An Efficient Hierarchical Multi-Authority Attribute Based Encryption Scheme for Profile Matching using a Fast Ate Pairing in Cloud Environment

Balaji Chandrasekaran, Student member, ACM, Yasuyuki Nogami, Member, IEEE and Ramadoss Balakrishnan, Member, IEEE

Abstract—In cloud environment, profile matching is a key technique in applications such as health care and social networks. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a suitable technique for data sharing in such environments. In this paper, we propose an asymmetric pairing based Hierarchical Multi-Authority CP-ABE (HM-CP-ABE) construction for profile matching. We utilize the fast Ate pairing to make the proposed HM-CP-ABE scheme efficient. The performance analysis of the proposed scheme shows improved efficiency in terms of computational costs for initialization, key generation and encryption using ELiPS library when compared with existing works.

Index Terms—Asymmetric Pairing, Attribute based encryption, Cloud computing, CP-ABE, Multi-Authority.

I. INTRODUCTION

Attribute based encryption (ABE) supports both confidentiality and fine-grained access control with a single encryption for data sharing in cloud environment. Profile matching process based on ABE is one of the typical applications of cloud environments such as health care [1], social networks [2], etc. Profile matching is a technique to find people with similar characteristics. In spite of the advantages, it has some severe problems like risk of privacy leakage. ABE scheme is the solution many researchers have put forwarded for these problems. The ABE scheme is categorized into two: Single-Authority ABE (SA-ABE) [3] and Multi-Authority ABE (MA-ABE) [4], [5], [6], [7].

In MA-ABE scheme, there are two sub-categories: with a Central Authority (CA) and without a Central Authority (CA). The first Single-Authority ABE (SA-ABE) scheme [3] was proposed by Sahai and Water. The major drawbacks of single trusted authority scheme are single point failure, key manage-

manent and performance bottleneck problem. To overcome the above mentioned problems, Luo et al. [4] proposed a Hierarchical Multi-Authority Ciphertext Policy ABE (HM-CP-ABE) for key management based on Shamir secret sharing scheme to avoid key leakage because of multiple user's key sharing in addition to ruling out of single point failure and performance bottleneck problem. They use symmetric pairing for HM-CP-ABE construction. Yang et al. [7] proposed a scheme to improve privacy and security in decentralizing multi-authority attribute-based encryption in cloud computing. Most often existing ABE schemes are constructed from bilinear pairings which makes an algorithm inefficient due to the high computational complexity of bilinear pairing. Therefore, the main focus of this paper is reducing the computation cost of bilinear pairing operations to improve the efficiency of the HM-CP-ABE scheme.

A. Our Contributions

The contributions of this paper are highlighted as follows:
1. We propose a more efficient HM-CP-ABE which is based on an asymmetric pairing construction that eliminates the key leakage issue, single point failure and performance bottleneck problem.
2. We apply fast Ate pairing [8] in this asymmetric pairing based HM-CP-ABE construction to reduce its computational costs for initialization, key generation and encryption and then compared with HM-CP-ABE [4], [5], [6], [7].

B. Paper Organization

The rest of this paper is organized as follows. Section II deals with the preliminaries. Section III explains with fast Ate pairing. Section IV explains the system model. Section V presents the proposed HM-CP-ABE scheme based on asymmetric pairing using Section III. Section VI presents with the experimental results and discussions. Section VII concludes the paper.

II. PRELIMINARIES

A. Shamir's secret sharing scheme

Shamir's secret sharing scheme, for \( t \)-out-of-\( n \) secret sharing [9] consists of the following steps.
First, choose a large prime \( p \), and let field \( \mathbb{F} = \mathbb{Z}/p\mathbb{Z} \).
To share secret \( x \) as
\[
x \implies (x_1, x_2, \ldots, x_n)
\]
do the following:
1. Choose coefficients \( f_1, \ldots, f_{t-1} \in \mathbb{Z}/p\mathbb{Z} \), which are to be the coefficients of degree \( t-1 \) polynomial \( f \).
2. Let \( f(z) = f_0 + f_1 \cdot z + \cdots + f_{t-1} \cdot z^{t-1} \), where \( f_0 = x \).
3. Give \( f(i) \) to party \( i \), \( i = 1, 2, \ldots, n \).

