Low Complexity Multi-Authority Attribute Based Encryption Scheme for Mobile Cloud Computing

Fei Li, Yogachandran Rahulamathavan, Muttukrishnan Rajarajan
School of Engineering and Mathematical Sciences
City University London, London
United Kingdom, EC1V 0HB
{fei.li.1, yogachandran.rahulamathavan.1, r.muttukrishnan}@city.ac.uk

Raphael C.-W Phan
Faculty of Engineering
Multimedia University
63100 Cyberjaya, Malaysia
raphael@mmu.edu.my

***Abstract*—In multi-authority attribute based encryption (MA-ABE) schemes, multiple authorities monitor different set of attributes and the corresponding shared-decryption-keys. In such schemes the encryptor can encrypt a message and upload to the cloud, whereas only users who satisfy a set of attributes from each authority can decrypt that message. We extend the well-known Chase and Chow's scheme [8] for mobile users by introducing a cloud based semi-trusted-authority (STA) between the mobile user and attribute authorities. In this work, substantial amount of communications and computations are outsourced to the STA without compromising the security and privacy of the MA-ABE scheme. The STA interacts with the attribute-authorities on behalf of the user and obtains the masked shared-decryption-keys. Later the STA combines all the keys and gets one masked-key which can only be unmasked by a user to decrypt the message. In particular, STA cannot decrypt the message nor determine the attributes of the mobile user, hence, the security and privacy of the proposed MA-ABE scheme is preserved.**

***Index Terms*—Mobile cloud computing; privacy; attribute-based encryption; access control.**

I. Introduction

Cloud computing has drawn great attention from both academia and industry. It provides computing infrastructure as services over the Internet and combines series of new technologies in order to provide a better service and user experience. Today, increasing number of enterprisers deploy their services over the cloud environment. As a result, users can gain access to their resources from anywhere. However, new security and privacy issues are raised due to the untrusted nature of cloud service providers. The cloud service providers may give access to third parties, who do not have the access permission, for commercial gain. Hence, it is crucial to secure the confidential personnel data in the cloud storage.

In recent years, electronic health record (EHR) systems are gradually deployed in healthcare. It improves physician efficiency, reduce costs, and medical errors, improve data availability and sharing [1]. In health application scenario, user usually uploads his personal health information (PHI) to the cloud storage. Hence, the usage of the PHI in the cloud should satisfy the related laws and legislation, such as Health Insurance Portability and Accountability Act (HIPAA) in the US and the EU Data Protection Directive [2] in Europe. The main concern is on how the data owner (patient) can control personal data stored on the third party's cloud storage in order to securely share or make the data unreadable to all others including the cloud provider.

In the healthcare application scenario, the patients should have the privilege not only to decide how to encrypt his/her PHI files, but also on who are authorized to gain access to the files. The data requester should satisfy both requirements to decrypt the files. Furthermore, the patients should also be able to revoke access to certain users if necessary. The attributes based encryption (ABE) has been considered as the suitable cryptographic technology for the cloud environments. It was firstly proposed by Sahai and Waters in [3], where they constructed an identity based encryption (IBE) of a message under several attributes that compose a fuzzy identity. There are two main types of ABE namely key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE).

In the KP-ABE scheme, private keys are associated with an access structure and the ciphertext is labeled with a set of attributes. When the access structure defined in the private key matches the attributes labeled with the ciphertext, then it decrypts the ciphertext [4]. Different from the KP-ABE scheme, in the CP-ABE system, access structure is assigned to the ciphertext and each private key are associated with a set of attributes. A user must have the private key to satisfy the policy in order to decrypt the ciphertext [5]. Using ABE, the data owner can enforce access policies based on the users' attributes. For instance, the patient uploads the encrypted PHI files to the cloud server using ABE where the access policy can be defined as follows: only user who has the attribute "Doctor" issued by "Hospital A" can be able to decrypt the PHI.

