Soft Comput (2017) 21:2643–2649
DOI 10.1007/s00500-015-1964-2

METHODOLOGIES AND APPLICATION

Towards secure and cost-effective fuzzy access control in mobile cloud computing

Wei Wu[1,2] · **Shun Hu**[3] · **Xu Yang**[1,2] · **Joseph K. Liu**[4] · **Man Ho Au**[5]

Published online: 11 December 2015

Abstract In this article, we suggest a secure and cost-effective fuzzy access control protocol in mobile cloud computing. It is especially designed for small and medium enterprises (SMEs) providing business-to-customers services. Our protocol allows the SME to outsource its services to a cloud to reduce the running cost. At the same time, it does not require any communication between the cloud and the SME during user authentication stage. That is, SME can be offline after users have been registered. Users directly deal with the cloud for gaining access. This helps the SME to save a lot of resources, including a large bandwidth connecting with the cloud and a strong firewall system. Meanwhile, the user database never leaves the SME. In addition, our protocol can withstand common attacks such as dictionary attacks for server and phishing attacks for client. Our security protection is especially important for mobile users as mobile devices are easily exposed to such attacks. Furthermore, our protocol provides user traceability to SME and it is very efficient for mobile devices.

Keywords Secure · Cost-effective · Fuzzy access-control · Mobile cloud · SME · B2C

Communicated by V. Loia.

W. Wu and S. Hu contributed equally to this work and should be considered as the co-first authors

✉ Wei Wu
weiwu81@gmail.com

Shun Hu
gzhushun@gmail.com

Xu Yang
yangxu9111@gmail.com

Joseph K. Liu
ksliu9@gmail.com

Man Ho Au
csallen@comp.polyu.edu.hk

1 Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China

2 State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an 710071, People's Republic of China

3 College of Information Science and Technology, Jinan University, Guangzhou 510632, People's Republic of China

4 Faculty of Information Technology, Monash University, Melbourne, Australia

5 Hong Kong Polytechnic University, Hong Kong, Hong Kong

1 Introduction

In the cloud computing era, users connecting to the Internet gain not only unlimited information of various kinds, but also an increment of computational power and data storage spaces. For example, commercial cloud computing platforms such as Amazon, Google AppEngine and Microsoft Azure are a natural fit to remedy the lack of local resources. This advantage is particularly appealing to mobile devices. Although the power of smart phones or smart devices increases rapidly in recent years, mobile applications always demand much more resources for improved interactivity of better user experience. According to a report from Smith's Point Analytics (2013) released in 2013, mobile cloud services platforms are projected to grow over the next four years from US$579 million to a staggering US$4.4 billion in 2017. As we can easily see, our world is shifting into the mobilesphere every day. Companies, especially small and medium enterprises (SME), may find more opportunities to connect with their customers in an incredibly cost-effective way through mobile connectivity. They can easily gain more

benefits from mobile cloud computing (MCC) technologies, especially for business-to-customer (B2C) business. For example, they only need to provide subscription registration and user interface for the services by outsourcing the core computation part to the cloud. This will greatly reduce the setup and running cost of any SME and increase their chance of success.

The envisioned success of MCC technologies relies not only on the mature network and communication infrastructure, but also upon the security mechanisms over these infrastructures. We can imagine if the security challenges are not well addressed, MCC would not be widely adopted. For example, a company using "Pay-as-you-go" model needs to have some kind of access control mechanisms to ensure that all connecting users have paid for their service.

In this article, we introduce a secure and cost-effective access control protocol in mobile cloud computing, which is specifically designed for SMEs providing B2C services. In our architecture, a SME providing various services to mobile users (e.g. online game, speed dating, online survey, etc.) outsources its core computation to a cloud. The SME is only responsible for user registration (and billing matters) while the access control part can be totally outsourced to the cloud. It is illustrated in Fig. 1. Within this architecture, our protocol possesses the following nice features:

1. The user database never leaves the SME. It does not need to share with the cloud any data about its subscribed users. This can highly protect the user privacy and increase user confidence.
2. There is no communication required between the SME and the cloud during user authentication process. Users directly deal with the cloud for the access. In this way, the SME does not need to rent a large bandwidth communication channel between its local server and the cloud. Furthermore, it is not necessary either to equip with a high-end and expensive security infrastructure such as a powerful firewall to detect and prevent DDoS attacks. All security measurements can be done on the cloud side.
3. Anyone who has stolen a mobile device but without knowing the corresponding password cannot gain access to the system. It can further withstand dictionary and phishing attacks against user password.
4. Our system allows the SME to provide fine-grained access control. Instead of specifying a set of particular users for the access, the SME can define an access policy (e.g. `LEVEL=VIP AND COUNTRY=US`). This can greatly increase the flexibility and reduce a lot of work load by the SME.

