Secure Sharing and Searching for Real-Time Video Data in Mobile Cloud

Joseph K. Liu, Man Ho Au, Willy Susilo, Kaitai Liang, Rongxing Lu, and Bala Srinivasan

Abstract

In this article we propose an infrastructure that allows mobile users to securely share and search for their real-time video data. Specifically, the proposed infrastructure takes the advantages of the cloud platform and 5G technology to achieve its goals, where mobile users (connected with some external video taking device) can share their real-time video with their friends or families through the cloud while any other user with no permission cannot get any information about the video. More importantly, the infrastructure security is guaranteed even if the cloud server is hacked. In addition, our infrastructure also allows secure searching within the user's own video data. We believe our solution is practical to be deployed in the existing telecommunication platforms.

The ubiquitousness of mobile applications and devices has been part of the new market that enables mobile developers to provide new services for their users. The success of mobile applications is also driven by data feeds and services in the cloud, and hence it leads to the notion of mobile cloud computing [1]. The demand for the transfer of huge amounts of data will need to be supported by rapid data transfer, which will make the application very usable and hence, enhance the users' experience.

The next major phase of the mobile telecommunication standard, known as 5G, will allow larger bandwidth. Ericsson [2] had conducted a test that achieved a connection speed of 5 Gb/s over the air which is part of its plans for 5G wireless technology. Recently, Samsung [3] also created a record for transmitting data over a 28 GHz 5G network at a speed of 7.5 Gb/s in a stationary environment, achieving a speed record using the future mobile standard. At the same time it has also managed to achieve an uninterrupted, stable connection of 1.2 Gb/s in a vehicle travelling at 100 km/h.

With the availability of 5G technology, sharing real-time high density video can be offered efficiently via the cloud platform. A 50 GB movie would take less than two minutes to download using a 5G connection, which is 250 times faster than today's standard 4G LTE network. With the lower latency and higher system spectral efficiency, 5G enables users to store and retrieve a high volume of real time video efficiently.

When incorporating the cloud, major security issues will arise, such as the leak of video data. Merely encrypting the video data in the cloud will not be a viable solution since many operations (such as data search) on the data will be crippled. Hence, a novel way to provide data protection has become necessary in this situation.

In this work we consider a futuristic, yet very practical and realistic, scenario as follows. Bob, who is a U.S. senator, is taking a vacation with his work colleagues. But unfortunately, since he has a young family member, they do not join him on the trip. Nevertheless, Bob would like to share his experiences with his family members. To do so, Bob is equipped with a high definition and versatile camera, which he uses throughout his trip (during skiing, diving, etc.). After the scene has been recorded, it will be uploaded to the cloud via the 5G in an encrypted format, and hence having a high definition video is possible. Bob's family members, having access to the right applications and the credentials, can access within the encrypted video collections provided by Bob. As the video scenes contain some parts that Bob does not want to share with any outsiders, leaking even the keywords will be disastrous, due to Bob's reputation in the public.

Our Contribution

In this article, we propose an infrastructure that allows mobile users to securely share and search for real-time video data. Mobile users can choose the set of people with whom they want to share (e.g. friends, family members). Users outside this set cannot obtain permission to access the file, or even receive any information (e.g. keywords) about the video data. We deploy cloud technology as the basis with some cryptographic primitives as the building blocks. Therefore, our security is guaranteed even if the cloud server is hacked or the video data is stolen. In addition, we also provide secure searching within a user's own video data. Our solution provides efficient and practical real-time video sharing and searching among mobile users by utilizing 5G technology in the cloud computing paradigm.

Existing Platforms for Real-Time Video

Existing cloud infrastructure allows people to store their files at an affordable price or for free. The list includes: Dropbox, Box, Justcloud, Baidu pan, and Google drive, among others.

Joseph K. Liu is with Monash University and Shenzhen University.

Man Ho Au is with The Hong Kong Polytechnic University.

Willy Susilo is with the University of Wollongong.

Kaitai Liang is with Aalto University.

Rongxing Lu is with Nanyang Technological University.