The management of attributes is a critical issue in the ABE systems. There can be multiple attribute-authorities (AAs) which are responsible for attribute management and key distribution. Chase [6] presented a multi-authority ABE (MA-ABE) scheme, which allows any polynomial number of independent authorities to monitor attributes and distribute secret keys. The data owner chooses a number, i.e. d_k for $k^{th} AA$, and a set of attributes from each AA, and encrypts a message. This encrypted message can be

978-0-7695-4944-6/12 $26.00 © 2012 IEEE
DOI 10.1109/SOSE.2013.12

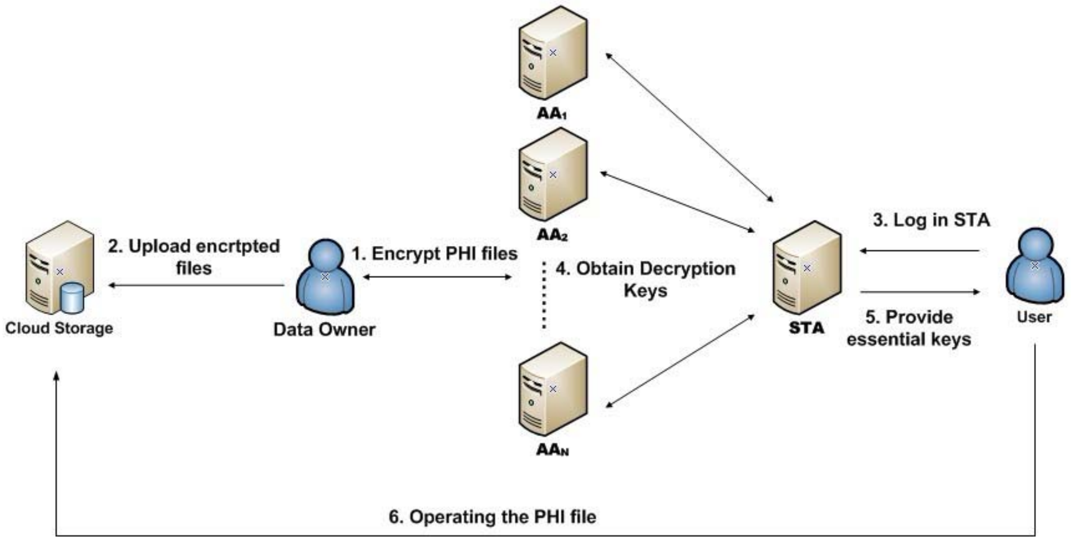
AA₁
AA₂
AA_N
2. Upload encrtpted files
1. Encrypt PHI files
4. Obtain Decryption Keys
3. Log in STA
5. Provide essential keys
Cloud Storage
Data Owner
STA
User
6. Operating the PHI file

Fig. 1. The proposed framework of multi-authority ABE for mobile cloud environment

decrypted only by users who satisfy d_k number of attributes from the k^{th} AA where $k = 1, \ldots, N$. However, in [6], a trusted central authority (CA) is needed for distributing all the keys. The CA is powerful enough to decrypt all the messages. The improved MA-ABE scheme without CA has been presented by Chase and Chow in [8]. In [8], each pair of AAs securely exchange a shared secret among them during the setup process. Chase and Chow also proposed an anonymous key issuing protocol, hence users can obtain secret keys from AAs without revealing their global identities (GIDs). Hence, decryptor (user) interacts with all the AAs in order to obtain the shared-decryption-keys. The interactions between the user and AAs are a burden to the mobile user in terms of communication and computational complexity.

In this paper, for the first time, we propose a MA-ABE scheme for mobile users. In particular, our scheme substantially reduces the communication and computation workload to the user compare to the Chase and Chow scheme in [8]. Hence, our scheme is suitable for resource constrained devices such as mobile-phones in order to access the cloud infrastructure. Mobile cloud computing (MCC) serves mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms [7]. For example, patient can use his mobile-phone to encrypt and upload the PHI file to the cloud storage. The doctor could also use their portable devices to decrypt the file.

