to a mobile device for establishing a communication channel. Wireless technologies such as IEEE 802.11 or Bluetooth are usually adopted so that the communication channel can be constructed automatically without user's intervention. ZIA emphasised that this channel must be a short-range or constrained channel to prevent the transmitted information from being eavesdropped. Constrained channel can be constructed by various technologies with distinct properties. infrared light-emitting diode (IR LED) is not supported by most mobile devices, thus it is not applicable for mobile scenario. Bluetooth and other wireless technologies provide the capability of zero user intervention, but in this manner, the constrained channel can be established without owner's awareness. Near field communication (NFC) [16–18] is a new technology which enables mobile devices to establish a short-range wireless and contactless communication channel. If NFC is deployed on mobile phones, users can also use it for conducting authentication based on the SIM cards. The wireless channel established based on NFC is a very secure constrained channel, because that it provides a very short range (< 0.2 m) of communication. However, most PCs have not supported NFC yet until now.

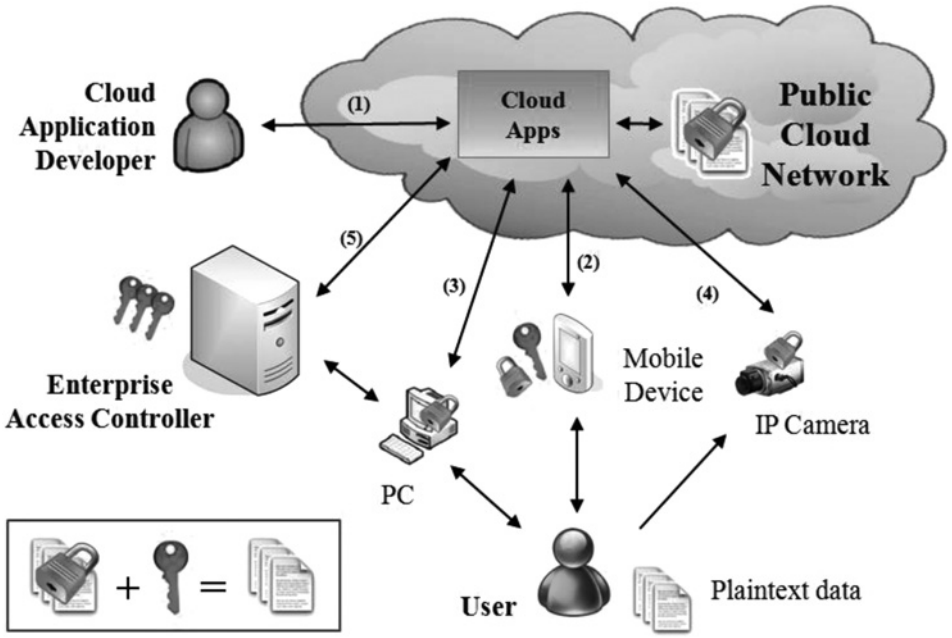
2.3 2D Barcode and quick response (QR) code

Compared with 1D barcode, 2D barcode [19–21] is able to store more information within an image for information exchange. The channel between 2D barcode displayer and scanner is a constrained channel, since that if images are taken with a longer distance, they become smaller, more ambiguous and more difficult to be decoded correctly. This kind of constrained channel requires users to perform a simple action: locate the scanner and displayer together. In this manner, the establishment of this channel can be confirmed by users with little user intervention. Another benefit of using 2D barcode is that the unique image with private key included provides users with the experience of owning a physical key.

QR code [22–24] is a kind of two-dimension barcode proposed in 1994; it is designed for encoding and decoding data between text contents and images rapidly. QR code is an easily used technology to store information such as URL of Web site, phone number and e-mail address. In numeric format, the max size of QR code message can be up to 7089 characters; in binary format, the maximum amount is 2953 bytes. QR code also supports error correction for fault tolerance. Different modes can be used according to different requirements. Level L mode provides 7% error correction rate, and Level H provides the highest error correction rate of 30%. Fault tolerance is a crucial feature since that perfect photographing cannot be guaranteed. With the fault tolerance feature, users do not have to try multiple times for decoding the captured images with few errors. In recent years, since the QR code technology has been supported by numerous mobile phones such as Android phones, it has been widely adopted in various mobile applications, and most mobile users are already familiar with using it.

3 System overview and architecture

Fig. 1 shows the system overview of uCloud. In this figure, the red key and red lock represent a pair of



(1) Application developer uploads cloud application.

(2) User applies an account via mobile phone.

(3) User uploads or downloads data to or from public cloud via PC under personal scenario.

(4) IP camera captures and encrypts images under home scenario.

(5) Enterprise user uploads or downloads data via Enterprise Access Controller under enterprise scenario.

Fig. 1 *System overview of uCloud*

private and public key, respectively. First of all, the cloud application developers can upload and register their applications onto an un-trusted public cloud. Second, the users can generate their own public and private key pairs by mobile applications, and register their accounts of the public cloud applications. For the personal usage scenario, users can encrypt and upload their files by PC applications and decrypt these files by showing 2D barcode images, which include the users' private keys, on mobile devices to PC applications. For the home surveillance scenario, IP cameras can encrypt streaming frames indirectly with the users' public keys. The encrypted streaming frames can be decrypted and displayed by PC applications after the private keys are extracted from 2D barcodes. Finally, for the enterprise scenario, the Enterprise Access Controller (EAC) provides the access control service to decide whether a user can access another user's secret files.

Fig. 2 shows the high level architecture of uCloud public cloud. Similar to traditional cloud service, cloud applications of uCloud are deployed upon cloud platforms (such as the Hadoop platform) in application servers. Based on this architecture, the uCloud module is included as a middleware between applications and cloud platform to manage security issues.

In the uCloud module, whenever a cloud application is registered by application developer, the Application Registration Manager registers the application, and returns the ID and password of this application. If any cloud application utilises any functionality of uCloud module, its ID and password must be verified by the application authenticator. Although the applications can maintain their own user accounts, the User Registration Manager manages a global identifier space where each account in each application is mapped to a unique global uCloud identifier. Finally, the Relationship Manager maintains the relationship or enterprise architecture between different users.

Whenever an enterprise user sends a request to access other user's data, the EAC must decide that whether this request is permitted via the decision maker module based on access control rules. If this request is permitted, the data manager module is executed to generate the encrypted owner's private key. All the private keys of enterprise users must be stored in the key storage of EAC to fulfill these requests and the backup requirement.

