Integration of Hierarchical Access Control and Keyword Search Encryption in Cloud Computing Environment

Chih Hung Wang and Chia-Chun Hsu

***Abstract*—An increasing number of applications have been proposed and discussed on the cloud computing environment because it can bring many benefits like reducing the cost of maintaining data centers in an enterprise, low data management cost and retrieval of data whenever you want, etc. As more and more sensitive information and personal data are centralized into the cloud servers, how to protect data privacy and combat the unauthorized accesses is an important issue in the cloud computing environment since the outsourcing-service server may not be fully trusted. One method to alleviate the security worries is storing data in the encrypted form. The drawback of the encrypted data is the limitation of usability. The keyword search encryption technique has been proposed to solve the query limitation for the encrypted data; however, most of the researches focus on processing an individual user request. The proposed approach combines the fine-grained access control and keyword search encryption to provide multiple users' access controls in the cloud environment with encrypted data protection. The user in this situation can retrieve the data file from the cloud database only if she/he gives appropriate keywords and presents the identity or position that satisfies the rules of the access rights.**

***Index Terms*—Cloud computing, keyword search encryption, hidden vector encryption, fine-grained access control.**

I. Introduction

With the rapid growth of networking technology and popularity of Internet, many Information Technique (IT) enterprises and organizations are tending to outsource their computer environment to reduce the cost of maintaining their own data centers and take advantage of low data management cost. Cloud computing is a computational model over a shared-virtualized pool of computing resource and provides resource of the computing infrastructure and outsourced storage space as service over the Internet. It is more reliable to centrally store the sensitive information and personal data in the cloud servers. However, since the outsourcing-service server may not be fully trusted, protecting the data privacy and resisting the unauthorized accesses become important issues in the cloud computing environment.

Access control is the component of security systems responsible to evaluate if a subject can be allowed to operate in a given way on a specific resource. Most often, access is granted according to a number of different constraints depending on the privileges of the users to meet different user requirements. In hierarchical access control systems, keys are organized in a partially ordered hierarchy, such that higher level keys can be derived from the lower level keys. To protect the data against unauthorized disclosure, the data streams which may be accessed by different members should be encrypted with different secret keys. Another important issue of cloud computing improvement is the protection of data privacy. One of the popular methods to alleviate the security worries is to encrypt the data before outsourcing and retrieve the encrypted data by the keyword-based search, in which the legal user can retrieve the encrypted file by the query tokens. Once the user wants to retrieve the file, he must know correct attributes and use a decrypted key to generate the token for querying out the file. In 2004, Boneh *et al.* [1] proposed the Public Key Encryption with Keyword Search (PEKS) as the primitive scheme of retrieving encrypted data by keywords. Later, Shao *et al.* [2] proposed a new scheme called PRES (Proxy Re-encryption with Keyword Search) to combine the two novel concepts. Zhang *et al.* [3] presented a more efficient construction of PEKS to solve the problem of conjunctive keyword search with subset keywords. Lots of researches have proposed improved schemes in the computational efficiency for the keyword search encryption. Park in [4] reduced the query token size into a constant and made the computation time of query operations in constant complexity. Further, some of previous papers proposed variety of query types which support not only equality query but also conjunction query to make the method more practical [5]. Byun and Lee [6] defined a security model for conjunctive keyword search on a practical relational database.

The property of fine-grained access control would be the necessary mechanism in cloud computing environment because the pay-as-you-go property, the main benefit and concept in cloud computing environment, makes the cloud service customer unwill to waste any bandwidth to access the file he does not need. We develop a conjunctive keyword search encryption with the property of the fine-grained data access control. In our scheme, data sender would not only generate the ciphertext but also create a fine-grained access structure which defines the file access right and makes the query process more precise. The user who wants to retrieve the certain file must not only know the correct keywords but also have the authorization right to the file. For example, in the healthcare application scenarios, a patient's record is stored in the outsourced database and can be exchanged with other hospitals through Internet for medical purposes. Not everyone has the permission to access a certain file; therefore, distributing the access right of each file can achieve the

Manuscript received November 25, 2012; revised January 28, 2013. This work was supported in part by the National Science Council under the Grant NSC 101-2219-E-415-001.