We introduce a cloud server based semi-trusted-authority (STA) between a mobile user and AAs. Mobile user only provides pseudonym of his identity to the STA. Then the STA interacts with all the AAs on-behalf of mobile user and obtains masked shared-decryption-keys. Later, STA combines all the keys and obtains a masked-key and pass it to the mobile user. Mobile user unmasks the received key and gets the decryption-key, which can be used to decrypt the encrypted message. Since all the distributed-keys provided by AAs to STA are masked, STA cannot decrypt the message. Moreover, STA cannot pool all the keys and obtain attributes of mobile user, hence, our algorithm preserve the security and privacy of Chase and Chow's MA-ABE scheme while outsourcing the computational and communication overhead to the STA. We assume that the STA will execute the protocol correctly in order to maintain his reputation, hence we assume that he will behave in a semi-honest manner, i.e. he is honest but curious so privacy is a real issue.

II. Preliminaries

This section describes the framework and security model of MA-ABE scheme for mobile devices in cloud environment.

A. System Model

There are five different parties involved in our framework, the data owner (encryptor), the user (decryptor), AAs, STA and cloud server. Fig. 1 depicts the main framework and work flow of the proposed system. The AAs stores user's attributes. Hence, the AA will know which subsets of its attributes are held by different users. Based on the work in [8], anonymous credentials are implemented for achieving privacy of users.

B. Background Knowledge

1) Bilinear Pairings: Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two multiplicative groups of prime order q, generated by g_1, g_2 respectively. A bilinear map is denoted as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, where it has the following three properties.

1 Bilinearity: $\forall x \in \mathbb{G}_1, \forall y \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_q$, there is $\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}$.

2 Non-degeneracy: For $\forall x \in \mathbb{G}_I, \forall y \in \mathbb{G}_2$, there is $\hat{e}(x, y) \neq 1$.

3 Computability: $\hat{e}$ is an efficient computation.

2) Complexity Assumptions: Definition 1. The Decisional Diffie-Hellman (DDH) Assumption

The Decisional Diffie-Hellman (DDH) problem in prime order group $\mathbb{G}$ is defined as follows;

Let us assume g is the generator of $\mathbb{G}$. For the input $g, g^a, g^b, g^c \in \mathbb{G}$, and a, b, c $\in \mathbb{Z}_q$, decide if $c = ab$ or c is a random element of $\mathbb{Z}_q$.

Definition 2. The Decisional Bilinear Diffie-Hellman assumption

Let $\forall a$, b, $c \in \mathbb{Z}_q$, g is the generator of $\mathbb{G}_1$. The Decisional BDH problem is defined as there is no probabilistic polynomial-time algorithm $\mathfrak{B}$ that can distinguish the tuple (A = g^a, B = g^b, C = g^c, $\hat{e}(g, g)^{abc}$) from the tuple (A = g^a, B = g^b, C = g^c, $\hat{e}(g, g)^z$) with more than a negligible advantage. The advantage $\mathfrak{B}$ is

$Pr[\mathfrak{B}(A, B, C, \hat{e}(g, g)^{abc}) = 0] - Pr[\mathfrak{B}(A, B, C, \hat{e}(g, g)^z)] = 0$, where the probability is taken over the random choice of the generator g, the random choice of a, b, c, z in $\mathbb{Z}_q$, and the random bits consumed by $\mathfrak{B}$.

C. Chase and Chow's MA-ABE Scheme

In this section we briefly describe the steps involved in Chase and Chow's MA-ABE Scheme [8] below. Hence, we will extend this scheme suitable for mobile cloud computing in the next section.

Setup

There are several initializations to be done in the setup stage.

- For a given security parameter λ and $\sigma \in \{0, 1\}^{poly(\lambda)}$, group bilinear parameters are generated by the authorities as
$$q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \leftarrow BDH\_Gen(1^\lambda; \sigma).$$
- Collision-Resistant Hash Function (CRHF) $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$. CRHF can be used to generate user identity u from the user global identity (GID).

