Verifiable Cryptographic Based Data Transformation System

[1]A.T.Tamilselvi,[2] C.S.Somu
[1]M.E. Student ,[2]Assistant professor
[1]Department of computer Science and Engineering
[1]DMI College of Engineering, Chennai, India.

***Abstract* - A sensitive data is stored and shared in third part sites. A public-key-based one-to-many encryption that allows users to encrypt and decrypt data based on user attributes. ABE is flexible access control of encrypted data stored in the cloud, using access polices and attributes associated with private keys and cipher texts. By using an encryption/Decryption standard our distributed secure computation system shows that our approach seamlessly integrates security enforcement at the user intensity with a certain trust level. This method is verifying the correctness of the conversion from encryption to decryption is made. In the existing system, Private and public key are generated by the source organization and proxy verifies the private key before issuing destination. But the existing system, attribute based master secret key is not unique at their attribute consideration. Hence a system check the unique attribute based encryption using proxies and outsourced decryption.**

***Keywords* - Attribute Based Encryption, Access Structure, Proxy Verification, Outsourced Decryption**

I. INTRODUCTION

The network security is consists of the previsions and policies adapted by network administrator to prevent the unauthorized access. And then network security is a layer of the protection. In network number of hackers is available in third party site so the security is important in data transformation. The encryption and decryption is more expensive in the data transformation between the receiver and sender. A cryptographic is a secure communication in third parties. Then it given storing and transmitting data in particular form whom it is intended can lead and process it. Cryptography is the study of secret writing.

Concerned with developing algorithms: Conceal the context of some message from all other sender and recipient verify the correctness of a message to the recipient. In public key cryptographic is the decrypt the data in receiver keys. The cryptographic key length is measures by bits of the keys the size will be in 128 bits. In attribute based encryption (ABE) is a public-key, and it is a flexible access control. Attribute is a quality of the particular factor. ABE is a new public key based one-to-many encryption. Based on the attributes and access policies only the private key and cipher text is generated. There are two kinds of ABE schemes: key-policy ABE and cipher text-policy ABE. In a CP-Abe scheme, every cipher text is associated with attributes, and private key associated with set of attributes. KP-ABE is the dual to CP-ABE in the sense that an access policy is encoded into the user's secret key. One of the main efficiency drawbacks of the existing ABE schemes is attribute based master secret key is not unique at the attribute consideration.

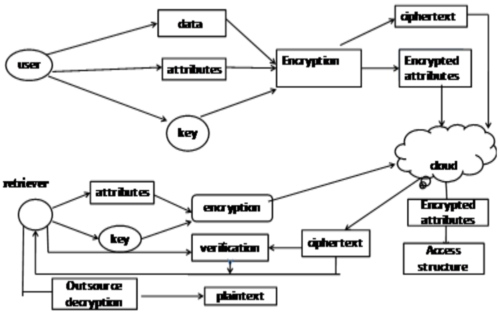

Figure 1: Attribute based encryption

The proxy operated by a cloud service using user's attributes and access policy to get a cipher text details to user. Identity is a set of attributes; it is commonly used in identity based encryption [8]. The fuzzy identity based encryption is measured by set overlap distance metrics. Fuzzy IDE hide the public key then it is used to encrypt the cipher text. The documents are stored in untrusted server relying on trusted server. The monotonic access structure is used ABE it will get a restricted access structure. The secret sharing schemes is limited access to the monotonic access structure [2]. It removes the restricted or fixed size of attributes. Attribute based re-encryption is a proxy transform a cipher text under an access structure and same plain text under an access policy. The chosen cipher text attributes are very difficult to handle in proxy re-encryption [9]. Encrypted data shares only the limited user's cross- grained level, so using the fine-grained access level using different access rights to an individual users [1]. The different user's to allow decrypting the different pieces of data in security policy.