Fig. 3 shows the low level architecture of PC application. Different PC applications can be provided and associated with different cloud applications. When users want to encrypt and upload their files to public cloud, the data protector is invoked. When downloading files, users can browse their files via cloud application on Web browser, and the Application Loader which further executes the proper PC application is automatically executed after any link of file is clicked to be downloaded. Then, the Matrix Code Decoder is executed to capture the QR code image displayed before the Webcam and extract user's private key. Finally, the

doi: 10.1049/iet-ifs.2012.0198

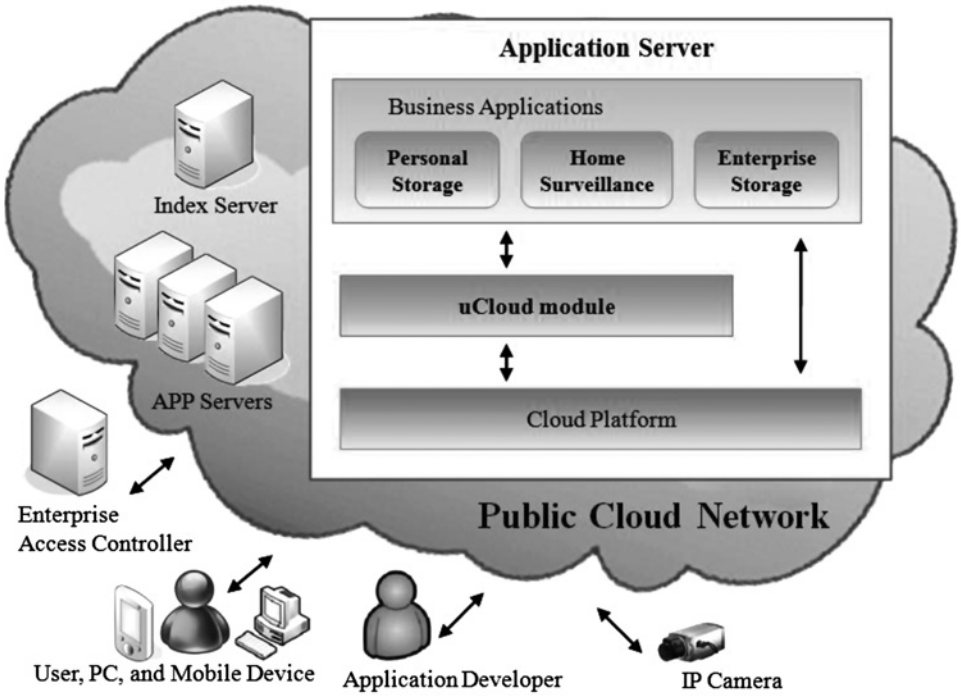


Fig. 2 *High level architecture of uCloud public cloud*

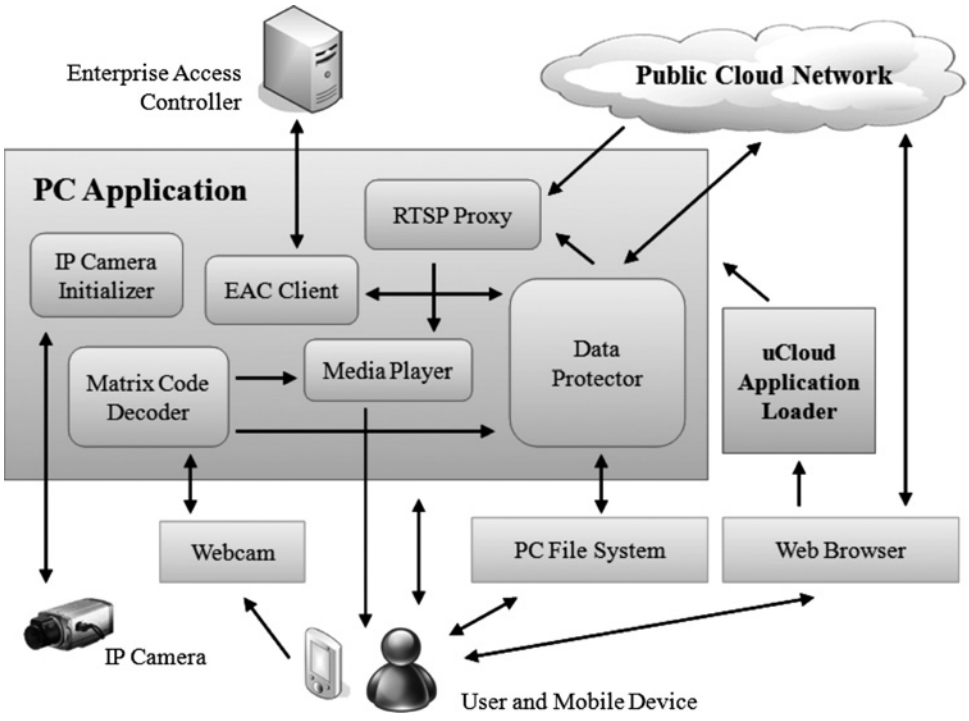


Fig. 3 *Low level architecture of PC application*

Data Protector communicates with cloud application to obtain the encrypted session key, and decrypt it by user's private key. If enterprise user wants to access other user's or group data, the EAC client is utilised to communicate with EAC, and obtain the correct session key for data decryption. If users want to monitor streaming video of IP camera, the Data Protector invokes the real time streaming protocol (RTSP) Proxy with the decryption session key for streaming decryption. Finally, the video can be displayed by the Media Player of PC application.

4 System assumptions

To define the scope of uCloud, several assumptions are defined. Four main parties are included in uCloud: mobile phone, PC application, enterprise access controller and public cloud. The mobile phone keeps the user's root secret (private key), so it is the root of trust of uCloud and is not compromisable. One user account can be associated with only one mobile phone. The PC and PC application are not compromisable; otherwise, the attacker can obtain the

decrypted files directly. Although PC is not compromised and private keys can be safely stored on PCs, multiple copies of keys must be maintained by users if the data are required to be decrypted on multiple PCs; it is the usibility concern rather than security concern to store keys on mobile phones rather than PCs. In addition, we assume that the public cloud is compromisable but the enterprise access controller is not. Since the enterprise access controller only manages the private keys of enterprise users, access control can be provided without maintaining the large files. Therefore users can feel safe to upload their sensitive data to public cloud. Also, the public cloud service provider must already have a public-key infrastructure (PKI), $\langle EK_{\text{uCloud}}, DK_{\text{uCloud}}\rangle$, to use, and the public key EK_{uCloud} is included directly into the PC and mobile phone applications and maintained in the enterprise access controller. Since numerous researches [25–27] have proposed various revocation mechanisms for PKI, such as creating certificates via CA, this paper does not focus on this issue. In addition, the mobile and PC applications must not be modified or replaced by adversaries before they are downloaded and installed by users; otherwise, they can be controlled by attackers. Moreover, users can delegate the access rights of sensitive files to the delegatees who they trust. Finally, we assume that RSA is not compromisable, and use RSA in most encryption and decryption operations of uCloud.

