Multi-authority Attribute Based Encryption Scheme with Revocation

XiaoFang Huang, Qi Tao, BaoDong Qin, ZhiQin Liu
Southwest University of Science and Technology, MianYang 621000, Sichuan, China
Email: taoqi15@foxmail.com

***Abstract*—Attribute Based Encryption (ABE) scheme can achieve information sharing of one-to-many users, without considering the number of users and the users identity. But, the traditional single Attribute Authority (AA) ABE scheme can hardly meet requirements of different agencies in distributed application environment and it is easy to form the system performance bottlenecks. Based on ciphertext-policy ABE scheme, this paper proposes a multi-authority revocable ABE scheme, where the classification manages user attributes, effectively relieving the management burden of single organization. In addition, it can achieve fine grained access control of shared information by adopting tree access strategy and secret sharing scheme, and support system attribute revocation. Finally, we show that the scheme is secure against chosen plaintext attack under the Decisional Bilinear Diffie-Hellman (DBDH) assumption.**

Keywords*—attribute based encryption, fine-grained access control, revocation, secure sharing.

I. INTRODUCTION

With the rapid development of Internet technology, more and more information needs to be dealt with and shared online. So, the owners of information increasingly required secure, flexible and efficient control to access to the shared information. The traditional PKI scheme can achieve shared information security, but it can hardly get all the sharers' public keys in a free Internet environment quickly, and it easily lead to much system processing overhead and bandwidth consumption. To solve the above problems, Identity Based Encryption (IBE) was first proposed by *Shamir* [1], which encrypts information with the user' unique identity instead of public key. *Sahai* and *Waters* [2] proposed to encrypt information with the attribute set instead of user's identity to multi-user share information security, without considering the number of users and the user's identity, but the scheme only support threshold policy. With the high efficiency, resisting collusion and flexible access control strategy, ABE scheme has a bright application prospect, such as directional broadcast, cloud storage, and other fields.

The papers [2-6, 8-11] belong to single authority ABE scheme, where both the users' decryption and signature private keys are produced by a single trusted Attribute Authority (AA) according to user's attribute set. This scheme uses a single AA to manage a huge number of users' attribute set, which easily causes congestion and reduces the system efficiency, besides, it doesn't conform to the actual working process of the division of cooperation. To improve the efficient of signal AA, *Chase* [12] proposed Multi-authorities Attribute-based Encryption (MA-ABE) scheme, where each AA manages a part of attribute set on his own and generates the corresponding private key components. To resist collusion attack, the attribute authorities can not communicate each other and the trusted Center Authority (CA) manages all attribute authorities together. However, once the CA is broken, the whole system will collapse, it greatly limits the security and practicability of system. In order to avoid the defects, *Lin* et al. [13] proposed a scheme that successfully removes the CA with the help of the distributed key generation protocol and joint zero secret sharing protocol which made the system no longer depend on an honest trusted CA. But, to avoid internal users' collusion, the scheme requested that the number of User Identity(UID) must be no more than the number of trusted AA, it leads the system can't add any Attribute Authorities. Besides, the scheme doesn't describe any about attribute revocation description, it severely limits the practical applications. The papers [14, 15] proposed the hierarchical attribute-based encryption scheme, which only realized the hierarchical of user structure, but, this scheme ignores both the hierarchical relationship and management model of Certification Authorities. So the paper [16] defined the hierarchical relationship and management model of Certification Authorities, but it just supports "and" policy and limits the flexibility of share information.

As for the disadvantages of the above schemes, this paper proposes a multi-authority revocable ABE scheme, where the attribute authorities classification manage users' attribute set, it doesn't effect the attribute authorities extensibility and can be secure even the CA is broken. The scheme can effective resistance collusion attack among users on the internet and improve the security of internet. It can be suitable for wireless or mobile network security mechanisms, such as on the mobile cloud, sharing information can be safely and efficiently. Besides, to improve the scalability of the system, the scheme put forward the attribute revocation mechanism and implements effective system attributes revocation.

II. PRELIMINARIES

A. Bilinear Maps

Let G_0 and G_1 be two multiplicative cyclic groups of prime order P. Let g be a generator of G_0 and e be a bilinear map, $e : G_0 \times G_0 \rightarrow G_1$. The bilinear map e has the following properties:

This work is supported in part by the National Science Foundation under Grant 61303230.