Bala Srinivasan is with Monash University.

0890-8044/15/$25.00 © 2015 IEEE

All of them allow their users to specify files for sharing. Some of them allow users to make their files publicly available. Service providers specializing in media sharing include: Youtube, Vimeo for video, Flickr, and Photobucket for photos. Table 1 summarizes these service providers with their web addresses.

Security of the stored content depends on the policy of the provider. For instance, the statement from Dropbox assured that 256-bit AES was employed to ensure the file's security. Nonetheless, it is also written that

> *"We have a small number of employees who must be able to access user data for the reasons stated in our privacy policy."*

In a nutshell, access control of the user's files is maintained by the service provider and they are trusted to carry their duty. Sharing is conducted at the file level using a white list approach. For each file or directory, the owner can specify a list of users who have right to access. The owner can also choose to make the file publicly available.

Searching is often supported at the global scope. For example, users can search from the videos available on Youtube. For Dropbox, how a file can be searched within the collection of the user is outside the service model of Dropbox. However, the videos made available for searching are not encrypted.

More on Skyfire[1]

Skyfire Rocket Optimizer (www.skyfire.com) is a cloud-based mobile video and media optimization technology. The target user of this optimizer is the mobile operator (e.g. 3, Vodafone etc.). The consumers of this operator consume a lot of bandwidth with the growing popularity of mobile video. To save bandwidth, the mobile operator could leverage video compression. This is the traditional approach. Skyfire claims to provide a new solution, that can measure the experience for the users. For example, if a user is at the edge of the cellular base station, video sent to that user should be optimized aggressively since the bandwidth is going to be limited. On the other hand, when the bandwidth is more than sufficient, there is not a pressing need for video compression for that user. By continuously monitoring the users and the network situation, Skyfire Rocket Optimizer can also apply the "transcoding" for the needed users. By doing so, they claim to allow mobile operators to guarantee the best user experience for their consumers.

Challenges in Existing Platforms

Although there are some existing platforms for sharing real-time video, they may not be able to achieve secure fine-grained sharing and secure searching simultaneously. These two important functions are very important to users who deal with large volume of data (e.g. large video), which will emerge in the 5G era. Thus we need to have a new infrastructure to provide secure sharing and searching for large real-time data (such as video).

Our Proposed Infrastructure

Network Infrastructure Overview

We first give an overview of our network infrastructure, as illustrated in Fig. 1, where a mobile user is connected with an external video-taking device (e.g. GoPro (gopro.com)) through WiFi, and the mobile device is connected through 5G with a cloud server with purposes of storage and sharing. Our security mechanisms will be built on top of this network infrastructure, which will be described later in this section.

Index	Name of service provider	URL of service provider
1	Dropbox	https://www.dropbox.com/
2	box	https://www.box.com/
3	justcloud	http://www.justcloud.com/
4	Baidu pan	http://pan.baidu.com/
5	Google drive	https://drive.google.com/
6	Youtube	http://www.youtube.com/
7	Vimeo	http://vimeo.com/
8	flickr	http://www.flickr.com/
9	photobucket	http://www.photobucket.com/

Table 1. Cloud storage service providers.

Cryptographic Functions Overview

We give a brief description of some cryptographic functions that are deployed in our infrastructure.

Advanced Encryption Standard (AES) [4, 5]: This is the most commonly used symmetric encryption scheme. In an AES encryption system, a user first generates a key *AES.key* (which is used to encrypt or decrypt a message), and next runs an AES encryption algorithm $AES.C \leftarrow AES.Enc(AES.key, m)$ with the key *AES.key* to encrypt a message m and get a ciphertext *AES.C*. By using the same key, the user can recover the message from its encrypted format via a decryption algorithm $m \leftarrow AES.Dec(AES.key, AES.C)$.