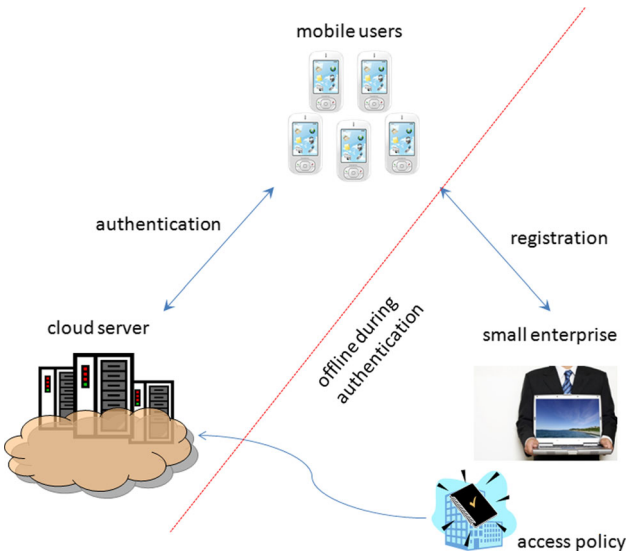

Fig. 1 The architecture of our access-control system

Paper organization. The remainder of this paper is organized as follows. In Sect. 2, we will further review some existing access control techniques and explain why they are not suitable for SME in mobile cloud computing environment before giving the details of our protocol. Section 3 describes the preliminaries required by our protocol. The details of our protocol and its analysis are given in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Existing access control techniques

In this section, we review some existing access control techniques, including simple username/password system, certificate with random challenge, and attribute-based certificate techniques.

Username/password system: This is the most simple (but efficient) access control mechanism. Each user has a unique (registered) username with a corresponding password. As a simple but representing example, the server stores the username and the hash (the output of a crytographic hash function such as SHA-1) of the password. (A crytographic hash function has the properties of one-wayness and collision resistance. One-wayness means given the output, it is computational infeasible to find its input. Collision resistance means that it is computational infeasible to find two different inputs such that their outputs are the same.) When the user types his/her username and password, the server computes the hash of the password and compares the one stored in its database. This mechanism is simple and efficient. However, it cannot withstand pre-computed dictionary attacks or rainbow table attacks (on the server side) and phishing attacks (on the user side). For the former attack (on the server side), an adversary checks the hash of the password stored on the server (we suppose the server is compromised) against its pre-computed table. If this pre-computed table is large enough,

the checking should be efficient. Although the attack may be defended by using salted hash (the hash of password plus a salted public value is stored instead of the hash of password), if the salted value is known (unfortunately it is usually the case if the adversary has broken into the server!), the adversary can still efficiently launch the aforementioned attack. For the latter attack, it is especially concerned in mobile devices, as it is easier to launch such an attack on mobile device than on normal PC computer. Once the adversary captures the username and password, it can easily pretend the victim to access the system and the user may have no way to detect it before any significant information loss occurs.

Certificate with random challenge: A digital certificate is a signature generated by a trusted party, called Certificate Authority (CA), on a user public key and his/her identification information (such as name, email, organization, etc.). This certificate and the corresponding secret key can be used in a simple authentication protocol: A user first sends the certificate to the cloud. The cloud verifies the certificate. If it is valid, it sends a random number (called the random challenge) to the user. The user signs this number using his secret key. The cloud then uses the public key (extracted from the certificate) to verify the user's signature. This classical authentication protocol can withstand phishing and replay attacks as the random number is different in each round.