978-1-4799-9964-4/15/$31.00 ©2015 IEEE

1) Bilinearity: $\forall a, b \in Z_P, \forall f, h \in G_0$, we have $e(f^a, h^b) = e(f, h)^{ab}$;
2) Non-degeneracy: $\exists f \in G_0$, we have $e(f, f) \neq 1$;

We say that G_0 is a bilinear group if the group operation in G_0 and the bilinear map are both efficiently computable. Notice that the map $e : G_0 \times G_0 \rightarrow G_1$ is symmetric.

B. (t, n)-Secret Sharing Scheme

The main idea of [6, 7] is that a secret a is divided into n shares in a such way that any subset of t or more shares can reconstruct the secret, but no subset of fewer than t shares can. The scheme is based on polynomial interpolation where a $t-1$ degree polynomial, $f(x)$, is uniquely defined by t points (x_i, y_i).

1) Setup:

- Randomly choose a secret $a \in Z_P$;
- Set $b_0 := a$;
- Choose $t-1$ random coefficients $b_1, b_2, \ldots, b_{t-1} \in Z_P$ and define $f(x) = \sum_{i=0}^{t-1} b_i x^i$;
- Compute $s_i = f(i) \bmod P$ and let s_i be the i-th share of the secret;

2) Secret reconstruction: For a (t, n)-secret sharing scheme, let $S \subseteq \{1, \ldots, n\}$ denote any subset that contains t value. Then, the function $f(x)$ can be reconstructed using the t-shares s_i where $i \in S$, from the following Lagrange interpolation

$$f(x) = \sum_{i \in S} s_i \cdot \triangle_{i,S}(x)$$

where $\triangle_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. So, thee secret can be obtained by $f(0) = b_0 = a$.

C. Multi-authority ABE

A Multi-authority ABE system consists of K attribute authorities and a central authority, Each attribute authority is also assigned a value s_k. The system uses the following algorithms:

Setup(κ). The setup algorithm takes as input a security parameter κ, and outputs a system public key PK, master secret key MK, and also outputs secret key for each of attribute authorities.

Attribute Key Generation. A randomized algorithm run by an attribute authority. It takes as input the authority's secret key, a user's ID, the authority's value s_k, and a set of attributes A_u in the authority's domain. (We will assume that the user's claim of these attributes has been verified before this algorithm is run). Output secret key for the user.

Central Key Generation. A randomized algorithm run by an attribute authority. Takes as input the master secret key and a user's ID and outputs secret key for the user.

Encryption. A randomized algorithm run by a sender. Takes as input a message, an access tree representing an access structure, and the public key. The algorithm will return the ciphertext that only users who have the secret key generated from the attributes that satisfy the access tree will be able to decrypt the message.

Decryption. A deterministic algorithm run by a user. Takes as input a ciphertext, and decryption keys associated with the access tree, and outputs a message only when the user's attributes satisfy the access tree.

As in [4] and [12], our scheme is proved secure in the non-adaptive security model, where the adversary must provide the challenge access tree he wishes to attack before receiving the public parameters from the challenger. Let κ be the security parameter. We require that the number of authorities, K, and the number of attributes monitored by each authority, AA_k, be upper bounded by a number n which is polynomial in κ. Suppose that the adversary in the *Init* phase chooses the challenge access tree τ^*. In *Phase1*, the adversary can make secret key requests for any attribute set ω with the restriction that ω does not satisfy the access tree.

The game is carried out between a challenger and an adversary, where the challenger simulates the protocol execution and answers queries from the adversary. Specifically, the game is as follows:

Init. The adversary chooses the challenge access tree τ^* and sends it to the challenger.

Setup. The adversary sends a list of attribute sets $A_C = A_C^1 \cdots A_C^K$, one for each authority. The challenger generates parameters for the system and sends them to the adversary. This means the system public key, public keys for all honest authorities, and secret keys for all corrupt authorities.

Phase1. The adversary makes as many secret key queries to the attribute authorities or the central authority, with the restriction that the attribute set ω cannot satisfy τ^*. The challenger returns corresponding secret keys.