Searchable Symmetric Encryption (SSE) (e.g. [6–8]): In a SSE system, a user is allowed to generate a key *SSE.key* for both the encryption and the decryption of a message. By using the key, the user can encrypt a keyword index I via an encryption algorithm $SSE.C \leftarrow SSE.Enc(SSE.key, I)$, and next upload the encrypted keyword index to a storage server. In the searching phase, the user (using the knowledge of the key) delivers a searchable trapdoor token $t_w \leftarrow Trpdr(SSE.key, w)$ associated with the keyword w to the server. The server checks this token with every ciphertext $Check(SSE.C, t_w)$. If the ciphertext is the encryption of the keyword w, it returns true. The user can also run a decryption algorithm with the key to decrypt the whole ciphertext.

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) (e.g. [9, 10]): CP-ABE is a kind of asymmetric encryption. In a CP-ABE scheme, a registered user is first issued a decryption key sk_{AS} from a trusted ABE key generation center (KGC), in which the key is associated with an attribute set AS describing the user (e.g. "`male`," "`student`"). To share data with other system users, the user is required to encrypt a message m under a specified access policy `policy` (e.g. male student) via an encryption algorithm $ABE.C \leftarrow ABE.Enc(\texttt{policy}, m)$. If the attribute set of a user's decryption key satisfies the above access policy, this user then is able to gain access to the data by running a decryption algorithm with the key $m \leftarrow ABE.Dec(ABE.C, sk_{AS})$.

Digital Signature (e.g. [11, 12]): In a digital signature system, a registered user is issued a signing/verification key pair (*ssk*, *svk*) with a corresponding certificate *cert* (which is used to guarantee the validity of the verification key and the identity

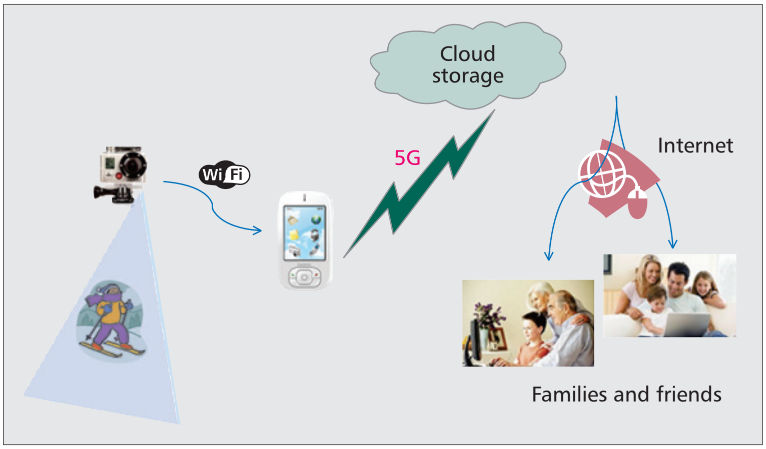

Figure 1. Our network infrastructure overview (security mechanisms will be built on top of this network infrastructure).

of the user) issued by a certificate authority (CA). The user is able to use the signing key to make a digital signature $s \leftarrow Sign_{ssk}(m)$ on a message m such that anyone holding the corresponding verification key and the message is able to verify the validity of the signature $Verify_{svk}(\sigma, m)$.

Detailed Description

There are three parties in our proposed infrastructure: the mobile user (with a 5G-connected mobile device) who can upload video to the cloud with an external video-taking device; the cloud server; and the normal user (who may use a normal PC computer or mobile device but cannot upload video). In addition there are two authorities: the key generation centre (KGC) for issuing the attribute-based user secret key, and the certificate authority (CA) for issuing the user certificate. There are several protocols in our infrastructure, as described below.

System Setup: The user with a mobile device downloads an app that is equipped with cryptographic functions such as AES encryption, searchable symmetric encryption (SSE), ciphertext-policy attribute-based encryption (ABE), and digital signature.