However, the above techniques (username/password and certificate with random challenge) both require a three-party authentication if the user database is kept in the SME side. Since the above techniques can only prove that the user interacting with the cloud is the claimed one, it cannot let the cloud know whether this user has the right to access or use the service provided by the SME. The cloud needs to send an inquiry to the SME to ask for the status or access level of this user. It seems to be a normal action and it may not cause too much trouble to the SME. Nevertheless, if there are many users who login to the cloud at the same time, even though the cloud can handle it (as we assume the cloud should have a larger bandwidth for many users to access at the same time), the SME may not be able to respond concurrently (since the SME is not supposed to equip with a very powerful server to handle this situation). The same thing happens if the cloud is being DDoS attacked. In order to provide a normal service in any circumstance, the SME may need to have a strong firewall and large bandwidth connecting to the cloud. That will definitely increase the running cost of the SME. In order to resolve this shortcoming, an attribute-based access control is preferred instead of the classical digital certificate. We briefly introduce it below.

Attribute-based access control: An attribute-based access control is a variant of attribute-based cryptography (Sahai and Waters 2005; Goyal et al. 2006; Bethencourt et al. 2007; Ostrovsky et al. 2007; Pirretti et al. 2010; Waters 2011; Lewko and Waters 2011a, b; Maji et al. 2011; Sahai et al. 2012; Lewko and Waters 2012; Rouselakis and Waters 2013; Garg et al. 2013; Hohenberger and Waters 2013; Chen et al. 2014; Hohenberger and Waters 2014; Li et al. 2014, 2015; Rouselakis and Waters 2015; Wei et al. 2015; Liu et al. 2015; Wu et al. 2015; Xhafa et al. 2014). In attribute-based cryptography, users are represented by *attribute*. For example, a user may have the attributes `SEX=MALE`; `DEPT=PHY SICS`; `STATUS=STUDENT`; `UNIVERSITY=ABC UNI` instead of his/her real name! In each round of authentication, there is a specific policy (for example, only `MALE AND STUDENT` are allowed to use the service). Those users whose attributes satisfy the policy can go through the authentication. In the case of access control, each user has a secret key associated with his/her attributes. The user uses the attribute-associated secret key to interact with the cloud. In this way, the cloud does not need to have any communication with the SME, as the interaction between the cloud and the user can determine whether this user has the access right. This is a great improvement over the above-mentioned classical systems. Nevertheless, all attribute-based cryptographic primitives are anonymous. Privacy is good in some sense, but maybe not preferred in some cases. For instance, the SME may want to have an event log and have a detailed statistical information about each user's habit for using its services. Anonymous access control does not provide the way for tracing any user. Thus in many practical scenarios, anonymous access is not the preferred way. Furthermore, attribute-based cryptographic primitives usually require complicated and inefficient mathematical algorithms such as a number of exponentiations and pairings. Some may even require to use composite order group for pairing operation which is only of theoretical interest but contains no practical value.

From the above discussion, we can see that it seems none of the existing technologies can provide a satisfactory access control mechanism in the mobile cloud computing paradigm. We summarize the comparison among these technologies and our proposed mechanism in Table 1.

3 Preliminaries

In this section, we shall briefly describe the necessary preliminaries required by our protocol.

3.1 Existentially unforgeable digital signatures

In the public-key setting, a digital signature scheme consists of three algorithms $(KeyGen, Sign, Ver)$ associated with key generation, signing and verification, respectively. These three algorithms are defined as follows:

1. $KeyGen$: This algorithm produces a private–public key pair (sk, pk). On input a security parameter λ, this algo-

Table 1 Comparisons among different mechanisms

Mechanism	Secure against dictionary attack	Secure against phishing attack	SME offine during authentication	User traceability	Efficient
Username/password system	×	×	×	✓	✓
Certificate with random challenge	✓	✓	×	✓	✓
Attribute-based access control	✓	✓	✓	×	×
Our protocol	✓	✓	✓	✓	✓

rithm produces a private signing key sk and a public verification key pk.
2. $Sign$: This algorithm produces digital signatures. On input a private key sk and a message m, this algorithm generates a signature σ on m.
3. $Verify$: This algorithm verifies the validity of digital signatures. On input a message-signature pair (m, σ) and a public key pk, this algorithm outputs "1" if σ is a valid signature, or "0" otherwise.

The correctness of a digital signature scheme requires that $Verify(m, pk, Sign(m, sk)) = 1$ for any pair (sk, pk) output by $KeyGen$.