Challenge. The adversary sends to the challenger two equal-length messages M_0, M_1. The challenger picks a random bit $b \in \{0, 1\}$, computes the encryption of M_b for access tree ω, and sends this ciphertext to the adversary.

Phase2. The adversary may make more secret key queries with subjection to the requirements described above.

Guess. The adversary outputs a guess $b' \in \{0, 1\}$.

The advantage of an adversary in this game is defined as $Pr[b' = b] - 1/2$.

D. Decisional Bilinear Diffie-Hellman Assumption

Suppose that $a, b, c, z \in Z_p$ are chosen randomly at uniform and G_0 is a group of prime order P with generator g. The Decisional Bilinear Diffie-Hellman (DBDH) assumption means that no polynomial-time adversary can be able to distinguish the tuple $(g^a, g^b, g^c, e(g, g)^{abc})$ from the tuple $(g^a, g^b, g^c, e(g, g)^z)$ with non-negligible probability.

III. Our Construction

A. The construction

The concrete construction is described as follows: G_0 and G_1 are two multiplicative cyclic groups of prime order P, g is a generator of G_0 and e is a bilinear map, $e : G_0 \times G_0 \rightarrow G_1$. Choose a random function $F : S \times ID \rightarrow Z_P$ and a hash function $H(x)$ to implement user's attribute string map to a random element of Z_P.

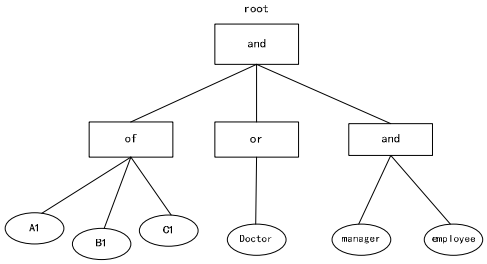

Fig. 1. Access strategy structure

- **Setup**: The CA chooses k seeds $\{s_1, s_2, \ldots, s_k\} \in Z_P$ for every attribute authorities, and sends s_i to the i-th attribute authority. Each attribute authority randomly and independently chooses $y_i \in Z_P$, computes $Y_i = e(g, g_{y_i})$ and sends Y_i to the CA. The CA computes $e(g, g_y) = \prod_{i=1}^{k} Y_i = e(g, g)^{\sum_{i=1}^{k} y_i}$. Output the system public key: $PK = (G_0, g, h = g^{\beta}, f = g^{1/\beta}, e(g, g)^{y})$, the secret key of CA: $MK = (s_1, s_2, \ldots, s_k, \beta)$, and the secret for each of attribute authorities AA_i: $MK_i = (s_i, y_i, \beta)$.
- **The k-th Attribute Authority(AA_k)**: Input MK_k, user's ID and user attribute set A_u, it generates the private key component SK_k as follows: Randomly choose $r \in Z_P; r_j \in Z_P$ for $\forall j \in A_u$, and use the random function $F(x)$ to define $y_{k,u} = F_{s_k}(ID)$. The private key component is: $SK_k = (D = g^{(y_k + y_{k,u} + r)/\beta}$, $\forall j \in A_u$: $D_j = g^r H(j)^{r_j}$, $D_j^{'} = g^{r_j})$.
- **The Center Authority(CA)**: Input MK, use the function F, define $y_{k,u} = F_{s_k}(ID)$, produce the user private key component: $D_{CA} = g^{\sum_{k=1}^{K} (y_{k,u}/\beta)}$.
- **Encrypt(PK, M, T)**: Encrypt according *Consent Control by Modular Addition scheme* and (t, n) *Secret Sharing Scheme*, the concrete operator is as follows:
  1) For an access strategy tree T (as in Fig. 1), it works as follows:
     Randomly choose $a \in Z_P$ as the root node of T and assign all child nodes as un-assigned, then recursively distribute shared secret key to un-assigned non-leaf node of T as follows:
     If the node is "of" and its child nodes are un-assigned: divide the secret key accord (t, n) Secret Sharing Scheme where $t \neq n$ and n is the total number of child nodes, t is the number of child nodes necessary to reconstruct the secret, produce the ith child node's secret key is $a_i = f(i)$ and assign it.
     If the node is "and" and its child nodes are un-assigned: divide the secret key accord (t, n) Secret Sharing Scheme where $t = n$, produce the ith child node's secret key is $a_i = f(i)$ and assign it.
     If the node is "or" and its child nodes are un-assigned: divide the secret key accord (t, n) Secret Sharing Scheme where n is the total number of child nodes and $t = 1$, produce the i-th child node's secret key is $a_i = f(i)$ and assign it.
  2) The ciphertext CT is: CT=$(T, \tilde{C} = Me(g, g)^{ya}$, $C = h^a$, $\forall x \in T : C_x = g^{a_x}$, $C'_x = H(x)^{a_x})$.
- **Decrypt(SK_i, D_{CA}, CT)**: If the user attribute set A_u does not satisfy access strategy T, return $\perp$; Otherwise, choose the smallest attribute set S_x, calculate $F_{DES} = \frac{e(D_i, C_x)}{e(D'_i, C'_x)} = e(g, g)^{ra_x}$, and $F_x = \prod_{z \in S_x} F_{DES}^{\triangle_{i,s'_x}(0)} = e(g, g)^{ra}$ by interpolation. Calculate $Y = e(C, D) = e(g, g)^{(y_{k,u} + r + y_k)a}$, $\frac{Y}{F_x} = e(g, g)^{(y_{k,u} + y_k)a}$. Calculate $Y' = e(g, g)^{a(\sum_{k=1}^{K} y_k + \sum_{k=1}^{K} y_{k,u})}$, Y_{CA}=$e(D_{CA}, C) = e(g, g)^{a(\sum_{k=1}^{K} y_{k,u})}$, let $F' = \frac{Y'}{Y_{CA}} = e(g, g)^{ay}$, recover the message $M = \tilde{c}/F'$.