The central authorities and attribute authorities are managed by identity, attribute related keys [7]. CP-ABE scheme managed different attribute authorities and no authority can independently decrypt any cipher text. A maximum hierarchy depth had to be fixed at setup. Nested dual system encryption to prove the security of an HIBE and selectively secure in ABE [3]. Access structures are converted as the minimal set, which can provide fast decryption ability. Higher decryption efficiency in that the decryption cost is independent to the depth of access structures. To refresh the master secret key the leakage information is allowable [10]. In outsource the decryption of ABE cipher text the size of the cipher text a time is complex in access formula. The CPA and CCA is secure in outsource decryption, user's saves the bandwidth and decryption time without increase the number of transformation [5]. Set of attributes and private keys are associated with access structures that specify which cipher texts the key holder will be allowed to decrypt. In cipher text policy challenging problem is to achieve the same level of expressivity and efficiency. The constant size cipher text is to be used in provable secure [4]. Cipher text policy hiding CP-ABE construction but only supporting restricted access structures. Protecting the privacy of plaintexts, cipher text policy hiding CP-ABE also protects the description of the access structures associated with cipher texts [6].

A. Our contribution

The cloud is used to verify the transformation is correct. It does not rely on random oracles. First modify the ABE with outsourced decryption and the new schemes will be implemented in further verifiable of transformation. Then rest of the paper will be using the proxy verification and outsourced decryption.

II. METHODOLOGY TO USE

A. Cipher text policy attribute based encryption:

Cipher text policy attribute-based encryption scheme, private key is associated with a set of attributes representing their capabilities, and a cipher text is encrypted such that only users whose attributes satisfy a certain policy can decrypt. These are the working process,

a. List of users U = fu$_1$; u$_2$; ::::; u$_n$g
b. List of Attributes A = {a$_1$; a$_2$; ::::; a$_k$}
c. Each user will be assigned a subset of attributes
d. D = {d$_1$; d$_2$; ::::; d$_x$} Where D 2 A
e. Each encrypted le will be assigned an access tree in which:
    i. Leaf Nodes are attributes in A.
    ii. Each none leaf node is a gate Node with assigned Threshold.
    iii. The threshold k$_x$; 0 k$_x$ num$_x$ where num$_x$ is the number of children for node x.
    iv. If the Node is an AND k$_x$ = num$_x$.
    v. If the Node is an OR k$_x$ = 1.

B. Attribute based encryption:

Attribute-based encryption (ABE) is a concept of public-key cryptography. The cryptographic key, a message is encrypted for a receiver using the receiver's public-key. Identity-based encryption (IBE) changed the public-key cryptography by allowing the public-key to be an arbitrary string, e.g., the email. Key length is equal to the number of bits in an encryption algorithms key. If an n bit of keys is available then it gets an (2^n) possible key.

Step 1: User to identify some of attributes.

Step 2: Based on the attributes only the master secret key will be generated, the size of the key is measured only the attributes.

Step 3: Private key and public key is received by the server to user.

Step 4: User to encrypt the data using public key then only the encrypted data send to the server.

Step 5: The encrypted data is sending it to the server. The user checks the decryption process because the original data only to be encrypted are not.

C. Proxy signature verification:

Key generation the original signer selects a random number $x \in Z_q^*$ as the private key and the corresponding public key is $y = g^x \bmod p$. Then, the original signer publishes (p, q, g, y).

Proxy key generation should do following steps:

1. Select a random number $k_A \in Z_q^*$.
2. Compute $r_A = g^{kA} \bmod p$, and sets $s_A = (x + k_A r_A) \bmod q$.
3. Forward (r_A, s_A) to the proxy signer.

On receiving the (r_A, s_A), the proxy signer should verify the validity by checking equation $g^{sA} . y r_A^{rA} \bmod p$. The proxy signer accepts s_A, if the equation holds, and continues following steps.

Proxy signature: the original signer to create a signature $S(s_A, m)$ using the proxy key s_A.

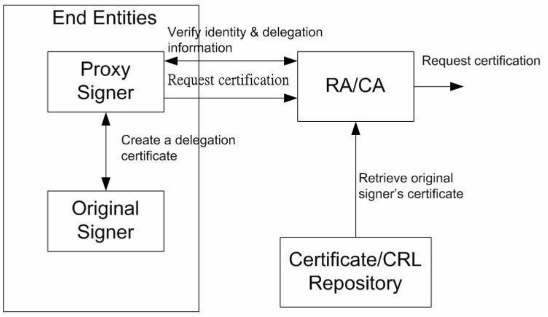
End Entities
Proxy Signer
Original Signer
Verify identity & delegation information
Request certification
RA/CA
Request certification
Create a delegation certificate
Retrieve original signer's certificate
Certificate/CRL Repository

Figure 2: proxy signature

1. Select a random $k \in Z_q^*$.
2. Compute $r = (g'^k \bmod p) \bmod q$.
3. Set $s = k^{-1} (h(m) + s_B r) \bmod q$.

The proxy signature is the tuple (g', r_A, e', r, s).