5 Detailed system design

The design of uCloud includes three scenarios: the personal usage, home surveillance and enterprise scenarios. To describe the protocols of them, numerous notations are defined in Table 1 and employed throughout this paper.

To register an account, the user-selected account UID_{APP} and π are encrypted with a random nonce R_1 by the uCloud's public key EK_{uCloud}. If this registration request is approved by cloud application, the cloud application logins to uCloud module by providing $\text{APPID}_{\text{uCloud}}$ and $\text{APPKEY}_{\text{uCloud}}$ to register the account. The registration result is sent back to user with a signature sig_1 of uCloud module, where $\text{sig}_1 = Da(\text{result}||\text{UID}_{\text{APP}}||\pi, DK_{\text{uCloud}})$. Then, the mobile application can generate a pair of public and private keys, encode the public key into a QR code image and send the public key to PC application. Finally, the PC application forwards the user's public key to the uCloud module.

Table 1 Notation description

Notation	Description
UID_{APP}, π	The user account and password used to login the cloud application
$\text{UID}_{\text{uCloud}}$	The global user identifier registered by cloud application and maintained by uCloud
$\text{APPID}_{\text{uCloud}}$, $\text{APPKEY}_{\text{uCloud}}$	The account and password used by cloud applications to login the uCloud module
Rx	Random numbers (can be used as session keys) ($x = 1, 2, 3, \ldots$)
EK_{uCloud}, DK_{uCloud}	The public and private keys of uCloud module
EK_{EAC}, DK_{EAC}	The public and private keys of Enterprise Access Controller
EK_{user}, DK_{user}	The user's public and private keys
EK_{cam}, DK_{cam}	The public and private keys of IP camera
$Ea(d, k)$, $Da(d, k)$	Asymmetric encryption and decryption functions for data d and key k
$Es(d, k)$, $Ds(d, k)$	Symmetric encryption and decryption functions for data d and key k ($Es = Ds$)
$Me(d)$, $Md(d)$	Matrix code encoding and decoding functions for data d
$\text{Path}\left(\text{UID}^1_{\text{APP}}, \text{UID}^2_{\text{APP}}\right)$	The hierarchical path from $\text{UID}^1_{\text{APP}}$ to $\text{UID}^2_{\text{APP}}$

5.1 Personal usage scenario

The personal usage scenario is designed for users to upload and backup their files on one PC or laptop, and download them to other trusted computers. In this scenario, regular files are protected by using the PC applications on users' computers and a personal storage cloud application. Fig. 4 shows the sequence diagram to encrypt and upload files. First, users enter their accounts and passwords to login the Personal Storage Application, and choose a random number R_2 to be the file encryption session key. The uploaded file is encrypted by this session key by symmetric encryption function, and this key is encrypted by user's private key; therefore users can extract it by using their private keys in the future.

Fig. 5 shows the sequence diagram of downloading the previously uploaded files. After logins to Personal Storage Application, the PC application sends a request to retrieve the files and related metadata by providing the file names. Then, uCloud module sends back the encrypted file and session key together with the signature $\text{sig}_2 = Da(h(Es(\text{File}, R_2)||Ea(R_2, EK_{\text{user}})), DK_{\text{uCloud}})$, where h is a hash