User Registration: User registration consists of two parts. In the first part, the user registers with a trusted ABE key generation centre (KGC) to obtain their attribute user secret key which is used for video sharing purposes. In the second part, the user registers with the cloud server for access control purposes. They are described as follows:

- A user (mobile user or normal user) first registers to a trusted ABE KGC, which in turn offers an attribute set AS describing him/herself (e.g. "`male`," "`student`," "`British`," "`Alice's friend`," "`Bob's family`") and the corresponding decryption key sk_{AS} associated with the attribute set to the user. The user stores the decryption key.
- A mobile user further registers him/herself with the cloud. The detail procedures are described as follows:
  - – A mobile user obtains a signing/verification key pair (ssk_{reg}, svk_{reg}) and a digital certificate *cert* (which contains the user identification information and the verification key) from the CA. The user signs the username `USERNAME` and the login password `PASSWORD` as $\sigma_{reg} \leftarrow Sign_{ssk_{reg}}$(`USERNAME`||$H$(`PASSWORD`)),and finally uploads the tuple (`USERNAME`, H(`PASSWORD`), σ_{reg}, *cert*) to the cloud server, where H is a target collision resistant hash function.
  - –The cloud server validates *cert*, extracts the svk_{reg} from the *cert*, and verifies the signature as $Verify_{svk_{reg}}$(σ_{reg}, (`USERNAME`||H(`PASSWORD`)). If all verifications are valid, the server stores the tuple (`USERNAME`, H(`PASSWORD`), *cert*) in its back end storage system.
  - –Next the user generates a SSE encryption/decryption key as *SSE.key* and stores *SSE.key* in the mobile device.
- At the end of the registration, every user (mobile or normal) gets his/her attribute-based decryption key sk_{AS} (with the associated attributes). A mobile user (who can upload video to the cloud) additionally gets a certificate *cert* (together with a signing/verification key pair), a searchable encryption/ decryption key *SSE.key*, and a username/password for the access to the cloud.

  The protocol is illustrated in Fig. 2a.

Video Upload — After using an external camera device to take a video, it is transferred to the user's mobile device via WiFi. The user further remarks the video by using some keywords as searchable indexes (e.g. date, location information, personal identification etc.). Before uploading the video to the cloud, the mobile device needs to encrypt the video in several layers. First, it uses AES to encrypt the video data. Then it uses SSE to encrypt the corresponding keywords. Finally, it uses ABE to encrypt the AES key under some desired attributes (e.g. "`Alice's family`"). The details of the protocol are described below.

- In order to encrypt a video data V, the user first generates a one-time AES encryption/decryption key as $AES.key_V$, and uses this key to symmetric encrypt the video data V as $AES.CV \leftarrow AES.Enc(AES.key_V, V)$.
- Next the user uses *SSE.key* (generated at the registration stage) to encrypt the corresponding keyword index $I_V = \{I_1, \dots, I_n\}$ as $SSE.C_V = SSE.Enc(SSE.key, I_V)$.
- In order to securely share the encrypted data with other users, the mobile user deploys ABE to encrypt the one-time AES key $AES.key_V$ as $ABE.C_V \leftarrow ABE.Enc$(`policy`, $AES.key_V$), where `policy` is an access control formula. `policy` always includes {OR "`USER=USERNAME`"}. For example, if the user has a username `Alice`, the policy can be set as "`USER=Alice`" OR "`Alice's family`" (that means user `Alice` or any user with the attribute `Alice's family` can decrypt). In this way, the user does not need to store any AES symmetric key as he/she can always retrieve every key from decrypting the attribute-based ciphertext from the cloud.
- The user signs the above encryptions as s_V ¨ $Sign_{ssk_{reg}}(AES.C_V||SSE.C_V||ABE.C_V)$.
- The user eventually logs in to the cloud by using his/her username and password, and uploads the tuple ($AES.C_V$, $SSE.C_V$, $ABE.C_V$, σ_V).
- If the verification on $Verify_{svk_{reg}}(AES.C_V||SSE.C_V||ABE.C_V$, σ_V) is valid, the cloud stores the tuple in the storage under the username `USERNAME` of the user. There is also a sequence number d for this entry. That is, this is the d-th video of the user whose username is `USERNAME`. The data structure of this storage is shown in Table 2.

  The protocol is illustrated in Fig. 2b.

Video Searching and Retrieval: In order to search for and retrieve a particular video (with a keyword), the video owner proceeds as follows.