The authors are with the Department of Computer Science and Information Engineering, National Chiayi University, Chiayi City 60004, Taiwan (e-mail: wangch@mail.ncyu.edu.tw, s1000422@mail.ncyu.edu.tw).

DOI: 10.7763/IJCCE.2013.V2.199

accountability of file access and enhance the data privacy as well as practicality for the real case applications.

II. Related Work

A. Keyword Search Encryption

In 2000, Song, Wagner and Perrig [7] introduced a practical concept of searchable encryption on encrypted data. To demand fine-grained and more expressive decryption capabilities, Boneh *et al.* [1] proposed the concept of PEKS which implies Identity Based Encryption (IBE). Using IBE system, the user can encrypt data by the particular public key with a given string. To support more expressive schemes such as conjunctive queries and multi-dimensional range queries, some public-key predicate encryption schemes inspired by the anonymous identity-based encryption have been proposed [8]–[12]. In the predicate encryption scheme, the secret keys correspond to predicates and the ciphertexts are associated with attributes; the secret key corresponding to a predicate can be used to decrypt a ciphertext associated with the attribute if and only if the predicate can match this attribute. Shi and Waters [11] proposed a delegation mechanism for a class of predicate encryption. Shen *et al.* [10] discussed the plaintext privacy and predicate privacy in the predicate encryption and used a symmetric-key approach to protect the privacy. Katz *et al.* [8] used the inner product method to construct the predicate for increasing the expressiveness. There are some extension schemes of predicate encryption by using the inner product method. Okamoto and Takashima [9] proposed the hierarchical mechanism for the predicate encryption but only selective security is proven. Moreover, Yoshino *et al.* [12] proposed the predicate encryption scheme based on the three groups in the symmetric cryptography system which can satisfy the selective security model under a non-interactive assumption.

In this paper, we illustrate how to integrate the fine-grained access control and keyword search encryption in one system by using the symmetric key predicate encryption scheme from Shen *et al.* [10] which supports the complex and expressive query by using the inner product computation.

B. Fine-Grained Access Control

To maintain the security of any particular host is becoming increasingly difficult because of the increasing number of worm attacks, various types of intrusions and dangers of insider attacks. Encrypting data can reduce the latent vulnerabilities but limits the users to selectively share their encrypted data at a fine-grained level. Fine-grained access control system can flexibly define the different access rights of the individual users.

Sahai and Waters [13] proposed a novel fuzzy identity-based encryption which allows data provider to arbitrarily label the sharing data in encrypted form and also called "attribute-based encryption". Thus, there are two major policies of fine-grained access control system, ciphertext-policy attribute-based encryption and key-policy attribute-based encryption. Ciphertext-policy attribute-based encryption was first proposed by Bethencourt, Sahai and Waters [14] in which an access structure would be associated with the ciphertexts and user's private keys would be associated with a set of attributes. Key-policy attribute-based encryption is first proposed by Goyal *et al.* [15] in which the access structure is specified in the private key and the ciphertexts are simply labeled with a set of descriptive attributes. In next session, we extend the method of [15] to construct our proposed model.

III. Proposed Integration Model

We present a model of fine grained access mechanism with keyword search property to be suitable for the following scenario. In health care system, the patient record would be labeled with some outpatient categories such as area, hospital, illness, etc. and patient identities such as race, age, sex, etc. We take outpatient categories as the access right of each user and patient identities as the keywords of the file. We would use the outpatient categories to build the access structure of each sharing patient record and thus to construct a fine-grained access control to meet different users' requirements. According to the access structure, the user would be limited to search the patient records which could be consistent with his access right.