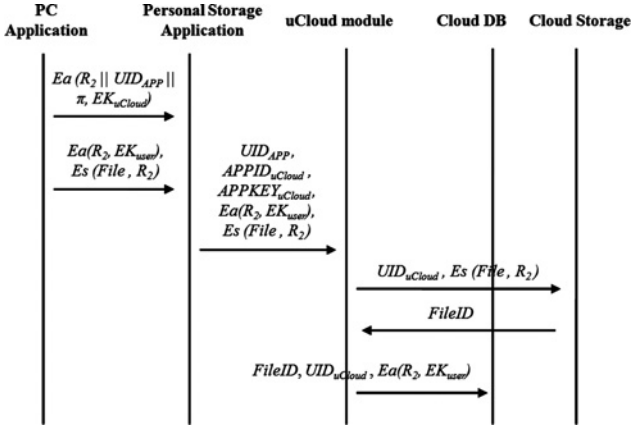


Fig. 4 *Sequence diagram of uploading file*

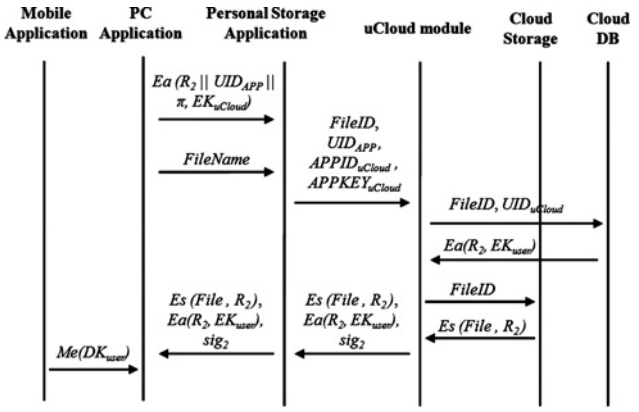


Fig. 5 *Sequence diagram of downloading file*

doi: 10.1049/iet-ifs.2012.0198

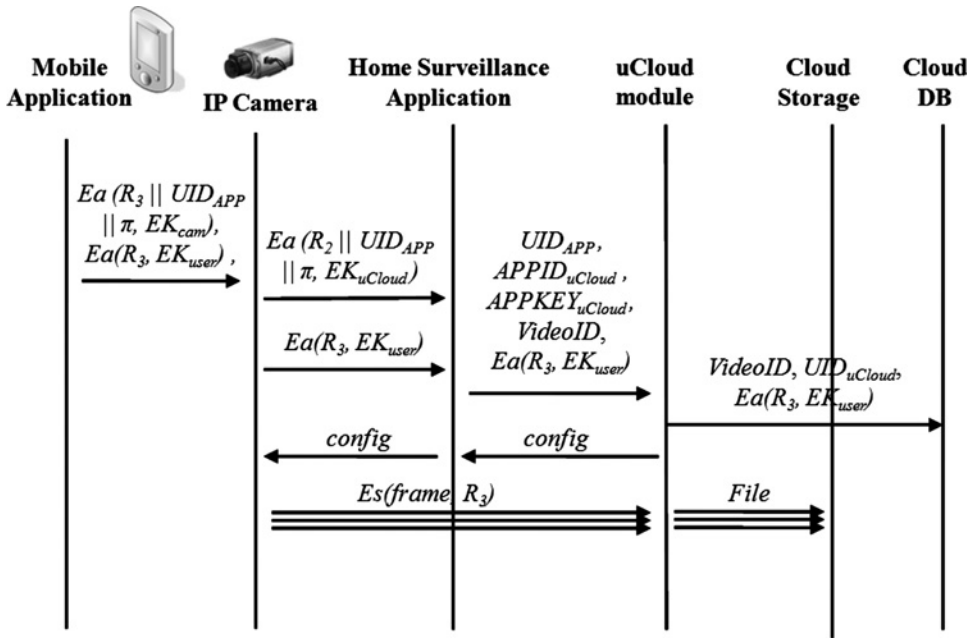


Fig. 6 *Sequence diagram of transmitting and storing streaming video*

function for reducing the size of signed message. After the encrypted file and metadata is downloaded, the user is asked to present a QR code containing user's private key. Therefore the PC application is able to decrypt the session key R_2, and use it to decrypt the file further.

5.2 Home surveillance scenario

The home surveillance scenario is designed for users to monitor streaming surveillance video or store the video onto cloud storage. In this scenario, a programmable IP camera is required for streaming frame encryption. To initialise the IP camera, the PC application generates a pair of camera's public and private keys, and embeds them into the IP camera application. After the IP camera application is installed, the camera's public key is sent to the mobile application by presenting the QR code image of it.

As shown in Fig. 6, whenever the user wants to transmit the streaming video of IP camera, the random number R_3 is selected as the encryption session key. Then, the user's account, password and session key are sent to the IP camera. After the IP camera logins to the Home Surveillance Application, it uploads the encrypted session key and requests the construction of RTSP communication. The uCloud module decides whether the connection can be established, and returns related configuration information if it is permitted. Therefore the IP camera can establish the RTSP connection to uCloud module, encrypts the frames by symmetric encryption function with the R_3 key, and transmits the encrypted streaming video to uCloud module.

Fig. 7 shows the sequence diagram of receiving surveillance streaming video. After login to the Home Surveillance Application, users can select which video to see. If any link of video is clicked, the PC application is automatically invoked by the Web browser and uCloud Application Loader to obtain the encrypted session key R_3 and establish an RTSP connection with uCloud module. The signature $sig_3 = Da(Ea(R_3, EK_{user})||config, DK_{uCloud})$. Then, the PC application extracts the user's private key, decrypts the streaming frames by R_3 and displays the video to users.

5.3 Enterprise scenario

For enterprise users, uCloud module provides a hierarchical key management scheme for basic data sharing. In general, a management hierarchy is established in enterprise with multiple levels. For example, Fig. 8 shows a hierarchy containing chairman, CEO, customer service and RD staffs. To protect their data, a pair of public and private keys is selected by each one of themselves. However, enterprise data are not personal property; at least, the owner's boss must also be able to access the owner's data. Therefore a copy of the owner's private key encrypted by the public key of the owner's direct boss must be maintained. Moreover, users can also delegate their keys to other users who they trust in case they are not familiar with computers or they want to backup their keys. In this case, the data owner's private key is also encrypted by the public key of the trusted person and maintained in uCloud.