- The user logs in to the cloud system by using the username/password.
- The user retrieves the key *SSE.key* from his/her mobile phone and generates the searchable trapdoor token as $t \leftarrow SSE.Tpdr(SSE.key, w)$ for a keyword w. He/she also uploads the token to the cloud server.
- The cloud server searches all $SSE.C_V$ for this user (with username `USERNAME`). If there is a match for the keyword w (by executing the algorithm $Check(SSE.C_V, t)$), the cloud

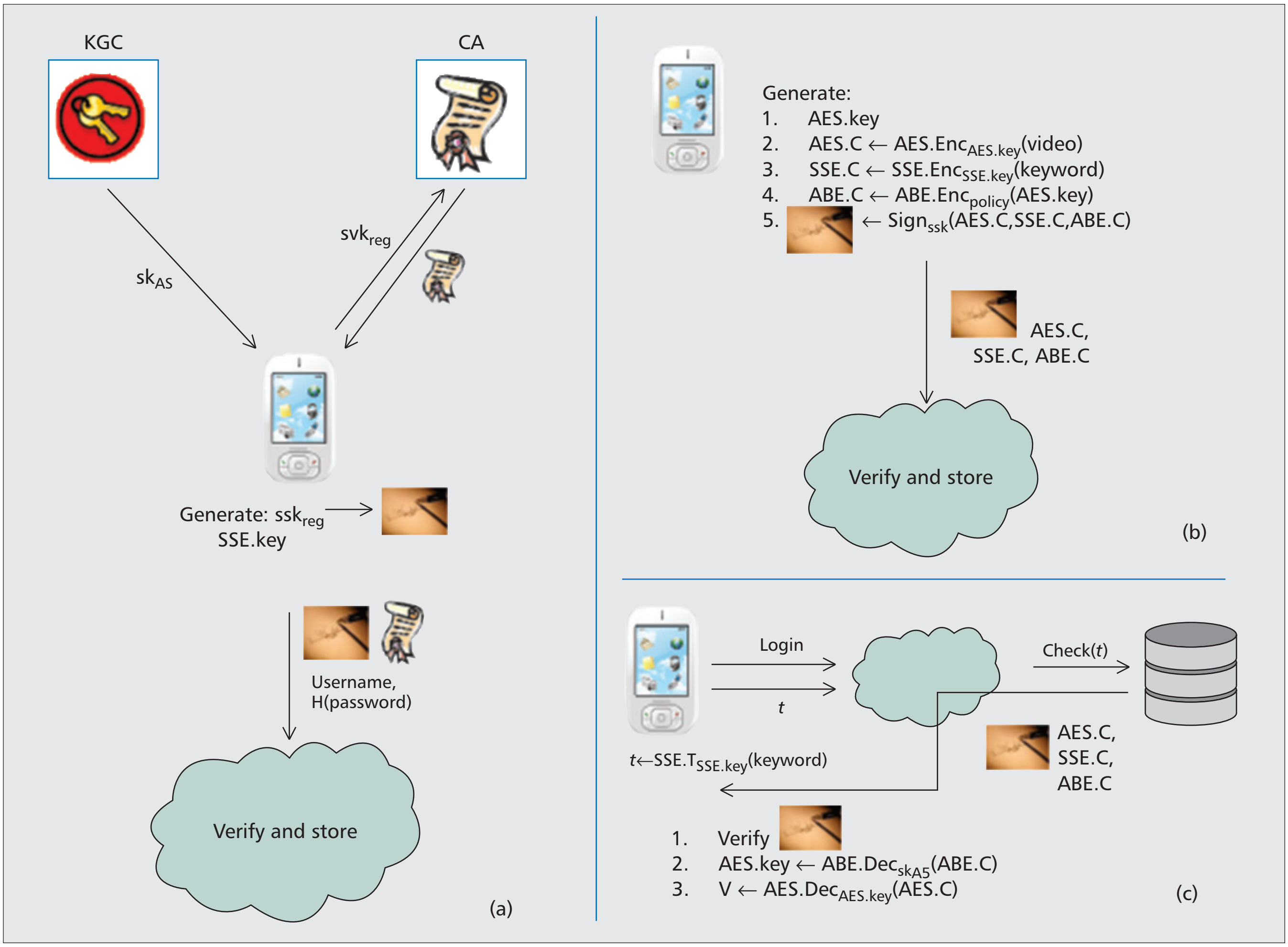
KGC
CA
sk_{AS}
svk_{reg}
Generate: ssk_{reg}
SSE.key
Username,
H(password)
Verify and store
(a)
Generate:
1. AES.key
2. AES.C ← $AES.Enc_{AES.key}$(video)
3. SSE.C ← $SSE.Enc_{SSE.key}$(keyword)
4. ABE.C ← $ABE.Enc_{policy}$(AES.key)
5. ← $Sign_{ssk}$(AES.C,SSE.C,ABE.C)
AES.C,
SSE.C, ABE.C
Verify and store
(b)
Login
t
Check(t)
t←$SSE.T_{SSE.key}$(keyword)
AES.C,
SSE.C,
ABE.C
1. Verify
2. AES.key ← $ABE.Dec_{skA5}$(ABE.C)
3. V ← $AES.Dec_{AES.key}$(AES.C)
(c)