Our proposed scheme combines the concept of fine-grained access control of [15] and predicate encryption of [4], [10]. The readers can refer to these papers for detail mathematics. There are four parties in our proposed model: key distribution center which defines the public parameter and delegates the user access right, data sender who uploads the sharing files to the cloud server, data consumer who would retrieve the files sharing on cloud server according his access right. Each user in our model would first register at key distribution center to get the user right as his key. The data sender uses his user right key to build the file access structure and create the temporary keys which would be used to encrypt the uploading file with keywords by using the public parameters. The data sender sends file access structure (together with public parameters) and ciphertext to the access control center and data query center respectively. Note that both of them are the part of the cloud server. Once the data consumer wants to retrieve the encrypted file, he must have adequate access right for the file's access structure and create the correct query token to retrieve the file. The detail illustrations of the construction of our model are stated below (also see Fig. 1).

We construct our scheme by using prime-order group of bilinear map and the concept of polynomial interpolation. Let G and G_T be the two groups of prime order p, and let $e: G \times G \to G_T$ be the bilinear map. Define the Lagrange coefficient $\Delta_{i,S}$ for $i \in Z_p$ and a set, S, of elements in Z_p : $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} {}^{x-j}\!/\!_{i-j}$.

A. $Setup(\rho, \ell, n)$

The input parameters include the security parameter ρ, a dimension ℓ of vector which indicates the user right attribute and a dimension n of vector which indicates the keyword attribute. A key distribution center (KDC) defines the

universe of user attributes $\Omega = \{1, 2, ..., \ell\}$ and randomly chooses $u_i \in Z_p$ for $i \in \Omega$. Finally, a key distribution center chooses $2n+3$ hash functions: $\left(H_{y,1} : G_T \to Z_p,\ H_{y,2} : G_T \to Z_p,\ H_Q : G_T \to G,\ H_{r,\lambda} : G_T \to G,\ H_{t,\lambda} : G_T \to G\right)$, for $\lambda = 1, 2, ..., n$. Assuming that $g \in G$ is a random generator, the public parameters are given:

$$PP = \left(p, G, G_T, e, g, H_{y,1}, H_{y,2}, H_Q, \left(H_{r,\lambda}, H_{t,\lambda}\right)_{\lambda \in [1,n]}, U_1 = g^{u_1}, U_2 = g^{u_2}, ..., U_\ell = g^{u_\ell} \right)$$

and the master key is denoted by $(u_1, u_2, ..., u_\ell)$.