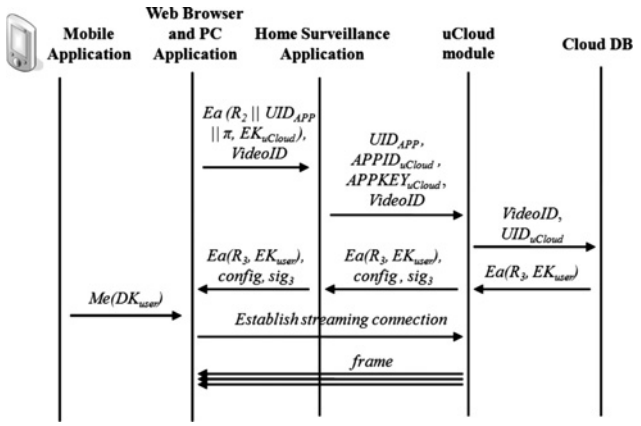


Fig. 7 *Sequence diagram of receiving and monitoring streaming video*

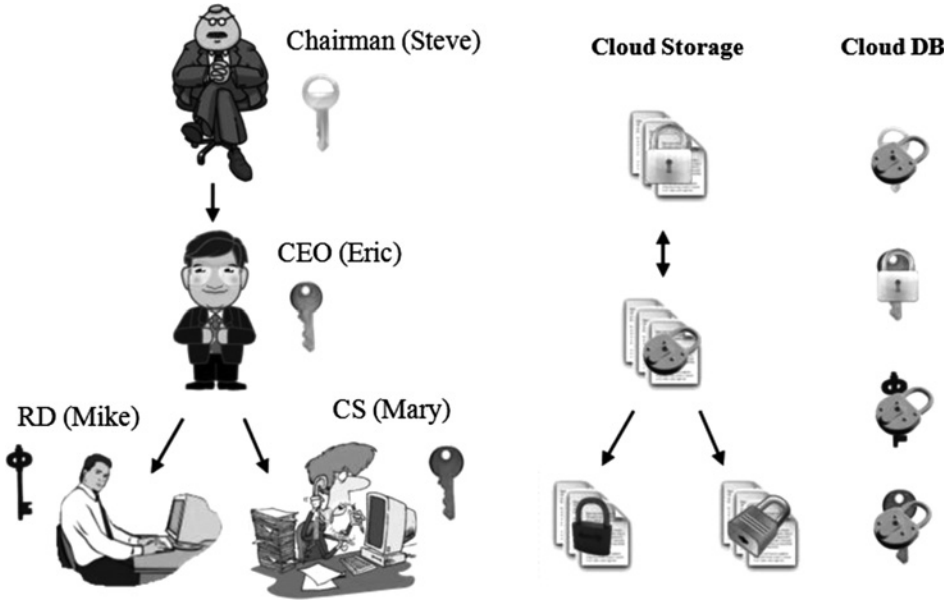


Fig. 8 *Example of enterprise hierarchy and key relationships*

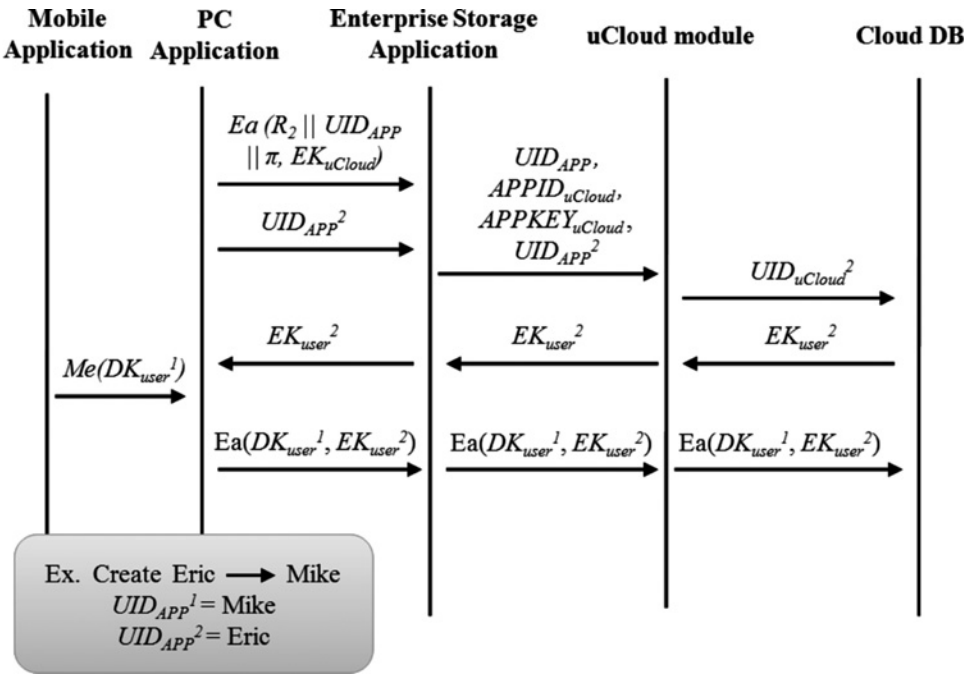


Fig. 9 *Sequence diagram of hierarchy path construction*

Fig. 9 shows an example of hierarchy path construction. In this case, the RD Mike wants to specify that the CEO Eric is his direct doss. To achieve this goal, Mike logins to the Enterprise Storage Application first to obtain the Eric's public key. Then, the Mike's private key is provided by the mobile application, encrypted by Eric's public key and uploaded to uCloud module.

Fig. 10 shows an example that the data owner's indirect boss wants to obtain the owner's data. In this case, the chairman Steve wants to access the RD Mike's data. According to the file ID, uCloud module finds out the file's owner and the encrypted session key $Ea(R_2, EK^3_{\text{user}})$ of this file. Then, any path from Steve to Mike is searched by depth-first search. If any path exists from Steve to Mike, the encrypted private keys of all users along with this path are gathered. The path, encrypted private keys, encrypted session key, encrypted file and signature sig_4, are sent back to the PC application together, where

$$\begin{aligned}\text{sig}_4 = Da(h(&\text{Path}(\text{UID}^1_{\text{APP}}, \text{UID}^3_{\text{APP}}) \| Ea(DK^2_{\text{user}}, EK^1_{\text{user}}) \| \\ &Ea(DK^3_{\text{user}}, EK^2_{\text{user}}) \| \cdots \| Ea(R_2, EK^3_{\text{user}}) \| \\ &Es(\text{File}, R_2)), DK_{u\text{Cloud}})\end{aligned}$$

After the private key DK^1_{user} is obtained, the DK^2_{user}, DK^3_{user}, R_2 and the file can be decrypted sequentially.

Although uCloud module provides the hierarchical data sharing mechanism, more complex access control

