Multi-user Attribute Based Searchable Encryption

Kulvaibhav Kaushik, Vijayaraghavan Varadharajan, Rajarathnam Nallusamy
Convergence Lab, Infosys Labs
Infosys Limited, Bangalore, India 560100
Email: Kulvaibhav_kaushik@infosys.com, Vijayaraghavan_V01@infosys.com, Rajarathnam_N@infosys.com

***Abstract*—Globalized services economy is increasingly adopting cloud computing in which applications, platforms, and infrastructre are available as services. Huge volume of data including sensitive data is outsourced and stored remotely in public clouds. There is a need for efficient protection of the privacy and security of users and their data without affecting its utility to authorized users. There needs to be a trade-off between data security and availability. Multi user encryption schemes provide partial solution by encrypting data and managing keys efficiently among the users but exposing the entire data accessible to all the users. But, in certain applications, the data need to be accessed by only specified users and they shall have access to only specifed parts of the data on a need to know basis. Ciphertext policy attribute based encryption supports keyword based search but require large amount of data for a single keyword thereby increasing the size and hence the cost of storage on the cloud. In this paper, we have proposed an encryption scheme which requires less storage space on the cloud but provides fine grained access control to authorized users. This is a hybrid scheme that combines the strenghts of both attribute based encryption and searchable encryption.**

I. Introduction

With the applications, platforms, and infrastructre available as services over the cloud which has no clear geographic boundary, there is an exponential growth of digital data. In the world of Internet of Things, in addition to the electronic data generated and consumed by humans, data are being generated and consumed by machines, devices, appliances, and son on. Storage as a service by cloud providers provides a low cost solution for data storage and access. Security and privacy of data in the cloud have emerged as an area of major concern, especially when the data is sensitive, such as financial and healthcare data.

The key requirements of the system are:

1) Secure storage of the data on the remote cloud server (non-trusted).
2) Allow authorized user to access and search over the data.

An architecture where the data encryption and decryption is performed at the user side only and the server doesnt have any access to the plaintext is the requirement. The proposed solution provides secure and searchable data storage at remote service provider using Attribute Based Encryption (ABE), which provides limited access of the data on need basis only to authorized users. The searchable encryption provides search over entire data and not keyword based search, without adding any overhead. The architecture has a data server and a proxy server. It provides a data search mechanism where high overhead computation is performed at the server and limited computation is performed at the client. It also provides solution for key revocation and certificate stealing. In this paper, we propose an architecture for search over the encrypted data stored at the server and providing fine grained access control that allows different users to access data as per the security policy. This eliminates the need for keyword based search and unauthorized data access by the storage server.

II. Related Work

The searchable encryption has emerged as an area of advanced research with Boneh et als method [1] for trapdoor generation based on hash functions pioneering in the field. Cryptographic cloud storage [2] proposed a three tier architectural model for data encryption and search. The private query on encrypted data [3] and no shared keys [4] approaches with an intermediary proxy server provide data encryption and search on encrypted data for multiple users but make entire data accessible to all the users. The Identity Based Encryption (IBE) and Hierarchical IBE [5] provide data storage with controlled access but require sharing of common secret key among the set of users in the hierarchy. Fine grained access control provides different access rights to different set of users and thus provides flexibility in access rights specification to different set of users. Software automated access control [6], [7] checks user legitimacy but stores data as plaintext and is useful only for secure servers. The attribute based encryption provides user controlled access to data. Threshold based ABE by Sahai and Waters [8], proposed ciphertext labeling with attributes and association of users private key with attribute set and threshold parameter. To decrypt the data, minimum number of matching required is equivalent to threshold parameter. Goyal et al. [9] proposed key policy ABE that associates attribute set with the ciphertext and the decryption key with a monotonic access tree over the attribute values. But the association is reverse in ciphertext policy ABE [10]. The ABE methods provide keyword based data search and require high overhead in data storage. In this paper, we

978-0-7695-4973-6/13 $26.00 © 2013 IEEE
DOI 10.1109/MDM.2013.94

propose fine grained access control architecture for search over the encrypted data with less overhead.

III. Preliminaries

The proposed scheme facilitates search over the entire encrypted data without any storage overhead.Bilinear pairing [11] and discrete log problem are non-polynomial hard problems and are used for data encryption and key generation in the proposed architecture.