Second, when \( t \) parties have a secret, then we have \( t \) points on the curve \( \leq (t-1) \) degree polynomial, so by the fact above, we get unique coefficients to a degree \( \leq (t-1) \) polynomial. The secret is coefficient \( f_0 \).

Formalizing, we use Lagrange Interpolation over a finite field. Given \((i, x_i)\) for \( i \in \mathcal{G}, \mathcal{G} \subseteq \{1, 2, \ldots, n\} \),
\[
f(z) = \sum_{i \in \mathcal{G}} x_i \prod_{j \in \mathcal{G}, j \neq i} \frac{z-j}{i-j} \tag{1}
\]
This is a linear system in \( t \) unknowns, the coefficients \( f_i \), with \( t \) equations. The existence of a unique solution is guaranteed by the fact stated above. So Gaussian elimination can be used to solve. Then,
\[
x = f_0 = f(0) = \sum_{i \in \mathcal{G}} x_i \prod_{j \in \mathcal{G}, j \neq i} \frac{-j}{i-j} \tag{2}
\]
Letting \( \prod_{j \in \mathcal{G}, j \neq i} \frac{-j}{i-j} = c_i \), we have that \( x = \sum_{i \in \mathcal{G}} c_i x_i \). Note that \( c_i \) is a constant independent of \( x_i \)'s. Thus we can compute \( c_i, i \in \mathcal{G} \) ahead of time without knowing the \( x_i \)'s, and then in linear time can find \( x \), the secret, once we have the \( x_i \)'s. So recoverability is quite efficient in this case.

Based on the Shamir's secret sharing properties we can ensure that recovering of \( n \) shares to get the secret, no \( n-1 \) shares provide any information about the secret. Therefore schemes using Shamir’s secret sharing properties have computational security.

B. Decisional Bilinear Diffie-Hellman (DBDH) assumption

Let \( a, b, c, z \in \mathbb{Z}_p \) be chosen at random, \( G \) be the group of prime order \( q \) and \( g \) is the generator of the group \( G \). DBDH problem [10] is a problem that no polynomial time adversary is able to distinguish the tuple \((g^a, g^b, g^c, e(g, g)^{abc})\) from the tuple \((g^a, g^b, g^c, e(g, g)^2)\) with a non-negligible advantage.
This can be formalized as follows:
\[
|Pr[A(g, g^a, g^b, g^c, e(g, g)^{abc}) = 0] - Pr[A(g, g^a, g^b, g^c, e(g, g)^2) = 0]| \geq \varepsilon
\]

III. FAST ATE PAIRING

Nogami et al., [8] proposed a faster Ate pairing \( e \) of embedding degree 12 as follows:
\[
e: G_2 \times G_1 \rightarrow \mathbb{F}_{q^{12}}^\times / (\mathbb{F}_{q^{12}}^\times)^T \tag{3}
\]
where \( G_1 = E[r] \cap \text{Ker}(\emptyset_q - \{1\}) \) and \( G_2 = E[r] \cap \text{Ker}(\emptyset_q - [q]) \) are the two source groups and \( r \) satisfies \( r \{q^{12} - 1\} \) and \( r \{q^{t-1} - 1\}, i < 12 \). \( E[r] \) be the set of \( r \) torsion points of \( E \). Let \([1] \) and \([q] \) denote the \( 1 \) and \( q \) times of scalar multiplications for a rational point respectively. The Frobenius endomorphism \( \Theta_q \) is defined as \( \Theta_q: E \rightarrow E, (x, y) \mapsto (x^{q}, y^{q}) \). Let \( G_1 \subset E(F_q) \) and \( G_2 = E(F_{q^{12}}) \) be the two source groups, \( s = t - 1 \), where \( t \) denotes the Frobenius trace of \( E(F_q) \). It satisfies the bilinearity and Non-degeneracy properties. The Ate pairing \( e \) is defined as follows:
\[
e(Q, P) = f_{s}(P) / (q^{12-1})^{r}, \forall P \in G_1, \forall Q \in G_2 \tag{4}
\]