doi: 10.1049/iet-ifs.2012.0198

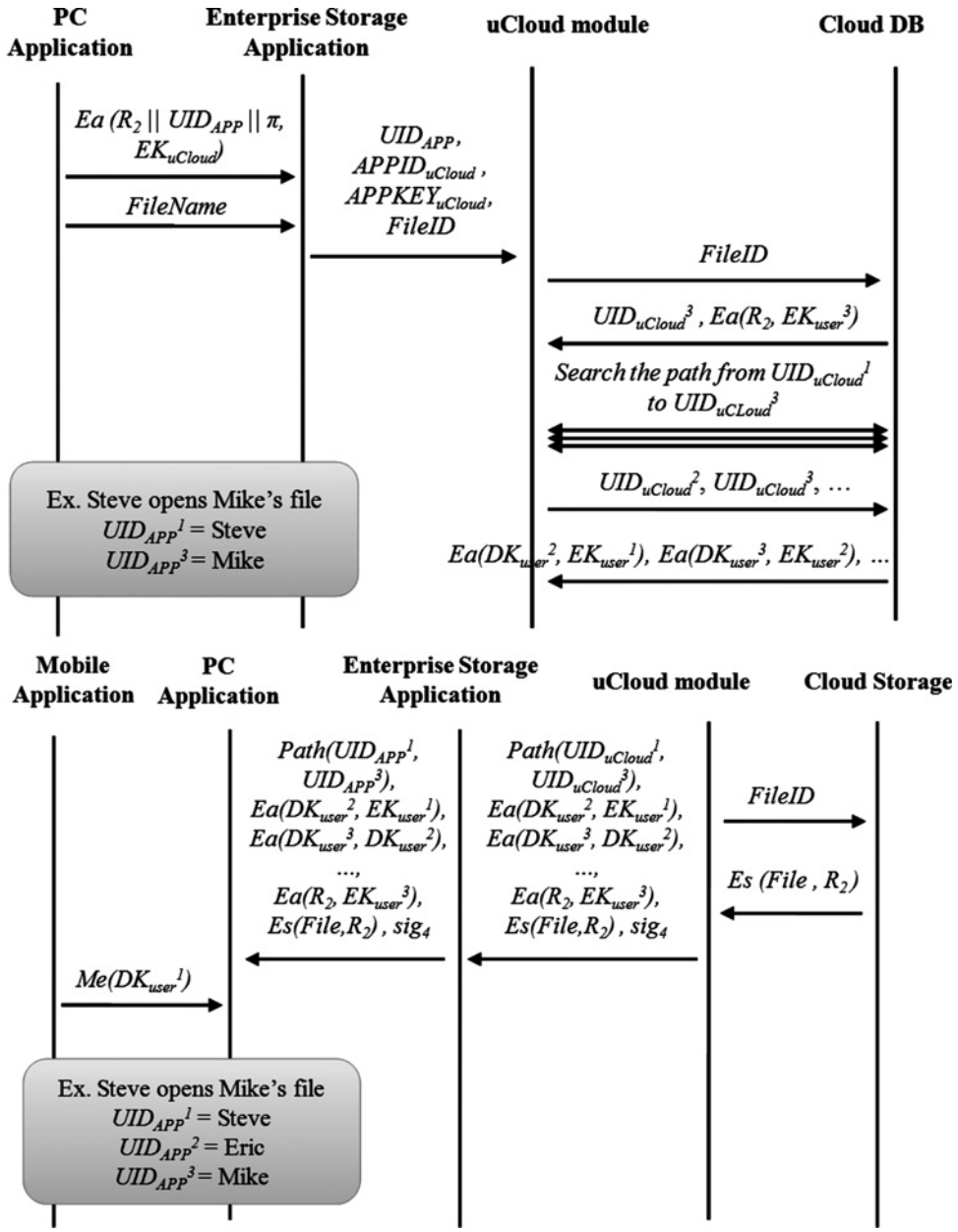


Fig. 10 *Sequence diagram of file sharing*

mechanism such as sharing data between users belonging to the same group but not superiors to each other is not supported. To solve this problem, enterprises can maintain their access control servers, EACs, to cooperate with uCloud module. Fig. 11 shows the sequence diagram of file sharing via EAC. In this case, the RD Mike wants to obtain the CS Mary's file. First, Mike encrypts his account and password by EAC's public key, sends the result to EAC, and requests for obtaining Mary's file. If this request is permitted, the EAC uses Mike's account and password to login the Enterprise Storage Application, and the Enterprise Storage Application logins to uCloud module. According to the file ID, uCloud module gathers the encrypted session key $Ea(R_2, EK^2_{\text{user}})$ and the encrypted file, and sends the results and sig_5 to EAC, where $\text{sig}_5 = Da(h)(\text{UID}^2_{\text{APP}} \| Ea(R_2, EK^2_{\text{user}}) \| Es(\text{File}, R_2))DK_{\text{uCloud}}$. Since EAC maintains Mary's private key, DK^2_{user}, this key is encrypted by Mike's public key and sent to the PC application. Finally, Mike can decrypt the file by using his private key, the DK^2_{user}, and R_2.

5.4 Security analysis

As the assumptions mentioned previously, the cloud service, including cloud applications, uCloud module, cloud storage and cloud DB, are compromisable. However, since user's data are properly encrypted by user's private key which is not stored on cloud, adversaries cannot access the plaintext of data. Although attackers can only delete the encrypted data and session keys, this problem can be solved by self-maintained backups; it is out of scope of this paper. Also, if the hierarchy maintained by uCloud module is modified, the protected data are still secure since the session keys are only encrypted by the public keys of other trusted users. Moreover, since the mobile device is the root of trust, it is not compromisable. If user's mobile phone is stolen, users can login to the cloud application and delete their data as soon as possible, since users usually bring their phones with them and use them frequently. Therefore only if the mobile phone is stolen and the cloud service