Figure 2. a) User Registration Protocol; b) Video Upload Protocol; c) Video Searching and Retrieval Protocol.

looks up the corresponding sequence number d for this entry and returns the corresponding tuple ($AES.C_V$, $SSE.C_V$, $ABE.C_V$, σ_V).

- The user verifies the signature $Verify_{svk_{reg}}$ ($AES.C_V || SSE.C_V || ABE.C_V$, σ_V). If it is valid, he/she uses sk_{AS} to decrypt $ABE.C_V$ and gets $AES.key_V \leftarrow ABE.Dec(sk_{AS}, ABE.C_V)$. Then he/she uses $AES.key_V$ to decrypt $AES.C_V$ and gets $V \leftarrow AES.Dec(AES.key_V, AES.C_V)$.

The protocol is illustrated in Fig. 2c.

Video Sharing: If the video owner wants to share one of his/her videos with their friends or another set of people (with some unique attributes), they can do the following:

- The video owner searches for the video entry V that he/she wants to share (by using **Video Searching and Retrieval**).
- The user asks the cloud to open the corresponding web page for public access such that the web page contains URLs for the download of the tuple ($AES.C_V$, $SSE.C_V$, $ABE.C_V$, σ_V) and *cert* of the user.[1]
- Those users with whom the video owner wants to share can access the web page to download all tuples and *cert* and execute the following steps to retrieve the video:
  - – Verify *cert* and extract the public key svk_{reg}.
  - – If the verification on $Verify_{svk_{reg}}(AES.C_V || SSE.C_V || ABE.C_V, \sigma_V)$ is valid, use sk_{AS} to decrypt $ABE.C_V$ to get $AES.key \leftarrow ABE.Dec(ABE.C_V)$.
  - – Use $AES.key$ to decrypt $AES.C_V$ to get $V \leftarrow AES.Dec(AES.C_V)$.

Sequence number	Encrypted video data (using AES)	Encrypted keywords (using SSE)	Encrypted AES Key (using ABE)	Signature
:	:	:	:	:
d	$AES.C_V$	$SSE.C_V$	$ABE.C_V$	σ_V
:	:	:	:	:

Table 2. Data structure for an entry of video data of the user whose username is USERNAME.

Note that currently the list of recipients is defined in the ciphertext. It can be changed dynamically by using proxy re-encryption technique [13].

Efficiency Analysis

We analyze the efficiency of our protocol. We use the benchmark result from Geekbench 3 (http://www.primatelabs.com/geekbench/) and the API from JPBC (http://gas.dia.unisa.it/projects/jpbc/docs/android.html) for simulation. The mobile

[1] Note that all these data are in encrypted format. Anyone who does not have the corresponding decryption key cannot get any information about the underlying plaintext (the video) even they have downloaded the tuple.

device we used is HTC M8. We deploy the signature scheme from [14] and the ABE scheme from [15] for the analysis.