A. Sextic Twist

The sextic twist is based on Barreto-Naehrig (BN) curve which is a class of non-supersingular pairing friendly curves. The BN curve is defined as \( E/\mathbb{F}_p \) with embedding degree 12. The characteristic and Frobenius trace are given by \( p(\chi) = 36\chi^4 - 36\chi^3 + 24\chi^2 - 6\chi + 1 \) and \( t(\chi) = 6\chi^2 + 1 \) respectively. Let \( \chi \) be an integer. The sextic twisted curve over \( F_{p^2} \) is given by \( E/\mathbb{F}_p: y^2 = x^3 + b, b \in E(F_p) \). The isomorphism \( \psi \) is defined as
\[
\psi: E'(F_{p^2})[r] \rightarrow G_2 \subset E(F_{q^{12}})[r], \quad (x, y) \mapsto (x\nu^{1/3}, y\nu^{1/2}). \tag{5}
\]

Nogami et al., [8] used an isomorphic substitution from \( G_2 \subset E(F_{q^{12}}) \) into \( G_2' \) in subfield-twisted elliptic curve \( E'(F_{p^2}) \). It speeds up scalar multiplications over \( G_2 \) and eliminates denominator calculations in Miller’s algorithm. Fast Ate pairing proposed by Nogami et al., [8] improves Miller’s algorithm about 30% than BKLS algorithm [11] using proper subfield arithmetic operations. Nogami et al., [8] computes the total cost of Ate pairing in 5.16 ms and 13.3 ms
IV. SYSTEM MODEL

The system model of profile matching process is shown in Fig. 1. The system model consists of Cloud Server (CS), Trusted Authority (TA), Attribute Authority (AA), data owner (Alice) and data consumer (Bob). This scheme consists of one fully trusted TA and N AAs honest-but-curious. The CS stores the encrypted data about user’s information which includes name, gender, photo, video, contact, hobbies and so on. TA is authoritative for the initialization phase and the key generation phase, access policy distribution for each AA, approval and administration of subordinate AA. Every AA is authoritative for the administration of part attribute set and subordinate AA and authoritative for the key generation and distribution. The system model is used as hierarchical structure of multi-authority [4] scheme. The TA is the highest hierarchy and other AAs are lower hierarchy. Data consumer can decrypt the data or message only if his/her attributes satisfies the access policy.

V. PROPOSED HM-CP-ABE

Fast Ate pairing algorithm [8] is applied in the proposed asymmetric pairing based HM-CP-ABE constructions. Let bilinear map e: \( G_1 \times G_2 \rightarrow G_T \), \( G_1 \) and \( G_2 \) be the bilinear groups of prime order \( p \). Let \( g_1 \) and \( g_2 \) be the generators of \( G_1 \) and \( G_2 \) respectively. The TA will allot a unique Global Identifier (GID) and Assign an Identifier (AID) for all users and each AA respectively. Usually GID is a digital signature string and verified by all AAs and cannot be obtained illegally.