The standard security requirement of digital signatures is existential unforgeability against adaptive chosen message attacks (Vaudenay 2005). This is defined by a game between the challenger and a probabilistic polynomial-time (PPT) adversary. A PPT adversary is allowed to adaptively make signing queries to a signing oracle which outputs a valid signature on the message chosen by the adversary. After all queries are made, let $M = \{m_i\}$ be the set of messages chosen by the adversary. The adversary outputs a pair (m^*, σ^*) and breaks the existential unforeability of a signature scheme if $Verify(m^*, \sigma^*, pk) = 1$ and $m^* \notin M$. A digital signature scheme is existentially unforgeable if no PPT adversary can win the game with a non-negligible probability. Examples of existentially unforgeable digital signature schemes include Schnorr signature (Schnorr 1989), BLS signature (Boneh et al. 2004) and BB signatures (Boneh and Boyen 2008)

3.2 Cryptographic hash function

A hash function maps a message of arbitrary size to a digest of fixed size. Hash function is the cornerstone of modern cryptography. A cryptographic hash function, denoted by H, must possess the following properties:

1. Pre-image resistance: Given a hash value h, no PPT algorithm can find a message m satisfying $h = H(m)$ with a non-negligible probability.
2. Second pre-image resistance: Given a message m_1, no PPT algorithm can find a different message m_2 satisfying $H(m_1) = H(m_2)$ with a non-negligible probability.
3. Collision resistance: No PPT algorithm can find a pair of different messages (m_1, m_2) satisfying $H(m_1) = H(m_2)$ with a non-negligible probability.

3.3 Complexity assumptions

Let G be a multiplicative group with prime order p, and g be the generator. Given an element $X \in G$, let $x = DL_g X$ denote the discrete logarithm of X on the base g in the group G, i.e., $X = g^x$.

The computational Diffie-Hellman problem is that given (G, g, p) and two random group elements $X, Y \in G$, output $Z \in G$ such that

$$DL_g Z = DL_g X \cdot DL_g Y.$$

The computational Diffie-Hellman assumption states that no PPT algorithm can solve the computational Diffie-Hellman problem with a non-negligible probability.

4 A secure and cost-effective access control protocol

Our protocol deploys the concept of attribute-based system to avoid any communication between the cloud and the SME during user authentication stage. The user database is kept inside the SME to achieve the highest data protection of users while it can also withstand pre-computed dictionary attacks or rainbow table attacks on the server side and phishing attacks on the user side. It can further provide traceability to the SME.

4.1 Description of protocol

There are three stages in our protocol: Setup, User registration, and User Authentication.

Setup stage: The SME executes the Diffie-Hellman key exchange protocol with each cloud server. Namely, with a common prime order (p) group G generated by a generator g, the SME (with a secret value $k_s \in Z_p$) publishes g^{k_s}. At the same time, the cloud (with a secret value $k_c \in Z_p$) also publishes g^{k_c}. We assume both g^{k_s} and g^{k_c} are certified by

an authority. Then the SME computes $K = (g^{k_c})^{k_s}$ while the cloud computes $K' = (g^{k_s})^{k_c}$. Since $K = K'$, both sides establish a common key. The SME chooses a signing key sk and a verification key pk and also publishes pk. The cloud can then verify any signature generated by the SME using pk.

User registration stage: A user with a user name USERNAME and a set of attributes ATTB chooses a secret key usk and the corresponding public key upk, password PASSWORD and sends {PASSWORD, upk} to the SME. The SME uses its signing key sk to generate two signatures as follows:

$$\sigma_1 = \mathrm{SIGN}_{sk}\Big(\mathrm{USERNAME}||upk|| H\big(\mathrm{PASSWORD}||PRF(K||\mathrm{USERNAME})\big)\Big)$$
$$\sigma_2 = \mathrm{SIGN}_{sk}(\mathrm{USERNAME}||\mathrm{ATTB}).$$

Here, "||" denotes concatenation of strings and H is a collision-resistance hash functions and PRF is a pseudo-random function (Goldreich et al. 1986). (Note that we can fix the length of each item. If an item is shorter than the predefined length, we can add some zero-padding to increase the length.)