Now, AAs interact with each other and execute the following:

- k^{th} AA chooses a random $v_k \in_R \mathbb{Z}_q$ and computes $Y_k = \hat{e}(g_1, g_2)^{v_k}$, and sends Y_k to the other AAs. Finally each AA computes $Y = \prod Y_k = \hat{e}(g_1, g_2)^{\sum_k v_k}$.
- Each pair of AAs shares a secret, k^{th} AA and j^{th} AA choose a random $s_{kj} \in \mathbb{Z}_q$ such that $s_{kj} = s_{jk}$, which is only known to themselves.
- k^{th} AA chooses a random $x_k \in \mathbb{Z}_q$ and computes $y_k = g_1^{x_k}$. Using the shared secret $s_{k,j}$ and u, AAs k and j computes $y_k^{x_j/(s_{kj}+u)}$ and $y_j^{x_k/(s_{kj}+u)}$, respectively.

Now each AA carries out the following steps individually.

TABLE I
PRIVATE AND PUBLIC KEYS IN CHASE AND CHOW'S MA-ABE SCHEME.

	Keys
k^{th} AA's private keys	1. v_k 2. x_k 3. $t_{k,i},\ i \in [1, ..., n_k]$
k, j^{th} AAs's private keys	1. $s_{k,j},\ j \in \{1, ..., N\}/\{k\}$ 2. $y_k^{x_j/(s_{kj}+u)} = y_j^{x_k/(s_{kj}+u)}$
System public keys	1. $Y = \hat{e}(g_1, g_2)^{\sum_k v_k}$ 2. $y_k = g_1^{x_k}$ 3. $T_{k,i} = g_2^{t_{k,i}},\ i \in [1, ..., n_k]$

- AA now generates the public key and private key for each attribute.
- For the i^{th} attribute stored in the k^{th} AA, where $i \in \{1, ..., n_k\}$ and $k \in \{1, ..., K\}$, the k^{th} AA chooses random $t^{k,i} \in \mathbb{Z}_q$.
- Computes $T_{k,i} = g_2^{t_{k,i}}$.

Table I shows all the public and private keys of the Chase and Chow's MA-ABE scheme.

Key Issuing

In order to get the distributed decryption-keys, the user has to do the following interaction with all the authorities. For authority k, the steps are as follows:

- For $j \in \{1, ..., N\} / \{k\}$, user u gets the $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ for $k > j$ or $D_{kj} = g_1^{R_{kj}} y_k^{(s_{kj}+u)/x_j}$ if $k < j$, where $R_{k,j} \in \mathbb{Z}_q$ is a random value.
- Authority k randomly picks a degree d_k polynomial $p_k(\cdot)$ with $p_k(0) = v_k - \sum_{j \in \{1,...,N\}/\{k\}} R_{kj}$.
- Authority k compute $S_{k,i} = g_1^{p(i)/t_{k,i}}$ for each eligible attribute i for the user u.
- After receiving the D_{kj}, user u computes $D_u = \prod_{(k,j) \in \{1,...,N\} \times \{1,...,N\}/\{k\}} D_{kj} = g_1^{R_u}$, where $R_u = \sum_{(k,j) \in \{1,...,N\} \times \{1,...,N\}/\{k\}} R_{kj}$.

Encryption

In order to encrypt message m for attribute set $\{\mathbb{A}_1^C, ..., \mathbb{A}_N^C\}$, encryptor picks a random $s \in_R \mathbb{Z}_q$, return
$$\left\langle E_0 = mY^s, E_1 = g_2^s, \left\{C_{k,i} = T_{k,i}^s\right\}_{i \in \mathbb{A}_k^C, \forall k \in [1,...,N]} \right\rangle.$$

Decryption

- For each authority k:

  1 For any attributes $i \in \mathbb{A}_k^C \cap \mathbb{A}_k^u$, user computes $\hat{e}(S_{k,i}, C_{k,i}) = \hat{e}(g_1, g_2)^{sp_k(i)}$.