Initialization: This algorithm is executed by the TA and randomly chooses the parameter \( \alpha \) and \( (\beta_1, \beta_2, ..., \beta_{\text{dep}}) \in \mathbb{Z}_p \), where \( \text{dep} \) is number of recursive times in key structure. It produces public key (PK) and master secret key (MSK) as the output which is calculated in (6) and (7) as follows:

\[
P_{K_0} = \left\{ g_1, g_2, h_1 = g_1^{g_2^0}, h_2 = g_2^{g_1^0}, h_3 = g_1^{g_2^0} \cdot h_4 = g_2^{g_1^0}, f_1 = g_1^g, f_2 = g_2^g, f_3 = g_2^g_1 \right\}
\]

\[
\text{MK}_0 = \{ \beta_1, \beta_2, \beta_{\text{dep}} \}
\]

After Initialization, suppose AA manages the attribute set \( \Lambda = \{A_0, A_1, A_2, ..., A_n \} \), where \( A_0 \) is the first layer attribute, \( A_i \) is the second layer attribute (i.e., \( \text{dep} = 2 \)), \( a_{i,j} \) is the \( j^{th} \) attribute in the attribute set \( A_i \), \( m \) is the number of attributes in the attribute set \( A_i \). The master key of the level 0of AA is calculated in (8) as follows:

\[
\text{MK}_1 = \left\{ \Lambda, D = g_2^{\alpha+r \over \beta_1}, D_{i,j} = g_1^{r_{i,j} \cdot H(s_{i,j})^{T_{i,j}}}, \right. \\
\left. D_{i,j}^{r_{i,j} \cdot E_i} = g_2^{2^{-r_{i,j} \cdot E_i}} \right\}
\]

where \( r \in \mathbb{Z}_p \) is used to represent \( \Lambda \) and \( r_{i,j} \in \mathbb{Z}_p \) is used to represent \( a_{i,j} \in A_i \) (0 ≤ \( i \leq n \), 1 ≤ \( j \leq m \)) and \( E_i \) is used to convert node and decrypt cross set matching for attributes. The sub AA is authorized by level 1 AA. The master key of \( AA_k + 1 \) is calculated in (9) as follows:

\[
\text{MK}_2 = \left\{ \Lambda, D = D_{i,j}^{r_{i,j} \cdot H(s_{i,j})^{T_{i,j}}}, \right. \\
\left. \text{MK}_2 = D_{i,j}^{r_{i,j} \cdot E_i} = E_i \right\}
\]

where the attribute set of Sub AA is \( \Lambda_{k} \in \mathbb{Z}_p \) is used to represent \( \Lambda \) and \( r_{i,j} \in \mathbb{Z}_p \) is used to represent \( a_{i,j} \in A_i \) (0 ≤ \( i \leq n \), 1 ≤ \( j \leq m \)) and \( E_i \) is used to convert node and decrypt cross set matching for attributes.

Key Generation: The \( AA_k \) of Alice attribute set is represented as

\[
\text{Alice}_{\text{AA}} = \{ A_{\text{Alice}_0}, A_{\text{Alice}_1}, ..., A_{\text{Alice}_n} \}
\]

where \( A_{\text{Alice}_0} \) is the set of individual attributes and \( A_{\text{Alice}_1} \) to \( A_{\text{Alice}_n} \) is the subset of attributes with depth 2.

\[
\text{Alice}_{\text{Alice}_i} = \{ a_{i,0}, a_{i,1}, ..., a_{i,m} \}
\]

The two local master keys of \( AA_k \) are \( \beta_{k,1} = \beta_{k,2} = 0 \). The Alice master key is computed as follows:

\[
\text{SK}_{\text{Alice}} = \left\{ \text{SK}_{\text{Alice}}^{\text{Alice}}, D_{\text{Alice}} = g_2^{a_{\text{Alice}} \cdot \text{MK}_2^{(\alpha+\sum a_{i,j} \cdot T_{i,j})}}, \right. \\
\left. \text{MK}_2^{(\alpha+\sum a_{i,j} \cdot T_{i,j})} \right\}
\]

where \( D_{\text{Alice}} \) is used to meet the Alice access structure to decrypt the file. \( D_{\text{Alice}} \) is issued by the TA.
Encryption: $AA_k$ generates the access tree $T^{(k)}$ which is based on the characteristics of Alice attributes to $AA_k$. From the root $R$ in $AA_k$, every node $x^{(k)}$ in $T^{(k)}$ has a corresponding polynomial $q_x$. The order of $q_x$ is calculated as follows:

Order of $q_x = \begin{cases} k_x - 1, & x^{(k)} \text{ is a non-leaf node} \\ 0, & x^{(k)} \text{ is a leaf node} \end{cases}

For every node $x^{(k)}$,

$x^{(k)} = \begin{cases} q_x(0) = \theta, \theta \in \mathbb{Z}_p, & x^{(k)} \text{ is a root node} \\ q_{\text{parent}(x)}(\text{index}(x)), & \text{otherwise} \end{cases}$

where $\text{parent}(x)$ be the parent of node $x^{(k)}$ and $\text{index}(x)$ returns the unique value associated with node $x^{(k)}$.

The $AA_k$ ciphertext is calculated as follows:

$CT_{\text{Alice}}^{(k)} = \begin{cases} T^{(k)}, C^{(k)} = \{h_1^{(k)}, h_2^{(k)}\}, & \forall y^{(k)}(\text{leaf node}) \in Y^{(k)}; C_y^{(k)} = g_2^{q_y(0)}, \\ C_x^{(k)} = H(\text{att}(y^{(k)}))^{q_y(0)}, & \forall x^{(k)}(\text{non-leaf node}) \in X^{(k)}; C_x^{(k)} = h_2^{q_x(0)} \end{cases}$ (13)

Alice sends the ciphertext $CT$ to the server.

Decryption: A user Bob decrypts the $AA_k$ ciphertext of $CT_{\text{Alice}}^{(k)}$ which is based on $T_{\text{Alice}}^{(k)}$ with the help of Bob secret key $SK_{\text{Bob}}^{(k)}$ which is based on Bob attributes set, if Bob satisfies the access structure of the ciphertext $CT_{\text{Alice}}^{(k)}$.

If $x \in Y^{(k)}$ is a leaf node, then

$\text{DecryptNode}(CT_{\text{Alice}}, SK_{\text{Bob}}^{(k)}, x, i)$ should be computed as follows:

$\text{DecryptNode}(CT^{(k)}, SK^{(k)}, x, i) = e(D_{i,j}^{(k)}, C_{x}^{(k)}) / e(D_{i,j}^{(k)}, C_{x}^{(k)}) = e(g_1, g_2)^{\Delta i^{(k)} \cdot q_x(0)}$ (15)

If $x \notin Y^{(k)}$ is a non-leaf node, then

$\text{DecryptNode}(CT_{\text{Alice}}, SK_{\text{Bob}}^{(k)}, x, i)$ should be computed as follows:

We can calculate $F_x$ of $x$ by using Lagrange interpolation method, $F_x = \prod F_z^{\Delta z} S_z^2$, where $iz = \text{index}(z), S_z = \{(\text{index}(z); z \in B_z)\}$ and Lagrange coefficient is $\Delta z S_z = \prod_{z \in B_z} g_x / g_x^z$.

Through upward recursion, we can obtain the value of

$\text{DecryptNode}(CT_{\text{Alice}}^{(k)}, SK_{\text{Bob}}^{(k)}, x, i)$ of root $R$ as follows:

$F_R^{(k)} = \begin{cases} e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)} = e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)}, & i \neq 0 \\ e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)} = e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)}, & i = 0 \end{cases}$

where $i \neq 0$, the transformation of $F_R^{(k)}$ is:

$F_R^{(k)} = e(C^{(k)}, C^{(k)}) / F_R^{(k)} = e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)}$ (16)

If the user Bob satisfies the access policy of $K AA$, which means $(F_R^{(k)})_{k=1}^K$ has no null values, then the computation is as follows:

$Q = \prod_{k=1}^K e(C^{(k)}, C^{(k)}) / F_R^{(k)} = e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)}$ (17)

$e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)} = e(H(\text{att}(y^{(k)}))^{q_y(0)}, q_y(0))$ (18)

The plaintext after decryption is as follows:

$M = e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)} / e(g_1, g_2)^{\Delta_i^{(k)} \cdot q_x(0)} = M$ (19)

A. Proposed Scheme: Proof of Security

A proof of security of the proposed scheme has been recently reported in [10] and is here repeated for the sake of clarity.