User authentication stage: A user sends σ_1, σ_2, upk, his/her attribute set ATTB and types his/her username USERNAME and password PASSWORD to the cloud. If ATTB cannot fulfill the required policy for this service, the cloud rejects the connection immediately. Otherwise, the cloud first verifies if $\mathrm{VERIFY}_{pk}(\sigma_2, \mathrm{USERNAME}||\ \mathrm{ATTB})$ returns true. If yes, it further computes $M = \mathrm{USERNAME}||upk||H\big(\mathrm{PASSWORD}||PRF(K'||\mathrm{USERNAME})\big)$ and verifies if $\mathrm{VERIFY}_{pk}(\sigma_1, M)$. If all verifications pass, the cloud sends a random number t to the user. The user computes a signature $\sigma_s = \mathrm{SIGN}_{usk}(t)$ using his/her secret key usk and sends σ_s back to the cloud. The cloud verifies if $\mathrm{VERIFY}_{upk}(\sigma_s, t)$. If it returns true, the cloud accepts the user. In practice, all communications between the user and the cloud should be protected under SSL connection. The flowchat of this stage is shown in Fig. 2. The frequently used notations of our protocol are also summarized in Table 2.

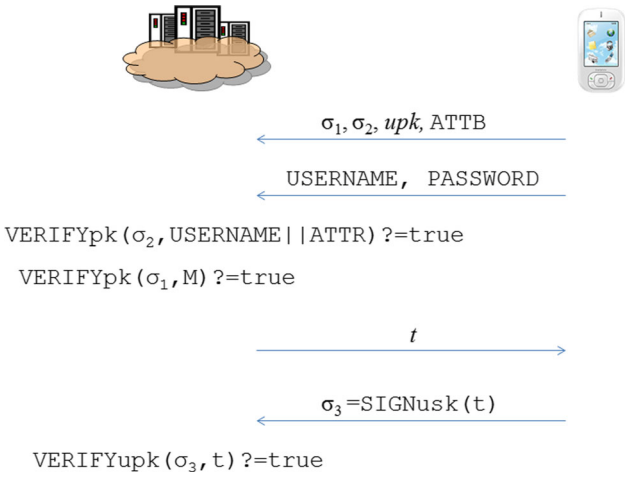

Fig. 2 User authentication stage

4.2 Extension

There are some extensions that our protocol can further include.

One-time password: In order to further increase the security protection, there is an option to add a one-time password (OTP) to the protocol. An OTP can be sent to the mobile device through SMS and the user is required to type the OTP during the authentication. In this case, the mobile number of the user should be added to the message signed by σ_1 and sent together with other information to the cloud by the user.

User revocation: User revocation can be implemented by having a user revocation list given by the SME to the cloud. The cloud first checks against the revocation list for the communicating user (with his/her username). If the username is on the list, the cloud rejects the connection immediately.

4.3 Security analysis

We briefly describe how our proposed protocol can defend against the following attacks and achieve some security features:

Dictionary attack: Systems are exposed to dictionary attacks if they store the hash (or salted hash) of users' password. In our protocol, the hash (or salted hash) of password is never stored on any server (including the SME server or the cloud server). Even though the adversary can somehow extract σ_1 (that contains the salted hash of the password), since the adversary does not know K or K' which are known to the cloud and the SME only (due to the computational Diffie-Hellman assumption), it is difficult to launch the attack. Moreoever, the "salt" value is an output of a pseudorandom function which takes the username as the input. In other words, the "salt" value is different for every user. It further increases the difficulty for dictionary attacks.

Phishing attack: Even the adversary uses some phishing technique to re-direct the user to another site to capture the username and password, the adversary still cannot pretend the user to authenticate with the cloud since it does not know the secret key usk inside the device. As long as the signature scheme is existentially unforgeable, the adversary can neither extract the secret key nor forge a signature even it has observed a polynomial number of signatures generated.

Authenticity: Users with attributes not satisfying the policy cannot gain access to the cloud. It can be seen from the fact that the attributes are included in σ_2 which is a signa-

Table 2 Frequently used notations

sk	Secret key of the SME (used to generate signature)
pk	Public key of the SME (used to verify signature from SME)
USERNAME	Username of a user
PASSWORD	Password of a user
ATTB	The attribute-set of a user
usk	User secret key
upk	User public key
SIGN_{key}	Signature generation algorithm by the signing key key
	It outputs a signature
VERIFY_{pkey}	Signature verification by the verification key $pkey$
	It outputs true or false indicating a valid or invalid signature respectively
H	A hash function
PRF	A pseudorandom function

ture signed by the SME. As long as the signature scheme is existentially unforgeable, the user has no way to modify the attribute list.

Traceability: Username is sent to the cloud by the user during the authentication stage. If a user uses a different username rather than the one included in σ_1, the verification will not pass. Thus the cloud can maintain a log of usernames who have been accessed for the service.