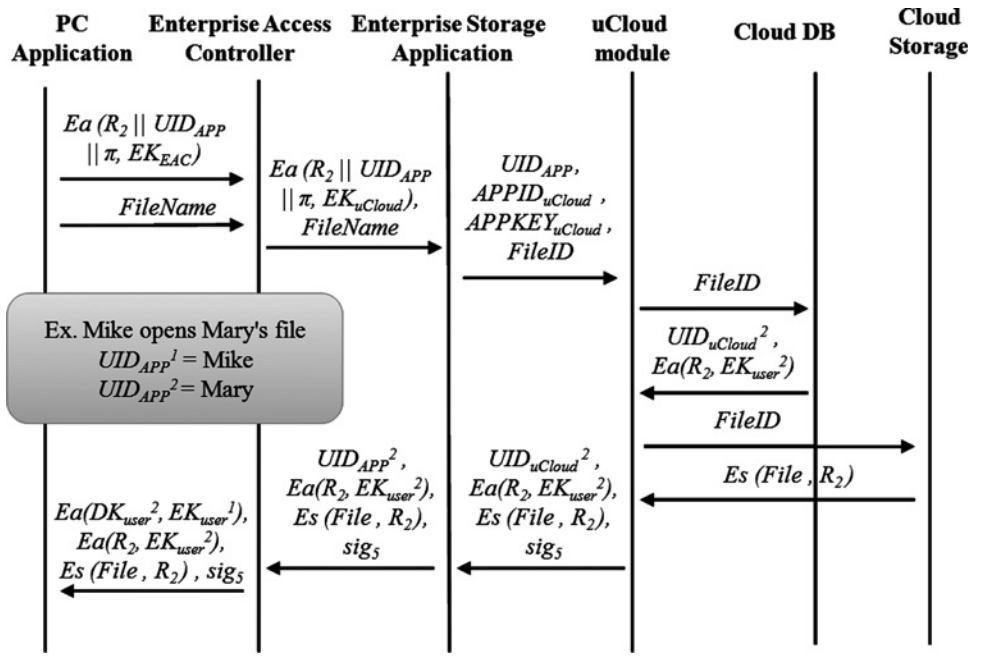


Fig. 11 *Sequence diagram of file sharing via EAC*

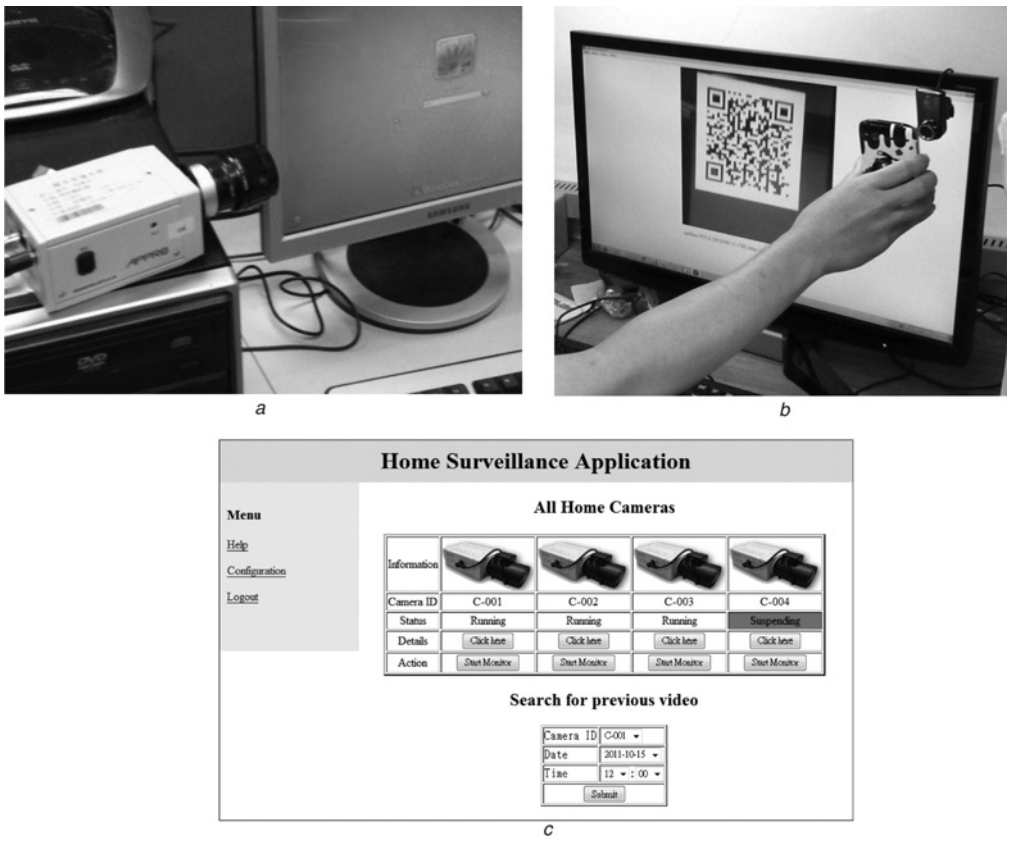


Fig. 12 *IP camera used*

a IP camera used in the home surveillance scenario
b Constrained channel construction by showing QR code image on mobile phone
c Screenshot of Home Surveillance Application

is compromised, the protected data are leaked. Finally, although the delegatees have the private keys of delegators, they cannot access to all the files of delegator without the grants of delegator. The reason is that the delegatee and public cloud operator do not collude (as defined in assumptions); therefore the access control and file decryption operations are separated.

As the messages transferred between different parties of PC application, cloud service and EAC, since all the messages and data are properly encrypted by public keys or session

doi: 10.1049/iet-ifs.2012.0198

Table 2 System comparison

	User-controlled decryption	Data sharing	Flexible Access Control	No ACM maintained by user	Private key stored on mobile phone
Traditional cloud services	no	yes	yes	yes	no
System designed by Curino *et al.* [11]	yes	no	no	yes	no
System designed by Kamara and Lauter [12]	yes	yes	no	yes	no
System designed by Dai and Zhou [13]	yes	yes	yes	no	no
System designed by Sanka *et al.* [14]	yes	yes	yes	yes	no
uCloud	yes	yes	yes	yes	yes

keys, attackers cannot extract the protected information. Also, the channel established between mobile and PC application is a constrained channel; 2D barcode images are difficult to be eavesdropped. Finally, almost all the returned messages are properly signed, so attackers cannot forge or modify them without having the private keys.

6 System prototype and evaluation

6.1 System prototype

To demonstrate the proposed system, the uCloud module and the personal storage, home surveillance and enterprise storage scenarios are implemented in the uCloud prototype. In the uCloud prototype, the mobile application is deployed on HTC G1. In the PC application, the data protector, RTSP Proxy and media player are implemented in C++; the EAC client and IP camera initialiser are implemented in Java. In the EAC, the user authenticator and decision maker are implemented in JSP and deployed on Tomcat server, and the key storage and access control rules are stored in MySQL server. The IP camera application is implemented in C and deployed on the IP camera DM368IPNC-MT5. Finally, on the cloud side, the Personal Storage Application, Home Surveillance Application, and Enterprise Storage Application and the uCloud module are implemented in JSP, Java and C++, and deployed on the Hadoop platform with HBase DB.

Fig. 12*a* shows the IP camera used in the home surveillance scenario. After the mobile device sends the session key to IP camera, the status of camera is changed to 'Running' on the Home Surveillance Application Webpage as shown in Fig. 12*c*. If the 'Start Monitor' button is clicked, the Matrix Code Decoder is automatically invoked. Then, as shown in Fig.re 12*b*, users can display QR code image containing the private key in front of the Webcam of PC. After the private key is obtained, it can be used to decrypt streaming video for users to monitor. The implementations of Personal Storage and Enterprise Storage Applications are similar with the Home Surveillance Application except the design of IP camera.

6.2 System comparison

Table 2 lists the system comparison between uCloud and related systems. Traditional cloud services such as Google Gmail provides plenty of functionalities but does not support the user-centric data protection mechanism. Curino *et al.*'s system includes a client-side application to encrypt and decrypt data, but does not support the data sharing functionality. Kamara and Lauter's system supports data sharing between owner and other users, but their access control mechanism is inflexible. Dai and Zhou's system allows owners to upload their ACMs, but owners have to define and maintain them. Finally, only uCloud stores users' private keys on their mobile phones to provide the mobility of key management and make users feel safe.

7 Conclusion and future works

7.1 Conclusion

In conclusion, we propose the uCloud to provide user-centric key management of cloud data protection, which includes a hierarchical structure for basic key backup and data sharing, and the EAC server to extent the capability of complex access control. In uCloud, the private keys are stored on users' mobile devices and presented via 2D barcode images when they are utilised to decrypt users' data. In this manner, even if the cloud services are compromised, the data are still safely protected. Therefore personal, family and enterprise users can feel save to upload their sensitive data up to cloud. Finally, the uCloud prototype including the uCloud module, these three scenarios, and EAC, is implemented and evaluated to show that it is convenient to be used.

7.2 Future works

Currently, since users usually do not know the security level of cloud services, we assume that all cloud services are compromisable. However, more secure designs cause less security problems. In the future, we will use the Trusted Platform Module (TPM) [28] hardware to enhance the security level of uCloud module. TPM can protect sensitive data on laptops simply by sealing the data with a non-migratable TPM storage key. TPM-based hypervisor [29], such as TrustVisor [30] can be utilised to provide code integrity, data integrity and data secrecy for uCloud module. In this manner, the execution of uCloud module can be isolated from malware. Moreover, the algorithm used for streaming video encryption will be improved to provide more efficient performance. For example, for H.264 video, if only the key frames are encrypted, the computation load can be decreased. Finally, since the proposed scheme mainly focuses on usability, therefore it only provides basic protection of user data. Several encryption technologies such as attribute-based encryption [31] and proxy re-encryption [32] will be integrated into uCloud for fine-grained access control and computation aggregation.