Theorem 1: If an adversary can break our construction with non-negligible advantage in the security model under Decisional Bilinear Diffie-Hellman (DBDH) assumption, then a simulator can be constructed to solve the DBDH problem.

Proof. The security game is based on the hardness of the DBDH assumption. Suppose attacker $atk$ can win the FH-CP-ABE game with advantage $\epsilon$. We construct a simulator sim that can distinguish a DBDH tuple from a random tuple with advantage $\epsilon$. Let $G_1$ and $G_2$ are a source group and $G_T$ is a target group. Let $g_1$ and $g_2$ are generators of $G_1$ and $G_2$ respectively. The challenger chooses the fair binary coin $h \in \{0,1\}$, $g_1 \in G_1, g_2 \in G_2, R \in G_T$ and $a, b, c \in \mathbb{Z}_p$. If $h = 0$, then the challenger defines $T$ to be $e(g_1, g_2)^{abc}$. Otherwise, he sets $T = e(g_1, g_2)^{abc}$ or $R$. The challenger then gives the simulator the DBDH details and then simulator sim now plays the role of challenger in the security game.

Init: During the init phase, sim receives the challenge access structure $T^*$ from attacker atk.

Setup: To provide a public key $PK$ to $atk$, sim randomly chooses $\alpha' \in \mathbb{Z}_p$ and note $\alpha = \alpha' + ab$. It calculates
e(g_1, g_2)^a like this: e(g_1, g_2)^a = e(g_1, g_2)^a e(g_1, g_2)^ab. Finally, sim sends public key PK to atk.

Phase 1: During this phase, atk submits an attribute set \(\mathcal{W}_f \in T\) (\(\mathcal{W}_f \in T^*\)) to sim. Simulator sim chooses \(ru^{(k)}_i\), \(ru^{(k)}_u\) \(\in\) \(\mathbb{Z}_p\) and set \(ru^{(k)}_i = ru^{(k)}_i - a\). It can obtain \(D^{(k)} = g_2^{a \cdot ru^{(k)}_i}\). For each attribute in \(\mathcal{W}_f\), sim wants to choose \(\tau_j \in \mathbb{Z}_p\). It computes the rest of the secret key of \(k^{th}\) AA as follows: \(D_i^{(k)} = g_2^{ru^{(k)}_{ij}} h(\alpha^{(k)}_{i,j})^{ru^{(k)}_{ij}}\). Finally, sim sends the secret key to atk.

Challenge: The attacker atk submits two equal length messages \(m_1\) and \(m_2\) along with a challenge access structure \(\mathcal{A}^c\). Sim randomly generates a bit \(\hat{h} \in \{0, 1\}\) and computes \(CT = e(g_1, g_2)^{a_{\alpha}^{(k)} \cdot ru^{(k)}_i}\). Hence, \(e(g_1, g_2)^{a_{\alpha}^{(k)} \cdot ru^{(k)}_i}\). Finally, sim sends the CT to atk.

Phase 2: Same as the Phase 1.

Guess: The attacker atk outputs a guess \(\hat{h}'\) of \(\hat{h}\). If \(\hat{h} = \hat{h}'\), simulator sim guess that \(T = e(g_1, g_2)^{a_{\alpha}^{(k)} \cdot ru^{(k)}_i}\). Otherwise, \(T\) is a random target group element in \(G_T\).

The advantage of the attacker is \(\varepsilon\), when \(T = e(g_1, g_2)^{a_{\alpha}^{(k)} \cdot ru^{(k)}_i}\). The advantage of the attacker is \(\varepsilon/2\), when \(T\) is a random target group element in \(G_T\). Finally, the advantage of the simulator in this security game is \(\varepsilon/2\).