Consider two cyclic groups of prime order q, $(G_1, +)$ and $(G_2, \cdot)$. The map between the two is given by $\hat{e}: G_1 \times G_1 \to G_2$ with the following properties:

1) Bilinear: $\forall P, Q \in G_1, \forall a, b \in Z_q$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
2) Non Degenerate: For $\forall P \in generator(G_1)$ $\hat{e}(P, P) \in generator(G_2)$
3) Computable: There exist algorithm to calculate $\hat{e}(P, Q)$ given any $P, Q \in G_1$. The Weil pairing and Tate pairing are among the few methods available for calculating elliptical pairing

A map satisfying the above properties forms an admissible bilinear pairing.

The Discrete Logarithm Problem (DLP) over finite cyclic group is defined below [12]. Let g and h be elements of a finite cyclic group. The solution x to the equation $g^x = h$, is the discrete logarithm of h to the base g. Consider an elliptical curve E defined over finite field F_q of prime order q. For any point P on the curve i.e. $P \in E(F_q)/\{O\}$ where O is the point at infinity, compute $Q = d \cdot P$. Elliptic Curve Discrete Logarithm Problem (ECDLP) states that given P and Q there exists no polynomial time algorithm to find d with a non-negligible probability.

IV. Proposed Architecture

Data storage on cloud involves different participating entities as shown in Figure 1.

1) Data owner/user: The individual or corporation accessing the data. They may be the entity which saves the data or searching for it. They possess attributes corresponding to their role and user key.
2) Data Server: Performs data storage, search and retrieval operations.
3) Proxy Server: An intermediary between the data user/ owner and the Data Server thorough which all communication between the two occurs.
4) Certificate generator/ key management authority: Provides the user and proxy with the certificates, primary key and secondary key.

V. Our Scheme

In the proposed solution the private keys or certificates are associated with a set of attributes S describing the data user. The data owner while saving the data at the remote server defines a policy through a monotonic access structure and associates it with the data. The attributes possessed by the data user must satisfy the associated access policy to access the corresponding data. The access tree is a composition of threshold gates AND and OR and the attributes represent the leaf nodes. For successful decryption of the ciphertext, satisfaction of policy or access tree with the private key of the user is required.

We define the access structure by tree T [9], [10]. Some terminologies associated with the access tree are:

- Let there exist w attributes in total marked as $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_w$
- The value of attributes be represented by : $\{v_i\}_{1 \leq i \leq w}$
- The access tree T is defined with node r as the root node. Any subtree defined over the tree T and rooted at node x is described as T_x
- Number of children of any node x is num_x
- Threshold value of node x is $k_x = \{\frac{1, ORgate/LeafNode}{num_x, ANDgate}$
- Parent of any node x on the tree is defined as parent(x)
- The leaf node corresponding to any attribute α_i in the tree is given by = $node(\alpha_i)$
- The attribute associated with a leaf node x is given by function att(x) and is defined only for leaf nodes
- The access tree T defines ordering of nodes. Children of any node x are numbered from 1 to num_x. The number associated with node x is returned by function index(x)
- All the children of a particular node are described by a set $S_x = [1, 2, \ldots num_x]$
- i^{th} child of any node x is given by function $child_i(x)$

Access tree satisfaction: Let γ represent a set of attributes and T_x denote the access tree such that γ satisfy T_x , then the condition of satisfaction is defined as $T_x(\gamma) = 1$. For any non-leaf node compute $T_x(\gamma)$ value recursively. Let x be a non-leaf node with threshold k_x. $T_x(\gamma)$ returns 1 if and only if k_x children of x return 1 i.e. $T_{(x')}(\gamma) = 1$, where $x^{'}$ represent children of node x. For any leaf node x, $T_x(\gamma) = 1$ if and only if $att(x) \in \gamma$.

Consider a bilinear group G_1 of prime order p, and let g be the generator of $G_1, g \in G_1$. Let $\hat{e} : G_1 \times G_1 \to G_2$ and $\hat{e^{'}} : G_2 \times G_2 \to G_3$ be bilinear pairings. Security parameter, κ, determines the size of the groups. We define the Lagrange coefficient as $\Delta_{i,s}$ for $i \in Z_p$ and a set S, of elements in $Z_p : \Delta_{i,s} = \prod_{j \in S, j \neq i} \frac{j}{j-i}$. We also define a hash function $H : \{0,1\}^* \to G_1$ that is modeled as a random oracle. The construction is as follows.