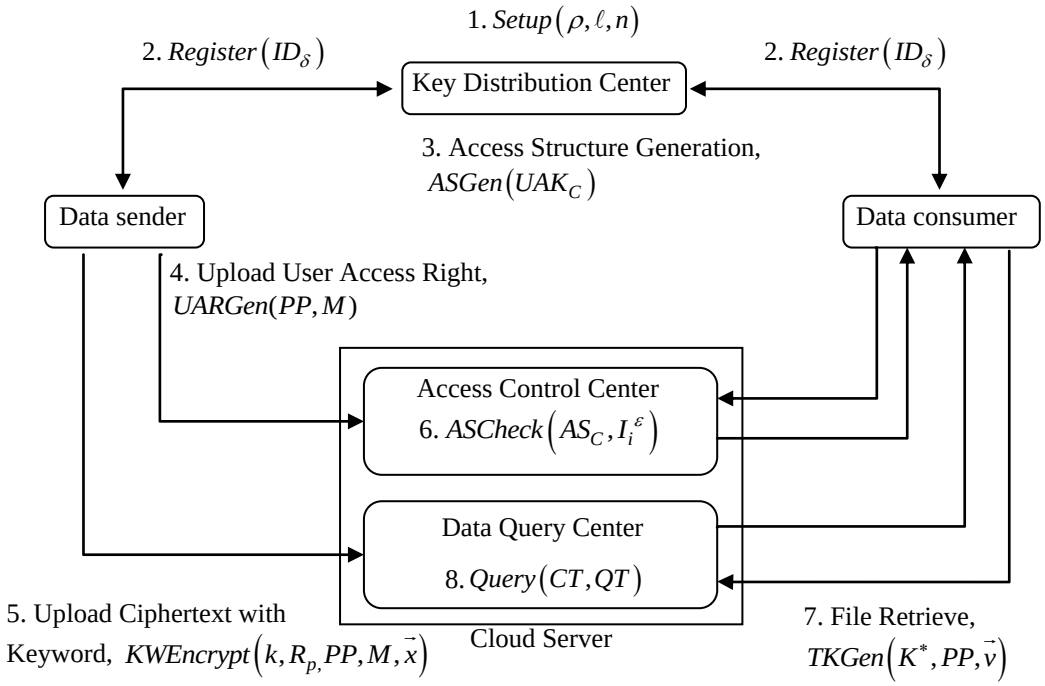

Fig. 1. The concept of the proposed model.

B. $Register(ID_\delta)$

When $user_\delta$ has registered at the key distribution center, he gets the partial master key u_i from KDC as the user access key UAK_δ for $i \in \pi_\delta$ and $\pi_\delta \subseteq \Omega$. That means each user in our system has different attribute set π_δ according his identity ID_δ.

C. $ASGen(UAK_C)$

This algorithm is performed by the key distribution center. It builds the file access structure by using the data consumer's user access key UAK_C. The output is a tree type access structure AS_C in which the non-leaf tree node represents a threshold gate. First thing to construct AS_C is deciding the polynomials of each node in AS_C by following top-down manner. The polynomial $f_X(\,)$ of each node X in AS_C would be set the degree $D_X = W_X - 1$ where W_X is the threshold value of the node X. For the root node r, KDC sets $f_r(0) = \mu$ where μ is random value in Z_p and stores μ as the root secret of the access structure AS_C. The polynomial $f_r(\,)$ can be completely constructed by randomly choosing D_r other points. For any other node X, $f_X(0) = f_{parent(X)}(index(X))$ where $parent(X)$ denotes the parent node of X and $index(X)$ denotes a number associated with the node X. Similarly, the polynomial $f_X(\,)$ can be completely constructed by randomly choosing D_X other points. After defining all of polynomials for each node in AS, KDC computes $I_L = g^{f_L(0)/u_i}$ for each leaf node L where $i \in \pi_C$. Finally, KDC delivers AS_C and a public information $R_P = e(g,g)^\mu$ to the access control center for $ASCheck(AS_C)$ process.

D. $UARGen(PP, M)$

This algorithm is performed by the data sender. The data sender defines the useful user access set $\omega \subseteq \Omega$ for the specific message M. After that, he randomly chooses $k \in Z_p$ and then computes $O_i = U_i^k = g^{u_i \cdot k}$ for all $i \in \omega$. Finally, the data sender uploads $\{O_i\}_{i \in \omega}$ as user access right UAR with the set ω to the access control center.

E. $KWEncrypt(k, R_p, PP, M, \vec{x})$

This algorithm is performed by the data sender and outputs the ciphertext CT. The inputs are the value k which is decided at $UARGen(PP, M)$ process, the public parameter PP, the plaintext $M \in G_T$ and the keyword vector $\vec{x} = (x_1, x_2, ..., x_n)$. First, the data sender generates

$K=(R_p)^k=e(g,g)^{\mu k}\in G_T$ and then computes $y_1=H_{y,1}(K)$, $y_2=H_{y,2}(K)$, $Q=H_Q(K)$, $r_\lambda=H_{r,\lambda}(K)$ and $t_\lambda=H_{t,\lambda}(K)$ for $\lambda=1,2,...,n$. Finally, the data sender chooses a random value $s\in Z_p$ and outputs ciphertext

$$CT=\begin{pmatrix} C=g^{y_1 s}, C_0=g^{y_2 s}, \{C_{1,i}=r_i^{x_i s}, C_{2,i}=t_i^{s}\}_{i=1}^{n}, \\ C_3=M\cdot e(g,Q)^{s(y_1+y_2)} \end{pmatrix}.$$