  2 Interpolate all the $\hat{e}(g_1, g_2)^{sp_k(i)}$ and gets $P_k = \hat{e}(g_1, g_2)^{sp_k(0)} = \hat{e}(g_1, g_2)^{s\left(v_k - \sum_{j \neq k} R_{kj}\right)}$.
- Multiply all the P_ks together and obtain
$$Q = \hat{e}(g_1, g_2)^{s(\sum\{vk\} - R_u)} = Y^s / \hat{e}\left(g_1^{R_u}, g_2^s\right).$$
- Compute $\hat{e}(D_u, E_1) \cdot Q = \hat{e}\left(g_1^{R_u}, g_2^s\right) \cdot Q = Y^s$.
- Recover message m by E_0/Y^s

III. Modified MA-ABE Scheme for Mobile Cloud Computing

In the previous section, we briefly described the Chase and Chow's MA-ABE scheme. It is obvious that the user has to request essential keys from all the AAs in order to get the decryption key. In the mobile environment, it is not practical for mobile devices to do such work. Firstly, such computational work is a heavy payload for mobile devices. Secondly, the mobile data network may not be reliable for communication with all AAs and it can cause large overheads and communication delays. In our proposed MA-ABE scheme, we utilize the cloud services to offload the computation and communication of the original MA-ABE model from the user's end to the cloud without compromising the security and privacy. Hence, we improve the efficiency of the whole procedure, hence, the new framework is more suitable for mobile environment.

We introduce a STA between the mobile user and AAs. STA can be a cloud based server with powerful computation ability and direct connection to all AAs. In the setup stage, we consider STA as an authority, which shares the public keys of the system (see Table I). In the following section we explain the steps involved in our method.

A. Proposed Construction

We extend the Chase's MA-ABE model for mobile environments. The proposed framework has the same setup stage as in the Chase and Chow's MA-ABE scheme, however, now STA also shares the public parameters. We explain each step of our scheme by comparing with the Chase and Chow's MA-ABE scheme (the steps which are different from Chase and Chow's MA-ABE scheme are denoted as $*$).

Setup

The setup process is the same as in Chase and Chow's MA-ABE scheme.

Key Issuing

The STA executes the following steps with each authority k on behalf of user u, hence the following communication and computational overheads have been offloaded to STA.

- For $j \in \{1, ..., N\} / \{k\}$, STA gets the $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ for $k > j$ or $D_{kj} = g_1^{R_{kj}} y_k^{(s_{kj}+u)/x_j}$ if $k < j$, where $R_{k,j} \in \mathbb{Z}_q$ is a random value.
- After receiving the D_{kj}, STA computes $D_u = \prod_{(k,j)\in\{1,...,N\}\times\{1,...,N\}/\{k\}} D_{kj} = g_1^{R_u}$, where $R_u = \sum_{(k,j)\in\{1,...,N\}\times\{1,...,N\}/\{k\}} R_{kj}$

* If user u satisfies d_k number of attribute, then k^{th} AA randomly picks a degree polynomial $p_k(\cdot)$ with degree d_k.
* If user u doesn't satisfy d_k number of attribute, then k^{th} AA randomly picks a degree polynomial $p_k(\cdot)$ with degree $n_k + 1$, where n_k is the total number of attributes monitored by the k^{th} AA.
* Now, using the pre-shared secret, r_k, between user u and k^{th} AA, define $p_k(0) = v_k + r_k - \sum_{j\in\{1,...,N\}/\{k\}} R_{kj}$.
* Authority k computes $S_{k,i} = g_1^{p(i)/t_{k,i}}$, $i \in [1, \ldots, n_k]$.

Encryption

Encryption step is same as the Chase and Chow's MA-ABE scheme.

Decryption by STA

* For each authority k:
  * STA finds the common attributes set between k^{th} AA and the encrypted message as $i \in \mathbb{A}_k^C \cap \mathbb{A}_k$.
  * Using $S_{k,i}$ and the corresponding $C_{k,i}$, STA computes $\hat{e}(S_{k,i}, C_{k,i}) = \hat{e}(g_1, g_2)^{sp_k(i)}$.
  * STA interpolates all $\hat{e}(g_1, g_2)^{sp_k(i)}$ and gets $P_k = \hat{e}(g_1, g_2)^{sp_k(0)} = \hat{e}(g_1, g_2)^{s\left(v_k + r_k - \sum_{j\neq k} R_{kj}\right)}$.
* STA multiplies all P_k's together and gets $Q = \hat{e}(g_1, g_2)^{s\sum v_k + s\sum r_k - sR_u} = \frac{Y^{s+s\sum r_k}}{\hat{e}\left(g_1^{R_u}, g_2^s\right)}$.
* STA computes $T = \hat{e}(D_u, E_1) \cdot Q = \hat{e}\left(g_1^{R_u}, g_2^s\right) \cdot Q = Y^{s+s\sum r_k}$, then sends T to the user.