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we show the computation cost of initialization phase, key generation phase and encryption for the proposed HM-CP-ABE scheme in cloud environment. We implemented the proposed HM-CP-ABE scheme based on the ELiPS [8] pairing library and GMP6.0.0 library and using the GCC4.8.2 compiler. The implementation uses a Barreto-Naehrig (BN) curve which is class of a non-supersingular pairing friendly curves \(y^2 = x^3 + b, b \in E(F_p)\) with embedding degree 12. The experimental results are shown in Fig. 2, Fig. 3 and Fig. 4. The number of attributes used in the simulation are \(\{20, 40, 60, 80, 100\}\).

As illustrated in Fig. 2, the proposed scheme reduces the computational cost for initialization phase and we can also show that the results are slightly increasing and approximately following a linear relationship with the number of attributes. The initialization phase is associated with the hierarchical structure of authority and number of attributes handled by AA. The schemes [5], [6] adopt the accountability mechanism that utilizes bilinear mappings multiple times. As we have shown in Fig. 2, when the number of attributes are varied, the initialization time of proposed work is less than [4], [5], [6] and [7].

As in [4], the proposed HM-CP-ABE scheme reduces the workload and communication cost since all pairing operations are computed in server and data owner update their own attributes in AA only, rather than in the entire attribute sets. The results show that the proposed HM-CP-ABE scheme enhances the efficiency and provide fine grained access control in cloud environment.
VII. CONCLUSION

In this paper, we proposed a more efficient asymmetric pairing based Hierarchical Multi-Authority CP-ABE (HM-CP-ABE) construction for profile matching. We utilized fast Ate pairing [5] which was based on a Barreto-Naehrig (BN) curve with embedding degree 12. It also speeds up scalar multiplications and eliminates denominator calculations in Miller’s algorithm. The proposed HM-CP-ABE scheme is then compared with existing schemes [4], [5], [6], [7]. The experimental results show a mitigation in terms of computational costs for initialization, key generation and encryption.

ACKNOWLEDGMENT

This work was supported by Ministry of Human Resource Development (MHRD) under the Government of India.

REFERENCES


Balaji Chandrasekaran received the B.E degree in Computer Science and Engineering from JJ College of Engineering and Technology, Anna University, Chennai in 2009 and the M.E degree in Software Engineering from Anna University, Tiruchirappalli in 2011. Currently, he is a PhD researcher in the Department of Computer Applications at National Institute of Technology, Tiruchirappalli, India. His research interests include: Cryptography, Cloud Computing, and Information Security. He is a student member of ACM.

Yasuyuki Nogami graduated from Shinshu University in 1994 and received the PhD degree in 1999 from Shinshu University. He is now an associate professor of Okayama University. His main fields of research are finite field theory and its applications such as recent public key cryptographic. He is now studying about elliptic curve cryptography, pairing-based cryptography, Lattice-based cryptography, pseudo random number generator, Advanced Encryption Standard, and homomorphic encryptions. Recently, he is a member of security research group in Okayama University and particularly focusing on IoT security from the viewpoints of software and hardware implementations. He is a member of IEICE and IEEE.

Ramadosh Balakrishnan received the M.Tech degree in Computer science and Engineering in 1995 from the Indian Institute of Technology, Delhi and the PhD degree in Applied Mathematics in 1983 from Indian Institute of Technology, Bombay. Currently he is working as a Professor of Computer Applications at National Institute of Technology, Tiruchirappalli. His research interests include: Software Testing Methodologies, Security and Privacy in Big Data and Cloud, software Metrics, Data Warehouse – EAI, Data Mining, WBL, and XML. He is a recipient of Best Teacher Award at National Institute of Technology, Tiruchirappalli, India during 2006-2007. He is a member of IEEE, Life Member (LM) of ISTE, New Delhi, Life Member (LM), Computer Society of India.