In our protocol, the cloud server is only required to verify a digital signature in each phase. It can be done in 0.02 seconds according to the benchmark result from JPBC (http://gas.dia.unisa.it/projects/jpbc/benchmark.html) for a desktop computer with Intel$^{(R)}$ Core$^{(TM)}$2 Quad CPU Q6600 @ 2.40GHz, 3 GB Ram, Ubuntu 10.04.

For the user side, we analyze the efficiency for user registration, video upload, and video retrieval phases. In the user registration phase, the mobile device requires 0.151 second. In the video upload phase, we deploy AES CBC mode so the streaming video can be processed. The maximum processing speed for AES is 293.9 MByte/s, which is much larger than the video bitrate (the maximum bitrate for 1080p Blu-ray Disc is just 40 Mb/s). In other words, the streaming video generated can be encrypted using AES immediately without any delay. That is, when the whole video has been generated, the AES encryption will be almost done. The time for SSE is negligible when compared to ABE. Thus we only consider the time for ABE encryption. Assume the policy contains four attributes. The ABE encryption takes 0.14 second. It further requires a signature generation with the entire video data. Assume the video data size is 2 Gbytes. It takes 2.97 seconds to generate the hash using SHA-1 and 0.15 second to generate the signature. Overall, it takes less than 4 seconds to complete.

In the video retrieval phase, again we assume the video data size is 2 G and the policy contains four attributes. It takes 2.97 seconds to generate the hash using SHA-1; 0.71 seconds to verify the signature; 4.5 seconds to ABE decrypt; and 6.97 seconds to AES decrypt. Overall, it takes less than 16 seconds to decrypt the video.

Conclusions

In this article we have proposed an infrastructure for secure sharing and searching for real-time video data. It is particularly suitable for mobile users by deploying 5G technology and a cloud computing platform. Our security is guaranteed even if the cloud server is hacked since data confidentiality is now protected by cryptographic encryption algorithms. In addition, we also provide secure searching functionality within a user's own video data. We believe our proposed infrastructure is practical to be deployed.

Acknowledgments

W. Susilo is partially supported by the Australian Research Council Discovery Project ARC DP130101383. K. Liang is supported by privacy-aware retrieval and modelling of genomic data (PRIGENDA, No. 13283250), the Academy of Finland, Finland. R. Lu is supported by Nanyang Technological University under Grant NTUSUG (M4081196) and MOE Tier 1 (M4011177).

References

[1] Smith's Point Analytics, "Mobile Cloud Platforms: The Backend of Mobile Apps," http://www.reportlinker.com/p01650001-summary/Mobile-Cloud-Platforms-The-Bac kend-of-Mobile-Apps.html, 2013.

[2] CNET, "Ericsson Hits Crazy-Fast 5Gb/s Wireless Speed in 5G Trial," http://www.cnet.com/news/ericsson-tests-out-crazy-fast-5-gbps-wireless-speed /, July 2014.

[3] Computer Weekly, "Samsung Claims 5G Speed Record," http://www.computerweekly.com/news/2240232676/Samsung-claims-5G-speed-record, Oct. 2014.

[4] United States National Institute of Standards and Technology (NIST), "Announcing the Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication 197, 2001.

[5] A. Alahmadi *et al.*, "Defense Against Primary User Emulation Attacks in Cognitive Radio Networks Using Advanced Encryption Standard," *IEEE Trans. Inf. Forens. Security*, vol. 9, no. 5, 2014, pp. 772–81.

[6] R. Curtmola *et al.*, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," *ACM Conf. Computer Communications Security*, A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds., ACM, 2006, pp. 79–88.

[7] D. Cash *et al.*, "Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries," *CRYPTO 2013*, ser. Lecture Notes in Computer Science, vol. 8042, Springer, 2013, pp. 353–73.

[8] D. Cash and S. Tessaro, "The Locality of Searchable Symmetric Encryption," *Proc. EUROCRYPT 2014*, ser. Lecture Notes in Computer Science, vol. 8441, Springer, 2014, pp. 351–68.