B. Revocation Scheme

This scheme mainly aims to achieve system attribute revocation. Suppose that the data owner holds the randomness a used in the ciphertext. The ciphertext updates is done by the data owner himself. Firstly, remove the revocable attribute from access strategy T, then recalculate the shared secret key a_x of access strategy leaves, and calculate $\{C_x = g^{a_x}$, $C'_x = H(x)^{a_x}\}_{\forall x \in T}$ instead of the corresponding part of cipher-text.

The concrete method is described as follows:

Attribute revocation is implemented by attribute authorities and the information owner. There are a synchronization array $List()$, between attribute authorities and the center authority. Besides, the attribute authorities all have a array to count the number of users, sign as $Account(A_{i,j}) = t$ (i: AA_i; j: attribute j; t: the number of registered user; $1 \leq i \leq K, 1 \leq j \leq n$). When the attribute $A_{i,j}$ is revoked, push $A_{i,j}$ to the array $List()$ and sign $List(A_{i,j}) = t$ where t is the total number of registered users. What's more, the concrete process is as follows:

The owner of shared information should remove the revocation attribute from the access strategy T. Then recalculate the shared secret a_x for every AA, and calculate $\{C_x = g^{a_x}$, $C'_x = H(x)^{a_x}\}_{\forall x \in T}$ to instead of the corresponding part of cipher-text.

When a user login and download share information: If the revocation array $List()$ is null or none of the user's attribute set A_u in the $List()$, continue. Otherwise, renew the user's private key and send it to the user, then do $List(A_{i,j}) = t$. If $t = 0$, remove the attribute $A_{i,j}$ from $List()$. This scheme can achieve the system attribute revocation without increasing the length and encryption complexity of cipher-text.

IV. Analysis

A. Security Analysis

Theorem 1. *If an adversary can break our construction with non-negligible advantage in the security model defined in Section II-D, then a simulator can be constructed to solve the DBDH problem.*

Proof: Suppose there exists a polynomial-time adversary A, that can attack our scheme with a non-negligible advantage

ε. We build a simulator S, that can play the DBDH game with advantage $\varepsilon/2$. The simulation proceeds as follows:

There is given a description of groups G_0, G_1 with an efficient bilinear map, e and generator g and a DBDH instance $(g^a, g^b, g^c, e(g,g)^z)$, where $z = abc$ or random. The simulator works as follows:

- **Init**. The adversary $\mathcal{A}$ selects a challenge access strategy T^*, and sends it to the simulator S.
- **Setup**. The simulator sets parameter $Y = e(A,B) = e(g,g)^{ab}$. it chooses a random parameter $\beta \in Z_P$, sets parameters $h = g^\beta, f = g^{1/\beta}$. Then it sends the public parameters to the adversary $\mathcal{A}$.
- **Phase 1**. The adversary A makes request for the private keys where attribute set w_i=$\{a_i | a_i \in \Omega \cap a_i \notin T^* \cup ID\}$ from simulator. The simulator selects a random function F_{s_k} for the Attribute Authority k (AA_k), it sets parameter $y_{k,u} = F_{s_k}(ID)$ where the ID is user's unique identify. Choose random parameters $r, s_k \in Z_P$, it sets parameter $D = g^{(y_k + y_{k,u} + r)/\beta}$. If the attribute $a_i \in w_i$, it sets parameters $D_{a_i} = g^r H(a_i)^{r_{a_i}}$, $D'_{a_i} = g^{r_{a_i}}$. It then sends the private key to the adversary.
- **Challenge**. The adversary $\mathcal{A}$ submits a challenge attribute set and two challenge messages M_0, M_1 to the simulator. Assume that the adversary never requests for the private key of the challenge set in Phase 1. Now, the simulator randomly chooses a bit, $b \in \{0,1\}$, then encrypts the message M_b: $CT = (T^*, \tilde{C} = M_b Z, C = h^\alpha, \forall i \in T^* : C_i = g^{\alpha_i}, C'_i = H(i)^{\alpha_i})$. Set the root node α, for the challenge tree T^*, encrypt as described in Section III-A. The simulator sends the cipher-text to the adversary A. If $Z = e(g,g)^{abc}$, we implicitly set $\alpha = c$, i.e., $Y^\alpha = Z = e(g,g)^{abc}$ and $C = h^c$, $C_i = g^{c_i}$. It indicates that the cipher-text is a valid random encryption of message M. Otherwise, if $Z = e(g,g)^z$ for a random z, $\tilde{C} = M_b e(g,g)^z$. Since z is random, CT will be a random element of G_1 from the adversary's point of view and the ciphertext contains no information about M_b.
- **Phase 2**. The simulator acts exactly as it did in *Phase 1*.
- **Guess**. The adversary submits a guess b' of b. If $b' = b$, the simulator will output 0, indicating that $Z = e(g,g)^{abc}$; otherwise the simulator outputs 1, indicating that the adversary gains no information about plain-text M_b.
  1) If $b' \neq b$, then we have $Pr[b' \neq b | Z = e(g,g)^z] = 1/2$;
  2) If $b' = b$, define the adversary' advantage as ε. We have $Pr[b' = b | Z = e(g,g)^{abc}] = 1/2 + \varepsilon$;
  3) So, the advantage of the simulator in the DBDH game is: Adv = $Pr[b' = b] - 1/2$ = $1/2 \cdot (Pr[b' = b | Z = e(g,g)^{abc}] + Pr[b' = b | Z = e(g,g)^z])$-1/2 = $\varepsilon/2$.

Only when the attribute set satisfies the access strategy T^*, the user can decryption and see the plain-text in this scheme. The scheme is security only when the communication is security between agency and the user. □

B. Efficiency Analysis

In this section, we compare the efficiency of our scheme and [4, 12], Both ours and [12] belong to multi-authority ABE, where the shemes prevent the collusion among users with user's ID and decryption has nothing to do with ID. Both ours and [4] belong to CP-ABE. The performance analysis and comparison results are shown in Table II. Let e be a bilinear map, $e : G_0 \times G_0 \rightarrow G_1$. The notations used in Table II are explained in Table I.