A. Setup(κ)

The setup algorithm runs at key management server. It selects a bilinear group G_1 of prime order p with generator g. It chooses random elements $\alpha, \beta, k \in Z_p$. It publishes

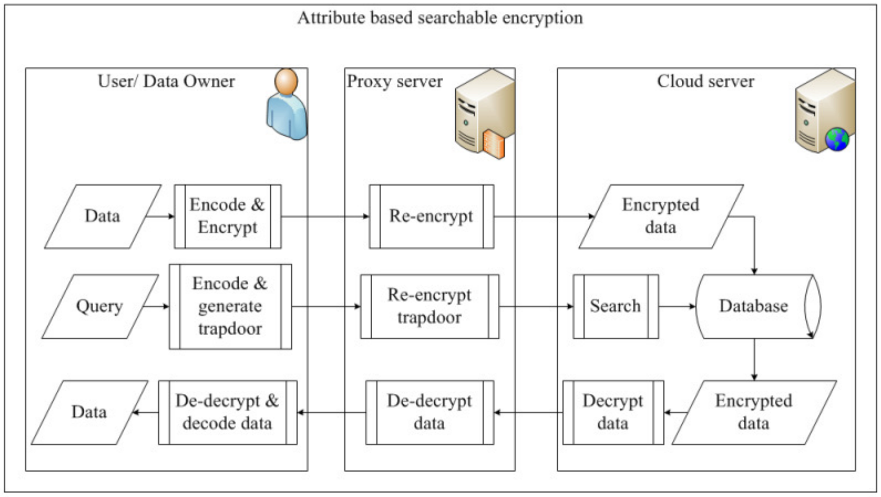

Figure 1. Architecture diagram

Public Key PK and the Master Key MK as:
Public parameters $PK = \{t, g^{\frac{1}{\beta}}, h = g^{\beta}, \hat{e}(g,g)^{\alpha}\}$, where $t \in R$
Private Parameters, possessed by Key/Certificate generator:
$MK = \{\beta, g^{\alpha}\}$
Also it generates primary and secondary key as:
$K_{pri_i} \in Z_p$
$K_{sec_i} = k - K_{pri_i}$
The key generator provides the user i with K_{pri_i}
and to the proxy server K_{sec_i}.

B. *Encode* (m, E_p, abp)

Encoding is used to map the data to a point on an elliptical curve [13]. An encoding method is described in the following steps.

- Select an elliptic curve$E_p(a, b)$
- Let the curve E_p has N points over it
- Let the sample input set be denoted by $In = \{0, 1, 2 \cdots, 9, A, B, \cdots Y, Z\}$ and hence the corresponding numeric conversion as $In_{num} = \{0, 1, 2 \cdots, 9, 10, 11, \cdots 34, 35\}$. Similarly for converting words or a sequence of characters entire dictionary of sample data is mapped to a defined set of numbers
- This converts the message into a series of numbers (Between 0 and 35 in the example considered)
- We select an Auxiliary Base Parameter, (ABP). For example abp = 20. The value of abp is required for decoding also
- For any number inIn_{num} say m, take $x = m \cdot (abp) + 1$
- Using x and chosen elliptic curve equation $E_p(a, b)$, compute y
- If the equation is not solvable for y, try with $x = m \cdot (abp) + 2$ and then $x = m \cdot (abp) + 3$ until a solution is obtained for y
- In practice, a solution is obtained till $x = m \cdot (abp) + (abp - 1)$

The solution point $M = \{x, y\}$ is a point on elliptical curve and thus the data m is converted to an elliptical curve point. Convert the entire message to point on elliptical curve.

C. *Encrypt* $(PK, M, A, K_{pri}, K_{sec})$

The encryption algorithm encrypts the message M under the tree access structure A. The algorithm chooses a polynomial q_x for each node x (including the leaves) in the tree A. Starting from root node r, the polynomials are chosen using a top down approach. For each node x in the tree, the degree d_x of the polynomial q_x is set to be one less than the threshold value k_x for that node, i.e., $d_x = k_x - 1$.

Beginning with the root node r, the proposed algorithm chooses a random $s \in Z_p$ and sets $q_r(0) = s$. To define the polynomial completely, it chooses d_r additional points on the polynomial q_r randomly. For any other node x, it sets $q_x(0) = q_{parent(x)}(index(x))$ and randomly chooses d_x other points to define q_x completely.