F. $ASCheck\left(AS_C, I_i^{\varepsilon}\right)$

This algorithm checks the access right of the user. The access control center performs the algorithm to compute $F_i=e(O_i,I_i^{\varepsilon})=e(O_i,I_i)^{\varepsilon}$ for $i\in\omega$ where ε is a random secret selected by the data consumer (that means the data consumer uploads the value of I_i^{ε} to the access control center). If $i\notin\pi$, the computation result of F_i would be equal to ϕ, which means the output result is invalid. In AS_C, for all nodes α that are children of the node β. Let S_β be an arbitrary W_β-sized set of child nodes α such that $F_\alpha\neq\phi$. If the set S_β doesn't exist that means the threshold value of node β is not satisfied and then the algorithm outputs $F_\beta=\phi$; otherwise the computation result of F_β would be as follows.

$$F_\beta=\prod_{\alpha\in S_\beta}F_\alpha^{\Delta_{i,S_\beta'}(0)}=\prod_{\alpha\in S_\beta}\left(e(g,g)^{\varepsilon\cdot k\cdot f_\alpha(0)}\right)^{\Delta_{i,S_\beta'}(0)}$$

$$=\prod_{\alpha\in S_\beta}\left(e(g,g)^{\varepsilon\cdot k\cdot f_{parent(\alpha)}(index(\alpha))}\right)^{\Delta_{i,S_\beta'}(0)}=\prod_{\alpha\in S_\beta}\left(e(g,g)^{\varepsilon\cdot k\cdot f_\beta(i)}\right)^{\Delta_{i,S_\beta'}(0)}$$

$=e(g,g)^{\varepsilon\cdot k\cdot f_\beta(0)}$, where $i=index(\alpha)$ and $S_\beta'=(index(\alpha):\alpha\in S_\beta)$. The access control center would recursively compute $F_\beta=\prod_{\alpha\in S_\beta}F_\alpha^{\Delta_{i,S_\beta'}(0)}$ in the button-up manner of access structure tree AS_C. If the user access is authorized, the access control center finally can give the data consumer a value of root node $F_r=e(g,g)^{\varepsilon\cdot k\cdot\mu}$; otherwise it gives a value of ϕ.

G. $TKGen\left(K^*, PP, \vec{v}, \varepsilon\right)$

This algorithm is to generate the query token *QT* which would be transmitted to the data query center to query out the file. The inputs consist of user right check result K^*, public parameter PP and query vector $\vec{v}=(v_1,v_2,...,v_n)$. If $K^*\neq\phi$, the data consumer computes $K'=\left(K^*\right)^{1/\varepsilon}=e(g,g)^{\mu k}$; otherwise stops the file retrieval process. After getting K', the data consumer generates $y_1'=H_{y,1}(K')$, $y_2'=H_{y,2}(K')$, $Q'=H_Q(K')$, $r_\lambda'=H_{r,\lambda}(K')$ and $t_\lambda'=H_{t,\lambda}(K')$ and randomly chooses $A,B,a_\lambda,b_\lambda,c_\lambda,d_\lambda\in Z_p$ such that $a_\lambda y_1'+b_\lambda y_2'=A$ and $c_\lambda y_1'+d_\lambda y_2'=B$ for $\lambda=1,2,...,n$. Finally, the data consumer computes the query token

$$QT=\begin{pmatrix} K=Q'\cdot\prod_{i=1}^{n}r_i'^{a_i v_i}t_i'^{c_i}, K_0=Q'\cdot\prod_{i=1}^{n}r_i'^{b_i v_i}t_i'^{d_i}, \\ K_1=g^A, K_2=g^B \end{pmatrix}.$$

H. $Query(CT, QT)$

This algorithm would output the querying file to the data consumer. Let $C_1'=\prod_{i=1}^{n}C_{1,i}$ and $C_2'=\prod_{i=1}^{n}C_{2,i}$. The querying file can be retrieved by the following equation:

$$M'\leftarrow\frac{e(K_1,C_1')\cdot e(K_2,C_2')}{e(K,C)\cdot e(K_0,C_0)}\cdot C_3$$