4.4 Performance evaluation

Our protocol is very efficient. The mobile device only needs to compute one exponentiation during the registration stage, and one signature generation (if Schnorr signature scheme (Schnorr 1989) is used, it only takes one exponentiation) during the authentication stage. Using the simulation data by jPBC (Caro 2015), a mobile device (HTC Desire HD A9191, Android 2.2) requires 71.5 ms to execute an exponentiation if elliptic curve (supersingular curve $y^2 = x^3 + x$) is used with 160 bits group order (equivalent to 1024 bits RSA security). On the SME server, it only needs to generate two signatures during the registration stage and the cloud only needs to verify three signatures during the authentication stage. The running time of these algorithms by the SME and the cloud should be regarded as negligible.

5 Conclusion

In this article, we have suggested a fuzzy access control protocol for SME providing B2C services on mobile cloud computing platform. Our protocol allows the SME to be offline during user authentication stage, which can greatly reduce the running cost of SME as it does not need to have a strong firewall or large bandwidth connecting with the cloud. Our protocol can further withstand common attacks such as dictionary attack on server or phishing attack on client. Traceability is provided to the SME and it is very efficient to mobile devices. We believe it is practical to be implemented in the commercial world.

Acknowledgements This work is supported by National Natural Science Foundation of China (61472083, 61402110, U1405255), Fok Ying Tung Education Foundation (141065), ISN Research Fund (ISN15-03), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, Ministry of Education of China, and Fujian Normal University Innovative Research Team (IRTL1207).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Analytics SP (2013) Mobile cloud platforms: the back end of mobile apps. http://www.reportlinker.com/p01650001-summary/Mobile-Cloud-Platforms-The-Back-end-of-Mobile-Apps.html

Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. In: 2007 IEEE symposium on security and privacy (S&P 2007), 20–23 May 2007, Oakland, IEEE Computer Society, pp 321–334 (2007)

Boneh D, Boyen X (2008) Short signatures without random oracles and the SDH assumption in bilinear groups. J Cryptol 21(2):149–177

Boneh D, Lynn B, Shacham H (2004) Short signatures from the weil pairing. J Cryptol 17(4):297–319

Caro AD (2015) The java pairing based cryptography library (jpbc). http://libeccio.dia.unisa.it/projects/jpbc/

Chen X, Li J, Huang X, Li J, Xiang Y, Wong DS (2014) Secure outsourced attribute-based signatures. IEEE Trans Parallel Distrib Syst 25(12):3285–3294

Garg S, Gentry C, Halevi S, Sahai A, Waters B (2013) Attribute-based encryption for circuits from multilinear maps. In: Canetti R, Garay JA (eds) Advances in cryptology—CRYPTO 2013–33rd annual cryptology conference, Santa Barbara, August 18–22, 2013. Proceedings part II. Lecture notes in computer science, vol 8043. Springer, pp 479–499 (2013)

Goldreich O, Goldwasser S, Micali S (1986) How to construct random functions. J ACM 33(4):792–807

Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: Juels A, Wright RN, di Vimercati SDC (eds) Proceedings of the 13th ACM conference on computer and communications security, CCS 2006, Alexandria, October 30–November 3, 2006. ACM, pp 89–98 (2006)

Hohenberger S, Waters B (2013) Attribute-based encryption with fast decryption. In: Kurosawa K, Hanaoka G (eds) Public-key cryptography—PKC 2013—16th international conference on practice and theory in public-key cryptography, Nara, February 26–March 1, 2013. Proceedings Lecture Notes in Computer Science, vol 7778. Springer, pp 162–179 (2013)

Hohenberger S, Waters B (2014) Online/offline attribute-based encryption. In: Krawczyk H (ed) Public-key cryptography—PKC 2014–17th international conference on practice and theory in public-key cryptography, Buenos Aires, March 26–28, 2014. Proceedings lecture notes in computer science, vol 8383. Springer, pp 293–310 (2014)

Lewko AB, Waters B (2011) Decentralizing attribute-based encryption. In: Paterson KG (2011), pp 568–588

Lewko AB, Waters B (2011) Unbounded HIBE and attribute-based encryption. In: Paterson KG (2011), pp 547–567

Lewko AB, Waters B (2012) New proof methods for attribute-based encryption: achieving full security through selective techniques. In: Safavi-Naini R, Canetti R (2012), pp 180–198