Let, Y be the set of leaf nodes in A and i reprersent attribute. Ciphertext CT is constructed by giving the tree access structure A and computing
$CT = (A, C^{''} = M \cdot (\hat{e}(g,g)^{\alpha s}) \cdot g^{K_{pri_i}}, X^{'} = h^{s}, X = \hat{e^{'}}(\hat{e}(g,g)^{\alpha s}, \hat{e}(g,g)^{\alpha s}), \forall y \in Y : C_y = g^{q_y(0)}, C_y^{'} =$

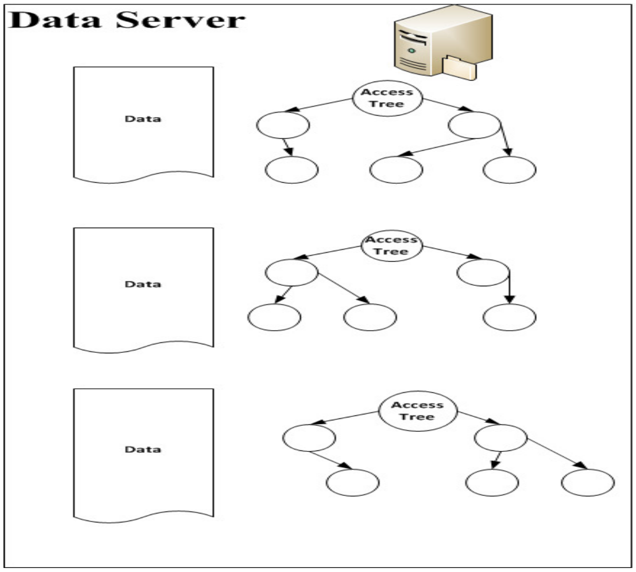

Figure 2. Server data storage scheme

$H(i)^{q_y(0)})$

The Ciphertext CT is sent by user to the proxy server, which re-encrypts $C^{''}$ using the secondary key stored at the proxy server for the particular user. At proxy server,

$C = C^{''} \cdot g^{K_{sec_i}} = M \cdot (\hat{e}(g,g)^{\alpha s}) \cdot g^{K_{pri_i}} \cdot g^{K_{sec_i}} = M \cdot (\hat{e}(g,g)^{\alpha s}) \cdot g^{K_{pri_i}+K_{sec_i}} = M \cdot (\hat{e}(g,g)^{\alpha s}) \cdot g^{k}$

Server receives from proxy $CT = (A, C, X^{'}, X, \forall y \in Y : C_y, C_y^{'})$ and saves it.

The data saved at server consists of two parts: Data and Access Tree as shown in Figure 2. Each data file is associated with an access tree.

D. *KeyGen* (MK, S)

It takes an attribute set S as input and outputs a key that identifies with that set. The proposed algorithm randomly chooses $r \in Z_p$, and then random $r_j \in Z_p$ for each attribute $j \in S$. Then it computes the Secret Key SK as
$SK = (D = g^{\frac{(\alpha+r)}{\beta}}, \forall j \in S : D_j = g^r \cdot H(j)^{r_j}, D_j^{'} = g^{r_j})$

E. *Search* $(PK, K_{pri}, K_{sec}, CT, SK, M)$

The user wants to search for word kw=m at the server. User maps the search word m to the elliptic curve E_p as M. The user prepares the search query as:

$sk_2^{'} = M \cdot g^{K_{pri_i}}$

The proxy uses the corresponding secondary key and converts the query as:

$sk_2 = sk_2^{'} \cdot g^{K_{sec_i}} = M \cdot g^{K_{pri_i}} \cdot g^{K_{sec_i}} = M \cdot g^{K_{pri_i}+K_{sec_i}} = M \cdot g^k$

We define a recursive algorithm TestAttr(CT,SK, x) which accepts as input a ciphertext $CT = (T, C, C, \forall y \in Y : C_y, C_y^{'})$, a node x from T and private key SK linked with an attribute set S. For any leaf node x, define i = att(x). If $i \in S$, then

$TestAttr(CT, SK, x) = \frac{\hat{e}(D_i, C_x)}{\hat{e}(D_i^{'}, C_x^{'})}$
$= \frac{\hat{e}(g^r \cdot H(i)^{r_i}, g^{q_x(0)})}{\hat{e}(g^{r_i}, H(i)^{q_x(0)})}$
$= \hat{e}(g,g)^{r \cdot q_x(0)}$

For $i \notin S$, we define TestAttr(CT,SK,x) = $\perp$. Considering the recursive case where x is a non-leaf node, the algorithm computes TestAttr(CT,SK,z) for all nodes z that are children of x and save the output as F_z. Let S_x denote a random k_x -sized set of child nodes z such that $F_z \neq \perp$. If no such set exists then the node is not satisfied and the function returns $\perp$.Otherwise, compute F_x and return it.