TABLE I. NOTATIONS

Notations	Definition
N	The amount of attributes
P	The generator of group
A_k	The attribute set of management
A_u	User's attribute set
A_c	Encrypt attribute set
L	Length

TABLE II. PERFORMANCE ANALYSIS AND COMPARISON

Type	Signal AA	Multiple AA	
Scheme	Scheme[4]	Scheme[12]	Our Scheme
AA_i secret key	$L_{Z_P} + L_{G_0}$	$(N+1)L_{Z_P}$	$2L_{Z_P}$
User secret key	$(2A_u + 1)L_{G_0}$	$(A_u + 1)L_{G_0}$	$(2A_u + 2)L_{G_0}$
Cipher-text	$(2A_c + 1)L_{G_0}$ + L_{G_1}	$(A_c+1)L_{G_0}$ + L_{G_1}	$(2A_c+1)L_{G_0}$ + L_{G_1}
Policy	and, or, threshold	threshold	and, or, threshold
Advantages	fine grained access control;	AA classification manages user attribute set;	·fine grained access control; ·AA classification manages user attribute set; ·support AA extension and remove; ·attribute revocation;
Disadvantages	·Attribute revocation expensive; ·single AA manages all users attribute set;	only support threshold policy	— —

As shown in Table II, the ABE scheme of [4] is signal authority while ours is multi-authority. Nevertheless, it does not increase cipher-text size, effectively relieves the single organization management burden and improves the flexibility management of attribute set. Compared with the scheme [12], our scheme improves the flexibility management of attribute set with the acceptable for the cipher-text size.

V. CONCLUSION

Recently, the problem that securely and efficiently shares information has aroused widely concern. The ABE scheme can satisfy this scenario. This paper proposes an multi-authority ABE scheme with revocation. This scheme designs multiple attribute authorities classification to manage user attribute set,

which can effectively relieves the single AA management burden. Besides, this scheme improves the system performance and scalability without affecting the attribute authority extension. Revocation mechanism let the information owner do effective system attributes revocation. This paper also gives the concrete description and accuracy analysis.

REFERENCES

[1] A. SHAMIR: Identity-based cryptosystems and signature schemes. In: Proceedings of cryptology 84 on Advances in cryptology, pp. 47–53. Springer-Verlag, Heidelberg, Berlin (1985)

[2] A. SAHAI and B. WATERS: Fuzzy identity-based encryption. In: Proceedings of the Advances in Cryptology, pp. 457–473. Springer-Verlag, Heidelberg, Berlin(2005)

[3] V. GOYAL, O. PANDEY, A. SAHAI, et al.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89–98. ACM Press, New York (2006)

[4] J. BETHENCOURT, A. SAHAI and B. WATERS: Ciphertext-Policy attribute-based encryption. In: Proceedings of the 2007 IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Computer Society, Washington (2007)

[5] B. Waters: Ciphertext-Policy attribute-based encryption: An expressive, efficient, and provably secure realization, http://eprint.iacr.org/2008/290.pdf

[6] L. IBRAIMI, Q. TANG, P. HARTEL, et al.: Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes. In: Proceedings of the 5th International Conference, 1–12, Xi'an (2009)

[7] A. SHAMIR: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)

[8] M. PIRRETTI and P. TRAYNOR: Secure attribute-based systems. Journal of Computer Security 18(5), 799 – 837 (2010)

[9] F. LIU and M. YANG: Ciphertext policy attribute based encryption scheme for cloud storage. Application Research of Computers 29(4), 1452–1456 (2012)

[10] J. X. WANG, M. ZHANG and Q. CHEN: An efficient attribute based encryption with attribute revocation. Computer Application 32(S1), 39–43 (2012)

[11] Y. W. DUAN and B. LANG: Extended ciphertext-policy attribute based encryption scheme. Journal of huazhong university of science and technology 40(1), 113–115 (2012)

[12] M. CHASE: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)

[13] H. LIN, Z. F. CAO, X. H. LIANG, et al.: Secure threshold multi authority attribute based encryption without a central authority. Information Sciences 180, 2618–2632 (2010)

[14] Z. G. WAN, J. Liu and R. H. Deng: HASBE:A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Transactions on Information Forensics and Security(TIFS) 7(2), 743-754 (2012)

[15] X. B. ZOU: A hierarchical attribute-based encryption scheme. Wuhan University Journal of Natural Sciences 18(3), 259-264 (2013)

[16] Q. Y. AI, D. TONG, Z. X. WANG and J. W. LIU: A hierarchical certification attribute-based encryption scheme. Wuhan University Journal of Natural Sciences 60(5), 441-446 (2014)