Decryption by User

* User computes $\prod \hat{e}(g_1^{r_k}, g_2^s) = \hat{e}(g_1, g_2)^{s\sum r_k} = Y^{s\sum r_k}$.
* In order to recover the message m, user computes $Y^s = T/Y^{s\sum r_k}$, recover m as E_0/Y^s .

B. Security Analysis

In this section, we analyze the security of the proposed MA-ABE scheme. As the proposed scheme is an extension of Chase and Chow's MA-ABE scheme, we show that the proposed scheme does not degrade the security and privacy of the encrypted message and mobile user compared to the original scheme. More precisely, we focus on the stages that differ between the two schemes, namely the Key Issuing stage and the Decryption stage.

During the Decryption stage, the STA performs the steps in place of the authority AA. In more detail, the STA only computes $T = Y^{s+s\sum r_k} = Y^s Y^{s\sum r_k}$ in contrast to the Y^s that is computed by the authority in the Chase-Chow scheme. As the required decryption key to decrypt the message m is Y^s, the STA cannot decrypt to obtain the message m, therefore the confidentiality of the message is ensured. More precisely, since the shared secret r_k is only known to the k^{th} AA and the mobile user, and thus the summation $\sum_k r_k$ can only be obtained by a mobile user; therefore the STA cannot obtain Y^s from its known expression of $T = Y^{s+s\sum r_k} = Y^s Y^{s\sum r_k}$.

During the Key Issuing stage, the STA performs most of the steps in place of the user in the Chase-Chow scheme. The k^{th} AA computes $S_{k,i} = g_1^{p(i)/t_{k,i}}$, $i \in [1, \ldots, n_k]$ and sends them to STA. If the user satisfies the minimum d_k number of attributes, then the degree of the polynomial chosen by AA is equal to d_k. Hence, d_k number of $S_{k,i}$ can be used to get the secret $p_k(0) = v_k + r_k - \sum_{j\in\{1,...,N\}/\{k\}} R_{kj}$ during the interpolation. If the user

TABLE II

COMPARISON OF COMPUTATIONAL COST FOR THE MOBILE USER IN THE CHASE AND CHOW'S SCHEME AND THE PROPOSED SCHEME.

	Chase and Chow's Scheme	Proposed Scheme
Key Issuing	$N(N-1)C_m$	–
Decryption	$Nd_k(C_p+C_e+C_m)+(N+1)C_m$	$(N+1)C_m+NC_p$
Total	$(N^2+Nd_k+1)C_m+Nd_kC_e+Nd_kC_p$	$(N+1)C_m+NC_p$

does not satisfy the minimum d_k number of attributes then the degree of the polynomial chosen by the k^{th} AA is equal to n_k+1. This is the crucial point, because the k^{th} AA sends only n_k number of $S_{k,i}=g_1^{p(i)/t_{k,i}}$, $i \in [1,\ldots,n_k]$ to the STA, where the STA would require n_k+1 number of $S_{k,i}$ to recover the secret $p_k(0)$. Therefore, the STA cannot be able to distinguish which set of attributes belongs to the mobile user, and furthermore cannot be able to pool all $S_{k,i}$'s from all AAs in order to find the attributes of mobile user. This preserves the privacy of the user.

C. Reduction in Computational overhead

In the Chase and Chow's MA-ABE scheme, the user takes part in the computation during the **Key Issuing** and **Decryption** steps. Denote the computational costs of a multiplication, an exponentiation and a pairing in $\mathbb{G}$ as C_m, C_e and C_p respectively.

Table II shows all the necessary computational cost for the user in both the Chase and Chow's scheme and the proposed scheme. It is obvious from Table II that substantial amount of computational task has been offloaded to cloud based STA in the proposed scheme.