[9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," *IEEE Symposium on Security and Privacy*, 2007, pp. 321–34.

[10] F. Guo *et al.*, "CP-ABE with Constant-Size Keys for Lightweight Devices," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, 2014, pp. 763–71.

[11] C. P. Schnorr, "Efficient Signature Generation by Smart Cards," *J. Cryptology*, vol. 4, no. 3, 1991, pp. 161–74.

[12] L. Chen and J. Li, "Flexible and Scalable Digital Signatures in TPM 2.0," *Proc. ACM Conference on Computer and Communications Security*, ACM, 2013, pp. 37–48.

[13] K. Liang *et al.*, "A DFA-Based Functional Proxy Re-Encryption Scheme for Secure Public Cloud Data Sharing," *IEEE Trans. Information Forensics Security*, vol. 9, no. 10, 2014, pp. 1667–80.

[14] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," *J. Cryptology*, vol. 17, no. 4, 2004, pp. 297–319.

[15] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," PKC 2011, vol. 6571; Springer, 2011, pp. 53–70.

Biographies

JOSEPH K. LIU received the Ph.D. degree in information engineering from the Chinese University of Hong Kong in July 2004, specializing in cyber security, protocols for securing wireless networks, privacy, authentication, and provable security. He is now a senior lecturer at Monash University, Australia, and an adjunct associate professor at Shenzhen University, China. Prior to that he was a research scientist in the Infocomm Security Department at the Institute for Infocomm Research, Singapore from 2007-2015. His current technical focus is particularly cyber security in the cloud computing paradigm, smart city, lightweight security, and privacy enhanced technology. He has published more than 80 refereed journal and conference papers and received the Best Paper Award from ESORICS 2014. He has served as the program chair of ProvSec 2007, 2014, Pairing 2015, and on the program committees of more than 35 international conferences.

MAN HO AU obtained his bachelor's (2003) and master's (2005) degrees from the Department of Information Engineering, Chinese University of Hong Kong. He received his Ph.D. from the University of Wollongong (UOW) in 2009. Currently he is an assistant professor in the Department of Computing, Hong Kong Polytechnic University (PolyU). Before joining PolyU in July 2014 he was a lecturer at UOW. He works in the area of information security. In particular, his research interests include applying public-key cryptographic techniques to systems with security and privacy concerns. He has published over 70 referred journal and conference papers.

WILLY SUSILLO received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, Australia. He is a professor in the School of Computer Science and Software Engineering and the director of the Centre for Computer and Information Security Research, University of Wollongong. He has been awarded the prestigious ARC Future Fellow award by the Australian Research Council. His main research interests include cryptography and information security. He has served as a program committee member in major international conferences.

KAITAI LIANG received the B.Eng. degree and the M.S. degree from South China Agricultural University, China. He received the Ph.D. degree from the Department of Computer Science, City University of Hong Kong (2014). He is currently a post-doctoral researcher in Department of Information and Computer Science, Aalto University, Finland. His research interest is applied cryptography, in particular, cryptographic protocols, encryption/signature, and RFID. He is also interested in cybersecurity, such as network security, database security, and security in cloud computing.

RONGXING LU received his Ph.D. degree in computer science from Shanghai Jiao Tong University, China, in 2006, and his Ph.D. degree (awarded Canada Governor General Gold Medal) in electrical and computer engineering from the University of Waterloo, Canada, in 2012. Since May 2013 he has been an assistant professor at the School of Electrical and Electronics Engineering, Nanyang Technological University. His research interests include computer network security, mobile and wireless communication security, and big data security and privacy.

BALA SRINIVASAN received the Ph.D. degree in computer science from the Indian Institute of Technology, Kangpur, India. He is a professor of information technology with Monash University, Clayton, Australia. He has more than 30 years of experience in academia, industry, and research organizations. He has authored and jointly edited six technical books and more than 300 refereed publications in international journals and conferences in the areas of multimedia databases, data communications, data mining, and distributed systems.