$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S_x^{'}}(0)}$
Where , i=index (z) and $S_x^{'} = \{index(z) : z \in S_x\}$

$= \prod_{z \in S_x} (\hat{e}(g,g)^{r \cdot q_z(0)})^{\Delta_{i,S_x^{'}}(0)}$

$= \prod_{z \in S_x} (\hat{e}(g,g)^{r \cdot q_{parent(z)}(index(z))})^{\Delta_{i,S_x^{'}}(0)}$ (By construction)

$= \prod_{z \in S_x} (\hat{e}(g,g)^{r \cdot q_x(i)})^{\Delta_{i,S_x^{'}}(0)}$

$= \prod_{z \in S_x} \hat{e}(g,g)^{r \cdot q_x(0)}$
(using polynomial interpolation)

After completion of TestAttr, a function is performed over the root node to check if the attributes provided match with the policy. If the tree is satisfied by S we set $TA = TestAttr(CT, SK, R) = \hat{e}(g,g)^{rq_R(0)} = \hat{e}(g,g)^{rs}$. The algorithm now decrypts by computing

$z_1 = \frac{\hat{e}(D, X^{'})}{TA} = \frac{\hat{e}((g)^{\frac{\alpha+r}{\beta}}, g^{\beta s})}{\hat{e}(g,g,)^{rs}} = \hat{e}(g,g,)^{\alpha s}$

Check if $X = \hat{e^{'}}(z_1, z_1)$

If Yes role match,
Check if $C = Z_1 \cdot sk_2$

If Yes then word also match.

F. *Decrypt* (CT, SK)

Server send to Proxy $\frac{C}{z_1} = \frac{M \cdot (\hat{e}(g,g)^{\alpha s}) \cdot g^k}{\hat{e}(g,g)^{\alpha s}} = M \cdot g^k$

Proxy sends $C^{''} = \frac{M\cdot g^k}{g^{k_{sec}}}$

The user receives $C^{''}$ from server. The message M is

$$M = \frac{C^{''}}{g^{k_{pri}}} = \frac{\frac{M\cdot g^k}{g^{k_{sec}}}}{g^{k_{pri}}} = \frac{M\cdot g^k}{g^{k_{sec}}\cdot g^{k_{pri}}} = \frac{M\cdot g^k}{g^{k_{sec}+k_{pri}}} = \frac{M\cdot g^k}{g^k} = M$$

G. Decode (M)

Consider each point $M = (x, y)$ and set m to be the greatest integer less than $\frac{x-1}{abp}$. Then the point (x,y) decodes as the symbol m.

$$m = \lfloor \frac{x-1}{abp} \rfloor$$

VI. Security Analysis

This section discusses the security requirements for data privacy and security on the cloud storage and analyses the security of the proposed model.

- User controlled access control: The attribute based searchable encryption provides for limited and controlled access to the data. The owner defines access policy and associates it with the ciphertext thus limiting access to the data only to authorized users.
- Non-sharing of keys: To provide controlled access to data among different users, key sharing between multiple users is not secure. Key sharing between multiusers also makes user revocation impractical. In the proposed framework each user possesses a unique key and also certificates. For every user different primary and secondary key is generated.
- Query unforgeability: Only the user is able to construct a viable query on the data saved. No adversary may construct a query even if it gets access to user keys since additional validation is performed at the proxy server.
- User revocation: In multi-user encryption, user revocation is the biggest bottleneck. Even ABE schemes render data accessible to the revoked users. The proposed scheme provides for data access only to the current users. The deletion of the secondary key from the lookup table at proxy server render the data no more accessible to the revoked users.
- Direct data search: The proposed scheme is applicable for search over the entire data and not over a limited set of keywords. Thus it removes the overhead involved in keyword based search over ABE.
- Collusion resistant: The scheme is free from collusion of users with different attributes to collude together and get access to data, they are unauthorized to access.