Proxy Signature Verification

To verify the proxy signature (g', r_A, e', r, s) on message m, a verifier should:

1. Query checks the proxy key for the original certificate is valid.
2. Verify that $1 \leq r \leq q$ and $1 \leq s \leq q$; if not holds, then reject the signature.
3. Compute $w = s^{-1} \bmod q$.
4. Compute $u_1 = w \cdot h(m) \bmod q$, $u_2 = rw \bmod q$, and $u_3 = e'u_2 \bmod q$.
5. Compute $v \cdot (g'^{u1}\, r_A^{u2}\, y^{u3} \bmod p) \bmod q$.

Accept the signature if $v = r$.

D. Quantization:

The quantization process is inherently loss because of the many-to-one mapping process. Concatenating outputs from multiple hash (master key information) functions provides private and public key information

C=Ek(M)

Where the product is the public key information, i.e., the generated cipher text with n user, we provide a unique attribute based id with Key specification as private key Therefore,

[n (n-1)]/2

$$\left(C_{\langle g^*\rangle}^{(\alpha^{-1})}, (\alpha^{-1})_i\right) = \mathsf{Inverse}\left(n, t, \hat{g}, \hat{h}, \left(C_{\langle g\rangle}^{(\alpha)}, \alpha_i\right)\right)$$

E. Coercion:

The goal of the coercion system is to force another party to act an involuntary manner. Coercion is an object to a primitive type. Input: H, an invertible function with attributes; K, Session key with exclusive OR operator,

$$\begin{aligned} R &= R \oplus H(L) \\ L &= L \oplus H(R \oplus k) \\ R &= R \oplus H(L \oplus k) \end{aligned}$$

Let (Lplain, Rplain) be the plain text, (Lcipher, Rcipher) be the corresponding cipher text. Where the inverse product with Lcipher gives the fully retrieved message and inverse product without Lciper gives transformed partial message.

$$k = H^{-1}(R_{cipher} \oplus H(L_{plain}) \oplus R_{plain}) \oplus L_{cipher}$$

$$k = H^{-1}(R_{cipher} \oplus H(L_{plain}) \oplus R_{plain})$$

III.EXPERIMENTAL DISCUSSION

The attribute based encryption is a public key. Attributes are used to generate the master secret key before the public and private key generation of proxy server. Only the proxy server to generate the public and private key then it transfers to sender. User transfers the plain text into decrypted data using public key. User verification process only encrypted data converted into decrypted data. The encrypted data only to transfer into the proxy server, then it stored in cloud storage area.

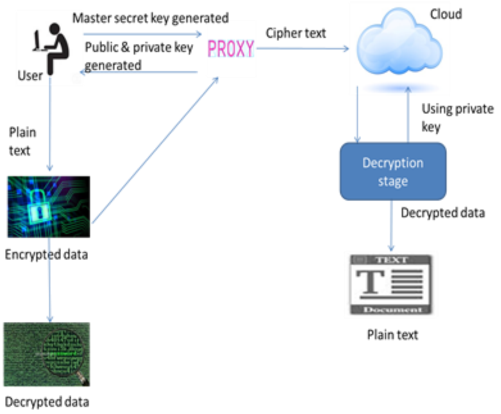

Figure 3: Architecture diagram

Decryption process is started in further data transformation. Based on the cipher text keys only the two keys are produced in the decryption process. The transform key and retrieve keys are to be converting the cipher text into a plain text. Transform key is to convert the cipher text into a partially cipher text data. Retrieve key is used in the partially cipher text data it will be converted into a plain text. Then the plain text and then the third party created plain text are compared it to verification and validation process. In between the proxy server is verification process to verify the private key for the user.

The master secret key is generated by using attribute based encryption. Based on the attributes only the master key is created. Then the proxy generates the public and private key to transfer the keys in sender via email. The access structure specification is uses the security purpose. Proxy signature verification algorithm is used in the proxy verification process. User checks the proxy server using the private key. Quantization is a many to one process for finding the receiver. Coercion is and retrieve key is used in find out the plain text.