D. Reduction in Communication overhead

The communication overhead is always an important factor for mobile environment. In Chase and Chow's work, user needs $N-1$ independent invocations for each authority during the key issuing stage. With the increase of total number of authorities, it can obviously generate a large network overhead. In our scheme, those communications have been leveraged to the cloud server, hence the mobile device is not necessarily involved in the numerous communication requests and responses. Fig. 2 shows the number of communication required for both the Chase and Chow's scheme and our proposed scheme. It is obvious from Fig. 2 that the number of communication required for Chase and Chow's scheme is increasing quadratically with number of AAs while in the proposed scheme it is constant.

IV. CONCLUSIONS

In this paper, we proposed a multi-authority attribute based encryption scheme appropriate for mobile cloud computing. We extended the well known Chase and Chow's multi-authority attribute based encryption scheme by offloading the computation and communication overhead to the cloud based semi-trusted-authority. The mobile user only provides his anonymous identity to the semi-trusted-authority, where the semi-trusted-authority interacts with all the attribute authorities and get a masked decryption key on behalf of the user. Only the user can be able to remove the mask and decrypt the message. We also showed that the proposed scheme preserves the security of the encrypted message and the privacy of the mobile user while providing an efficient access to the Chase and Chow's multi-authority attribute based encryption scheme from the mobile devices.

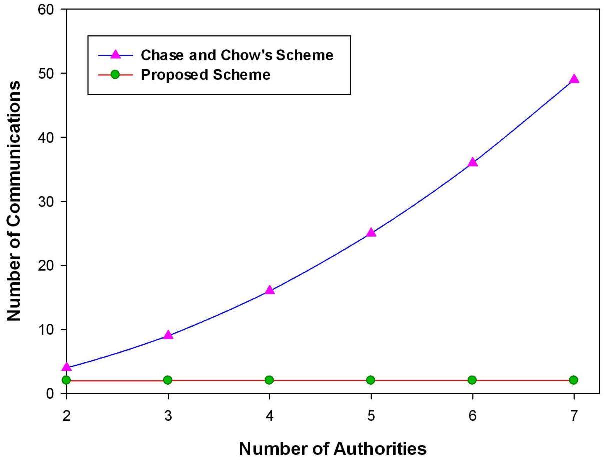

Fig. 2. Comparison of communication overheads in Chase and Chow's scheme and the proposed scheme.

REFERENCES

[1] J. Sun, X. Zhu, C. Zhang, and Y. Fang, HCPP: Cryptography Based Secure EHR System for Patient Privacy and Emergency Healthcare. In 31st Int. Conf. Distributed Computing Systems (ICDCS), pp. 373–382, 20–24 Jun. 2011.

[2] S. Hinde, Privacy legislation: a comparison of the US and European approaches. In Computers and Security, Vol. 22, Issue 5, pp. 378–387, ISSN 0167-4048, Jul. 2003.

[3] A. Sahai, and B. Waters, Fuzzy Identity-Based Encryption. Advances in Cryptology EUROCRYPT, vol.3494, pp. 557, 2005.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Proc. 13th ACM conf. Comp. and Commun. security (CCS '06), New York, USA, pp. 89-98 2006.

[5] J. Bethencourt, A. Sahai, B. Waters. Ciphertext-Policy Attribute-Based Encryption. In IEEE Symposium on Security and Privacy, SP 07, pp. 321–334, May 2007.

[6] M. Chase, Multi-authority Attribute Based Encryption, In LNCS, Berlin Heidelberg, pp. 515–534, vol. 4392, 2007.

[7] Z. Sanaei, S. Abolfazli, A. Gani, M. Shiraz. SAMI: Service-Based Arbitrated Multi-Tier Infrastructure for Mobile Cloud Computing, In IEEE Workshop on Mobile Cloud Computing, Beijing, China, 2012.

[8] M. Chase and Sherman S.M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In Proc. 16th ACM conf. Comp. and Commun. Security (CCS '09), New York, USA, pp. 121-130, 2009.