Correctness:

If $K'=K$, then the value of $y_1'=y_1$, $y_2'=y_2$, $Q'=Q$, $r_\lambda'=r_\lambda$ and $t_\lambda'=t_\lambda$ for $\lambda=1,2,...,n$. Thus, let

$$\delta=e(K_1,C_1')\cdot e(K_2,C_2')$$

$$=e\left(g^A,\prod_{i=1}^{n}C_{1,i}\right)\cdot e\left(g^B,\prod_{i=1}^{n}C_{2,i}\right)$$

$$=e\left(g^A,\prod_{i=1}^{n}r_i^{x_i s}\right)\cdot e\left(g^B,\prod_{i=1}^{n}t_i^{s}\right),\text{ and}$$

$$\varphi=e(K,C)\cdot e(K_0,C_0)$$

$$=e\left(Q'\cdot\prod_{i=1}^{n}r_i'^{a_i v_i}t_i'^{c_i},g^{y_1 s}\right)\cdot e\left(Q'\cdot\prod_{i=1}^{n}r_i'^{b_i v_i}t_i'^{d_i},g^{y_2 s}\right)$$

$$=e\left(Q',g^{y_1 s}\right)\cdot e\left(Q',g^{y_2 s}\right)e\left(\prod_{i=1}^{n}r_i'^{a_i v_i}t_i'^{c_i},g^{y_1 s}\right)$$

$$\cdot e\left(\prod_{i=1}^{n}r_i'^{b_i v_i}t_i'^{d_i},g^{y_2 s}\right)$$

$$=e(Q',g)^{s(y_1+y_2)}\cdot e\left(\prod_{i=1}^{n}r_i'^{a_i v_i},g^{y_1 s}\right)\cdot e\left(\prod_{i=1}^{n}r_i'^{b_i v_i},g^{y_2 s}\right)$$

$$\cdot e\left(\prod_{i=1}^{n}t_i'^{c_i},g^{y_1 s}\right)\cdot e\left(\prod_{i=1}^{n}t_i'^{d_i},g^{y_2 s}\right)$$

$$=e(Q',g)^{s(y_1+y_2)}\cdot e\left(\prod_{i=1}^{n}r_i'^{a_i v_i y_1+b_i v_i y_2},g^{s}\right)$$

$$\cdot e\left(\prod_{i=1}^{n}t_i'^{c_i y_1+d_i y_2},g^{s}\right)$$

$$=e(Q',g)^{s(y_1+y_2)}\cdot e\left(\prod_{i=1}^{n}r_i'^{Av_i},g^{s}\right)\cdot e\left(\prod_{i=1}^{n}t_i'^{B},g^{s}\right)$$

thus,

$$\frac{e(K_1,C_1')\cdot e(K_2,C_2')}{e(K,C)\cdot e(K_0,C_0)}\cdot C_3=\frac{\delta}{\varphi}\cdot C_3=\frac{M\cdot e(g,Q)^{s(y_1+y_2)}}{e(Q',g)^{s(y_1+y_2)}}=M$$

IV. Analysis and Discussion

The proposed model is to integrate the method of [15] and [10] to achieve a fine-grained data access with keyword search encryption. The two security complexity assumptions applied in our scheme, the bilinear map complexity assumption and the extension format of bilinear map complexity assumption, are deriving from [10], [15]. Inevitably, our scheme can achieve the level of the security that [10], [15]. proposed. The reader can refer to their papers for the details of security proofs. Although our proposed model can achieve highly secure strength in data protection,

the privacy of the common share key K and the security of the hash functions are critical to the enciphering mechanism for both messages and attributes.