MODULES DESCRIPTION

Attributes / Master secret key generation:

Attributes and master keys are used in number of operation is done. Here attributes are in string of characters and then based on the attributes only the master secret key will be generated by using XML generation. The key value/size is determining only the details of attributes.

Access structure specification:

This section is used only the security purpose. And then multiple parties need to work together to obtain a resource. The encrypted data is defined dynamically.

Private key generation:

Private key is known only the parties that exchange secret messages.

Proxy Application Generate Key

```
string key0 = objConnect.Encrypt(lblUName.Text.Trim().Substring(0, 3) + eml.Text.Trim().Substring(0, 5), false);
string key = objConnect.Encrypt(lblPvt.Text.Trim().Substring(0, 2) + eml.Text.Trim().Substring(0, 2) +
               lblDesi.Text.Trim().Substring(0, 2), true);
```

These are all the private and public key generation process.

Transform/Retrieve key generation:

Decryption strategy is allowing the cipher text then it will be converting two keys for decryption process. Transform key is partially decrypting the data then it will go the process of retrieve key. After it decrypt the partially cipher text into the original/plain text.

Performance analyses:

Finally the key strength is visualized graphically. The real time process of data transformation is done in sender to receiver.

IV. SECURITY ANALYSES

A. Bilinear maps:

Let G and GT be two multiplicative cyclic groups of prime order p. Let g be a generator of G and e :G^{x}G-> GT be a bilinear map with the properties:

1. Bilinearity: for all u;v 2 G and a;b 2 Zp, we have e(ua;vb) = e(u;v)ab.
2. Non-degeneracy: e(g;g) 6= 1.

The bilinear group operate G and bilinear map. e : G^{x}G->GT are both efficiently computable.

B. Access structure:

Definition 1 (Access Structure [5]) Let {P1, P2, : : :, Pn} be a set of parties. A collection A_2{P1;P2;:::;Pn} is monotone if 8B;C : if B 2 A and B _C then C ε A. An access structure is a collection (resp., monotone collection) A of non-empty subsets of {P1;P2; : : : ;Pn}, i.e., A _ 2{P1;P2;:::;Pn}n\{ϕ}. The sets A are called the authorized sets, and the sets are not in A are called the unauthorized sets.

V. PERFORMANCE ANALYSES

Our implementation adopts the key encapsulation mechanism, where the ABE cipher text is the encryption of a symmetric key and the message is encrypted separately using a symmetric encryption scheme under this. These modifications reduce the sizes of the ABE cipher text and the partially-decrypted cipher text by two elements in, respectively, without sacrificing security and verifiability.

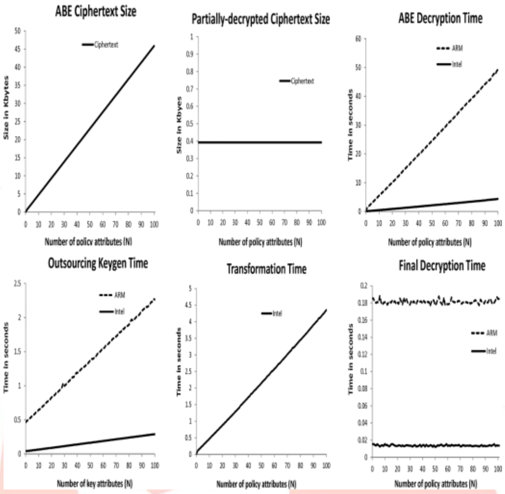

Figure 4: Various performance analyse details

In our experiments, we do not consider the effect of symmetric encryption. Thus, all the datum on decryption time and cipher text size presented in Fig. 2 are only associated with the key encapsulation variant of our ABE scheme. For each cipher text policy, we repeat our experiment 100 times on the PC and 30 times on the ARM device and we take the average values as the experimental results. In Fig. 2, we show the size of standard ABE cipher text and partially-decrypted cipher text, the standard ABE decryption time on the Intel and the ARM platforms, the time of generating an outsourcing key, the time of transforming the ABE cipher text, and the time of decrypting the transformed cipher text on the Intel and the ARM platforms.