Li J, Chen X, Huang X (2015) New attribute-based authentication and its application in anonymous cloud access service. IJWGS 11(1):125–141

Li J, Huang X, Li J, Chen X, Xiang Y (2014) Securely outsourcing attribute-based encryption with checkability. IEEE Trans Parallel Distrib Syst 25(8):2201–2210

Liu Z, Weng J, Li J, Yang J, Fu C, Jia C (2015) Cloud-based electronic health record system supporting fuzzy keyword search. Soft Computi, pp 1–13

Maji HK, Prabhakaran M, Rosulek M (2011) Attribute-based signatures. In: Kiayias A (ed) Topics in cryptology—CT-RSA 2011–the cryptographers' track at the RSA conference 2011, San Francisco, February 14–18, 2011. Proceedings Lecture Notes in Computer Science, vol 6558. Springer, pp 376–392

Ostrovsky R, Sahai A, Waters B (2007) Attribute-based encryption with non-monotonic access structures. In: Ning P, di Vimercati SDC, Syverson PF (eds) Proceedings of the 2007 ACM conference on computer and communications security, CCS 2007, Alexandria, October 28–31, 2007. ACM, pp. 195–203 (2007)

Paterson KG (ed) Advances in cryptology—EUROCRYPT 2011–30th annual international conference on the theory and applications of cryptographic techniques, Tallinn, May 15–19, 2011. Proceedings lecture notes in computer science, vol 6632. Springer (2011)

Pirretti M, Traynor P, McDaniel P, Waters B (2010) Secure attribute-based systems. J Comput Secur 18(5):799–837

Rouselakis Y, Waters B (2013) Practical constructions and new proof methods for large universe attribute-based encryption. In: Sadeghi A, Gligor VD, Yung M (eds) 2013 ACM SIGSAC conference on computer and communications security, CCS'13, Berlin, November 4–8, 2013. ACM, pp 463–474 (2013)

Rouselakis Y, Waters B (2015) Efficient statically-secure large-universe multi-authority attribute-based encryption. In: Böhme R, Okamoto T (eds) Financial Cryptography and Data Security–19th International Conference, FC 2015, San Juan, Puerto Rico, January 26–30, 2015, Revised selected papers. Lecture notes in computer science, vol 8975. Springer, pp 315–332 (2015)

Safavi-Naini R, Canetti R (eds) (2012) Advances in Cryptology—CRYPTO 2012–32nd annual cryptology conference, Santa Barbara, August 19–23, 2012. Proceedings lecture notes in computer science, vol 7417. Springer

Sahai A, Seyalioglu H, Waters B (2012) Dynamic credentials and ciphertext delegation for attribute-based encryption. In: Safavi-Naini R, Canetti R (2012), pp 199–217

Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Cramer R (ed) Advances in cryptology—EUROCRYPT 2005, 24th annual international conference on the theory and applications of cryptographic techniques, Aarhus, May 22–26, 2005, Proceedings lecture notes in computer science, vol 3494. Springer, pp 457–473

Schnorr C (1989) Efficient identification and signatures for smart cards. In: Brassard G (ed) Advances in cryptology—CRYPTO '89, 9th annual international cryptology conference, Santa Barbara, August 20–24, 1989, Proceedings lecture notes in computer science, vol 435. Springer, pp 239–252

Vaudenay S (2005) A classical introduction to cryptography: applications for communications security. Springer

Waters B (2011) Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano D, Fazio N, Gennaro R, Nicolosi A (eds) Public key cryptography—PKC 2011–14th international conference on practice and theory in public key cryptography, Taormina, March 6–9, 2011. Proceedings lecture notes in computer science, vol 6571. Springer, pp 53–70

Wei J, Huang X, Hu X, Liu W (2015) Revocable threshold attribute-based signature against signing key exposure. In: Lopez J, Wu Y (eds) Information security practice and experience–11th international conference, ISPEC 2015, Beijing, May 5–8, 2015. Proceedings lecture notes in computer science, vol 9065. Springer, pp 316–330

Wu Z, Liang B, You L, Jian Z, Li J (2015) High-dimension space projection-based biometric encryption for fingerprint with fuzzy minutia. Soft Comput, pp 1–12

Xhafa F, Wang J, Chen X, Liu JK, Li J, Krause P (2014) An efficient phr service system supporting fuzzy keyword search and fine-grained access control. Soft Comput 18(9):1795–1802