A. Security Model for searchable ABE

Let the adversary A query for any private keys which cannot be used to decrypt the challenge ciphertext. In the proposed scheme access trees are associated with the ciphertexts and the private keys are associated with attributes describing the roles. In the given definition for security the adversary will choose an encryption over an access structure $A*$ to be challenged. It can query for any private key S such that S does not satisfy $S*$. The security game is as follows:

Setup. The Setup algorithm is run by challenger C and it gives the public parameters, PK to the adversary.
Phase 1. The adversary generates repeated private keys corresponding to sets of attributes $S_1, \cdots, S_{q_1}$.
Challenge. The adversary submits two messages M_0 and M_1 of equal length. Also the adversary gives a challenge access structure (access tree) $A*$ such that not even a single Phase 1 set $(S_1, \cdots, S_{q_1})$ satisfies the access structure. Challenger C flips a coin b randomly, and encrypts M_b under $A*$. Ciphertext $CT*$ is given to the adversary.
Phase 2. Phase 1 is repeated with restriction that not even a single attribute set of $S_{q_1+1}, \cdots, S_q$ satisfy the access structure corresponding to the challenge. The adversary guesses the output $b^{'}$ made for b. The advantage of winning of an adversary A in this challenge game is defined as $Adv(A) = \{pr[b^{'} = b] - \frac{1}{2}\}$. The model can be easily extended to eliminate the chance for CCA (Chosen Ciphertext Attacks) by allowing for decryption queries in Phase 1 and Phase 2.
Definition 1 The proposed scheme is secure if all the polynomial time adversaries have an insignificant advantage in the above game.
The above game shows security of the proposed scheme against chosen plaintext attacks. The security of the proposed scheme may be extended to CCA by applying a random oracle technique.

VII. Conclusions

In this paper, we have proposed a scheme to ensure secure storage, search and retrieval of data on cloud using attribute based encryption. The proposed mechanism saves the data at the server with each dataset associated with an access policy. For a file with m words, the ciphertext storage required is equivalent to m items, as compared to the schemes based on trapdoor based search where additional data is stored for search. The scheme provides search over entire data leading to larger search space and thus improved search results, increasing the utility of the data saved. The proposed scheme provides for the major part of the decryption at the server end and the ciphertext is returned to the user as a DLP problem which is secure but at the same time involves less computation. The scheme first makes a check for the policy satisfaction and only on fulfillment, it makes search for the keyword.
The framework provides user controlled access over the data stored at cloud. The proposed scheme is based on the

non-polynomial time hard problems of bilinear pairing and discrete logarithmic problem which makes it more secure. The proposed architecture provides an efficient, secure and fine grained access method over encrypted data stored in a cloud and is easily deployable on the existing cloud architectures.

References

[1] D. Boneh and M. Franklin: Identity-Based Encryption from the Weil Pairing. SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003.

[2] Seny Kamara and Kristin Lauter: Cryptographic Cloud Storage. Proceedings of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization, January 2010

[3] Yang, Y.J., Ding X.H, Deng, R.H. and Bao, F.:Multi-User Private Queries over Encrypted Databases. Int. J. High Performance Computing and Networking, Vol. 1, Nos. 1/2/3, pp.64-74 (2008).

[4] Dong Changyu, Russello Giovanni, and Dulay Naranker.,Shared and searchable encrypted data for untrusted servers,In Lecture Notes in Computer Science. Springer, 2008.

[5] C. Gentry and A. Silverberg: Hierarchical ID-Based Cryptography. In Y. Zheng, editor, Proceedings of Asiacrypt 2002, LNCS, Springer-Verlag, 2002.

[6] Rita Gavriloaie, Wolfgang Nejdl, Daniel Olmedilla, Kent E. Seamons, and Marianne Winslett: No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In ESWS, pages 342-356, 2004.

[7] Jiangtao Li, Ninghui Li, and William H. Winsborough: Automated trust negotiation using cryptographic credentials. In ACM Conference on Computer and Communications Security, pages 46-57, 2005.

[8] A. Sahai and B.Waters: Fuzzy Identity Based Encryption. In Advances in Cryptology Eurocrypt, volume 3494 of LNCS, pages 457473. Springer, 2005.

[9] Vipul Goyal, Omkant Pandey, Amit Sahai, Brent Waters: Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In ACM conference on Computer and Communications Security (A CM CCS), 2006.

[10] John Bethencourt, Amit Sahai, Brent Waters: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy (2007)

[11] Certicom Research, Standards for Efficient Crpytography Group (SECG) – SEC 1: Elliptic Curve Cryptography. Version 1.0, September 20, 2000. See $http://www.secg.org/secg\_docs.htm$.

[12] Chris Studholme: The discrete log problem. Research paper requirement (milestone) of the Ph.D. program at the University of Toronto, June 2002.

[13] Padma Bh, D.Chandravathi , P. Prapoorna Roja: Encoding And Decoding of a Message in the Implementation of Elliptic Curve Cryptography using Koblitzs Method. In (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 05, 2010, 1904-1907.