In our construction system, the user needs to store the user access key with the size of $|\pi_\delta| \times |G| \leq \ell \times |G|$. The proposed system uses the public parameter PP and the common share key K to build the temporary key for encryption and query-token generation, and thus in our system the data consumer only needs to upload the elements of the user check tokens $UT = I_i^{\varepsilon}$ with the size of $|\pi_\delta| \times |G|$ and the query tokens QT with the size of $4|G|$ for user fine-grained access check and query. Further, the cost of stored space includes the ciphertext CT with the size of $(2n+2)|G| + |G_T|$ stored in the data query center and O_i with the size of $|\omega| \times |G|$ stored in the access control center.

V. Conclusion

It is the first attempt to design the model that combines the fine-grained data access and the keyword search encryption. The design model is suitable for the cloud computing environment since it provides precise data access mechanism and fast queries for the encrypted files. The proposed model applied the concept of attribute based encryption to build the fine-grained data access mechanism which can be extended to include the keyword search encryption with hierarchical view. However, in our model, the number of query tokens is increasing with the number of access categories of the search files. How to reduce the number of query tokens in the proposed model is our future work. Moreover, we would try to add a revocation function to make a complete solution in securely retrieving private data from the remote cloud servers.

References

[1] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Eurocrypt 2004, LNCS*, vol. 3027, pp. 506-522, 2004.

[2] J. Shao, Z. Cao, X. Liang, and H. Lin, "Proxy re-encryption with keyword search," *Information Sciences*, vol. 180, pp. 2576-2587, 2010.

[3] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," *Journal of Network and Computer Applications*, vol. 34, pp. 262-267, 2011.

[4] J. H. Park, "Efficient hidden vector encryption for conjunctive queries on encrypted data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 10, pp. 1483-1497, 2011.

[5] Y. H. Hwang and P.J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," *Pairing-based Cryptography - PAIRING 2007, LNCS*, vol. 4575, pp. 2-22, 2007.

[6] J. W. Byun and D. H. Lee, "On a security model of conjunctive keyword search over encrypted relational database," *Journal of Systems and Software*, vol. 84, pp. 1364-1372, 2011.

[7] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searching on encrypted data," in *Proc. 2000 IEEE Symposium on Security and Privacy*, IEEE Press, 2000, pp. 44-55.

[8] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proc. the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology*, 2008, pp. 146-162.

[9] T. Okamoto and K. Takashima, "Hierarchical predicate encryption for inner-products," in *Proc. the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, 2009, pp. 214-231.

[10] E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in *Proc. the 6th Theory of Cryptography Conference on Theory of Cryptography*, 2009, pp. 457-473.

[11] E. Shi and B. Waters, "Delegating capabilities in predicate encryption systems," in *Proc. the 35th international colloquium on Automata*, 2008, pp. 560-578.

[12] P. Yoshino, N. Kunihiro, K. Naganuma, and H. Sato, "Symmetric inner-product predicate encryption based on three groups," *ProvSec 2012, LNCS*, vol. 7496, pp. 215-234, 2012.

[13] A. Sahai and B. Waters, "Fuzzy identity based encryption," *Advances in Cryptology-Eurocrypt 2005*, *LNCS*, vol. 3494, pp. 457-473, 2005.

[14] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. the 2007 IEEE Symposium on Security and Privacy*, 2007, pp. 321-334.

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. of the 13th ACM conference on Computer and communications security*, pp. 89-98, 2006.

Chih-Hung Wang was born in Kaohsiung, Taiwan in 1968. He received the BS degree in Information Science from Tunghi University and MS degree in Information Engineering from National Chung Cheng University, Taiwan in 1991 and 1993, respectively. He received the Ph.D. degree in Information Engineering from National Cheng Kung University, Taiwan in 1998. He is presently an associate professor of Department of Computer Science and Information Engineering, National Chiayi University, Taiwan. His research interests include cryptography, information security and data compression.

Chia-Chun Hsu was born in Taoyuan Taiwan, in 1988. He received the BS degree in Department of Computer Science and Information Engineering, National Chiayi University, Taiwan. He is presently a master student of Department of Computer Science and Information Engineering, National Chiayi University, Taiwan. His research interests include cryptographic protocols and information security.