8 Acknowledgment

This research was supported by Information and Communication Research Laboratories, Industrial Technology Research Institute (ITRI), Taiwan, Republic of China under project code C352SN1200.

9 References

1 Vouk, M.A.: 'Cloud computing – issues, research and implementations', *J. Comput. Inf. Technol*., 2008, **16**, (4), pp. 235–246
2 Weiss, A.: 'Computing in the clouds', *NetWorker*, 2007, **11**, (4), pp. 16–25
3 Wang, L., Laszewski, G., Kunze, M., Tao, J.: 'Cloud computing: a perspective study', *New Gener. Comput*., 2010, **28**, pp. 137–146
4 Heiser, J., Nicolett, M.: 'Assessing the security risks of cloud computing'. Gartner, Incorporated, 3 June 2008
5 Kandukuri, B.R., Paturi, R., Rakshit, A.: 'Cloud security issues'. Proc. Working IEEE SCC 2009: Int. Conf. Services Computing 2009 (SCC 2009 WIP), 2009
6 Chow, R., Golle, P., Jakobsson, M., *et al*: 'Controlling data in the cloud: outsourcing computation without outsourcing control'. Proc. 2009 ACM workshop on Cloud Computing Security, 13 November 2009
7 Kornowski, R., Zeeli, D., Averbuch, M.: 'Intensive home-care surveillance prevents hospitalization and improves morbidity rates among elderly patients with severe congestive heart failure', *Am. Heart J*., 1995, **129**, pp. 762–766
8 Kapoor, B., Chhabra, A.: 'Dynamic probe window based optimization for surveillance in home security system', *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)*, 2010, **2**, (1), pp. 106–114
9 Lu, M.T., Yao, J.J., Chen, H.H.: 'A complexity-aware video adaptation mechanism for live streaming systems', *EURASIP J. Adv. Signal Process*., 2007, **2007**, pp. 1–10
10 Kannan, S., Gavrilovska, A., Schwan, K.: 'Cloud4Home – enhancing data services with @Home clouds'. Int. Conf. Distributed Computing Systems (ICDCS 2011), 2011
11 Curino, C., Jones, E.P.C., Popa, R.A., *et al*: 'Relational cloud: a database-as-a-service for the cloud'. Proc. Fifth Biennial Conf. Innovative Data Systems Research (CIDR'11|), 2011
12 Kamara, S., Lauter, K.: 'Cryptographic cloud storage'. ACM Workshop on Cloud Security, 2009
13 Dai, L., Zhou, Q.: 'A PKI-based mechanism for secure and efficient access to outsourced data'. Proc. Second Int. Conf. Networking and Digital Society (ICNDS), June 2010, (1), pp. 640
14 Sanka, S., Hota, C., Rajarajan, M.: 'Secure data access in cloud computing'. IEEE Fourth Int. Conf. Internet Multimedia Systems Architectures and Applications (IMSAA 2010), Bangalore, December 2010
15 Corner, M.D., Noble, B.D.: 'Zero-interaction authentication'. Proc. Eighth Annual Int. Conf. Mobile Computing and Networking (MobiCom '02), 2002
16 Finkenzeller, K.: 'RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication' (Wiley & Sons LTD, 2010, 3rd edn.)
17 Haselsteiner, E., Breitfuss, K.: 'Security in near field communication (NFC) strengths and weaknesses'. Workshop on RFID Security, 2006
18 Michahelles, F., Thiesse, F., Schmidt, A., Williams, J.R.: 'Pervasive RFID and near field communication technology', *IEEE Pervasive Comput*., 2007, **6**, (3), **39**, pp. 2–5
19 Ottaviani, E., Pava, A., Bottazi, M., Brunclli, E., Casclli, F., Guerreo, M.: 'A common image processing framework for 2D barcode reading'. Proc. Seventh Int. Conf. Image Processing and its Applications, 1999, pp. 652–655
20 Gao, J.Z., Prakash, L., Jagatesan, R.: 'Understanding 2D-BarCode technology and applications in M-commerce – design and implementation of a 2D barcode processing solution'. Proc. 31st Int. Conf. Annual International Computer Software and Applications, 2007, pp. 49–56
21 Kato, H., Tan, K.T.: '2D barcodes for mobile phones'. Proc. Second Int. Conf. Mobile Technology, Applications and Systems, 2005, p 8
22 Seino, K., Kuwabara, S., Mikami, S., *et al*: 'Development of the traceability system which secures the safety of fishery products using the QR code and a digital signature'. Proc. MTS/IEEE TECHNO-OCEAN, Kobe, 2004, vol. 1, pp. 476–481
23 Ohbuchi, E., Hanaizumi, H., Hock, L.A.: 'Barcode readers using the camera device in mobile phones'. IEEE Int. Conf. Cyberworlds (CW04), 2004
24 Chaisatien, P., Akahori, K.: 'Introducing QR code in classroom management and communication via mobile phone application system'. Proc. World Conf. Educational Multimedia, Hypermedia and Telecommunications, 2006, pp. 2181–2187
25 Arnes, A., Knapskog, S.J.: 'Selecting revocation solutions for PKI'. Proc. Fifth Nordic Workshop on Secure IT Systems (NORDSEC 2000), 2000
26 Khurana, H., Gligor, V.D.: 'Review and revocation of access privileges distributed with PKI certificates'. Proc. Security Protocols Workshop, 2002, pp. 100–112
27 Critchlow, D., Zhang, N.: 'Revocation invocation for accountable anonymous PKI certificate trees'. Proc. Ninth IEEE Symp. Computers and Communications (ISCC'2004), 2004, pp. 386–392
28 Bajikar, S.: 'Trusted Platform Module (TPM) based security on notebook PCs-white paper' (Mobile Platforms Group Intel Corporation, 2002), pp. 1–20
29 Bressoud, T.C., Schneider, F.B.: 'Hypervisor-based fault-tolerance'. Proc. Symp. Operating Systems Principles, 1995, pp. 1–11
30 McCune, J.M., Li, Y., Qu, N., *et al*: 'TrustVisor: efficient TCB reduction and attestation'. Proc. IEEE Symp. Security and Privacy, Oakland, 2010
31 Goyal, V., Pandey, O., Sahai, A., Waters, B.: 'Attribute-based encryption for fine-grained access control of encrypted data'. Proc. CCS'06, 2006
32 Blaze, M., Bleumer, G., Strauss, M.: 'Divertible protocols and atomic proxy cryptography'. Proc. EUROCRYPT '98, 1998

doi: 10.1049/iet-ifs.2012.0198