VI. CONCLUSION

Check the attributes and master secret key based on encryption and decryption. Proxy server is an intermediate between the end-users. The proxy server successfully sends the decrypted data into the receiver, then the user to verify the proxy server because of the keys are all generated by a proxy server only. The verification and validation is successful in decrypt and encrypt process. The entire process will be done on cloud storage, so the each and every process will be very complicated. In future proxy server how to be handled in multi cloud storage is to be exercised.

REFERENCES

[1]. V.Goyal, O.Pandey, A.Sahai, and B.Waters, "**Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data**" ACM Conf. Computer and Communications Security, 2006.

[2]. R. Ostrovsky, A. Sahai, and B. Waters, "**Attribute-based encryption with non-monotonic access structures**," in Proc. ACM Conf. Computer and Communications Security, 2007, pp. 195–203.

[3]. A. B. Leek and B. Waters, "**Unbounded HIBE and attribute-based encryption**," in Proc. EUROCRYPT, 2011, pp. 547–567.

[4]. N.Attrapadung, J.Herranz, F.Laguillaumie, B.Libert, E. de Panafieu, and C.Ràfols, "**Attribute-based encryption schemes with constant-size cipher texts**," Theory. Compute. Sci., vol. 422, pp. 15–38, 2012.

[5]. M. Green, S. Hohenberger, and B.Waters, "**Outsourcing the decryption of ABE cipher texts**," in Proc. USENIX Security Symp, San Francisco, CA, USA, 2011.

[6]. T. Okamoto and K. Takashima, "**Fully secure functional encryption with general relations from the decisional linear assumption**," in Proc. CRYPTO, 2010, pp. 191–208.

[7]. T. Okamoto and K. Takashima "**Fully Secure Multi-authority Cipher text-Policy Attribute-Based Encryption without Random Oracles**" ACM Conf. Computer and Communications Security, 2006.

[8]. A. Sahai and B. Waters. "**Fuzzy Identity-Based Encryption**" in Proc. EUROCRYPT, 2005, pp. 457–473.

[9]. G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "**Improved proxy re-encryption schemes with applications to secure distributed storage**," in Proc. NDSS, San Diego, CA, USA, 2005.

[10]. NDSS, San Diego, CA, USA, 2005. "**Leakage-resilient Attribute-based Encryptions with Fast Decryption: Model, Analysis and Construction**" Public Key Cryptography, 2013.

[11]. A. B. Lewko and B. Waters, "**Decentralizing attribute-based encryption**, "in Proc. EUROCRYPT, 2011, pp. 568–588.

[12]. A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "**Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption**," in Proc. EUROCRYPT, 2010,pp. 62–91.

[13]. B. Chevalier- Memes, J.-S. Coron, N. McCullagh, D. Backaches, and M. Scott, "**Secure delegation of elliptic-curve pairing**," in Proc.CARDIS, 2010, pp. 24–35.

[14]. B. G. Kang, M. S. Lee, and J. H. Park, "**Efficient delegation of pairing computation**," IACR Cryptology ePrint Archive, vol. 2005, p. 259, 2005.

[15]. C. Gentry and S. Halevi, "**Implementing gentry's fully-homomorphic encryption scheme**," in Proc. EUROCRYPT, 2011, pp. 129–148.

[16]. C. Gentry, "**Fully homomorphic encryption using ideal lattices**," in Proc. STOC, 2009, pp. 169–178.

[17]. S. Hohenberger and B. Waters, "**Attribute-based encryption with fast decryption," in Proc. Public Key Cryptography**, 2013, pp. 162–179.

[18]. J. B. Nielsen, "**Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case**," in Proc. CRYPTO, 2002, pp. 111–126.

[19]. S. Goldwasser and Y. T. Kalai, "**On the (in)security of the fiat-shamir paradigm**," in Proc. FOCS, 2003, pp. 102–113.

[20]. M. Bellare, A. Boldyreva, and A. Palacio, "**A uninstantiable random oracle model scheme for a hybrid-encryption problem**," in Proc. EUROCRYPT, 2004, pp. 171–188.

[21]. R. Gennaro, C. Gentry, and B. Parno, "**Non-interactive verifiable computing: Outsourcing computation to un trusted workers**," in Proc. CRYPTO, 2010, pp. 465–482.

[22]. K.-M. Chung, Y. T. Kalai, and S. P. Vadhan, "**Improved delegation of computation is using fully homomorphic encryption**," in Proc. CRYPTO, 2010, pp. 483–501.