Attribute-based re-encryption scheme in the standard model

4 authors, including:

Qiuliang Xu
Shandong University
90 PUBLICATIONS **188** CITATIONS

Attribute-Based Re-Encryption Scheme in the Standard Model

☐ **GUO Shanqing[1], ZENG Yingpei[2], WEI Juan[1], XU Qiuliang1[†]**

1. School of Computer Science and Technology, Shandong University, Jinan 250101, Shandong, China;

2. State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210093, Jiangsu, China

Abstract: In this paper, we propose a new attribute-based proxy re-encryption scheme, where a semi-trusted proxy, with some additional information, can transform a ciphertext under a set of attributes into a new ciphertext under another set of attributes on the same message, but not vice versa, furthermore, its security was proved in the standard model based on decisional bilinear Diffie-Hellman assumption. This scheme can be used to realize fine-grained selectively sharing of encrypted data, but the general proxy re-encryption scheme severely can not do it, so the proposed scheme can be thought as an improvement of general traditional proxy re-encryption scheme.

Key words: attribute-based; re-encryption scheme; standard model; network storage

CLC number: TP309.7; TN 918.4

0 Introduction

In the past, when Alice asks mail server to forward her encrypted email to Bob, the mail server only decrypts the encrypted email using the secret key stored in the mail server, re-encrypts it using Bob's public key and then forward it to Bob. The obvious problem with this strategy is that the mail server must be completely trusted. which is an unlikely real-world expectation. Unfortunately, there is no better solution available until Mambo et al[1] suggested that there might be more efficient approaches which involve partial decryptions in 1997, but offered no additional security benefits for Alice's secret key. In 1998, the approach[2] based on the ElGamal cryptosystem was proposed, which introduced the notion of a "re-encryption key" $\mathrm{rk}_{A \to B}$. Using $\mathrm{rk}_{A \to B}$ allowed the proxy to re-encrypt from one secret key to another without ever learning the plaintext. In 2003, Dodis et al[3] proposed a general scheme for proxy encryption (not re-encryption) using only standard public key cryptosystems. However, their system required Alice and Bob to pre-share a secret. Although it is clear that such a pre-sharing phase is possible to accomplish securely (e.g., via Diffie-Hellman), it is undesirable. It may be that Alice and Bob has no prior relationship, and bidirectional communication is impossible. In 2005, Ateniese et al[4] constructed the first unidirectional, collusion resistant re-encryption without any required pre-sharing between parties, based on bilinear maps.

When the proxy re-encryption schemes were applied into the fields of the network storage services, there exist some drawbacks. The sensitive information stored in the network storage services was kept in encrypted

form for protecting its owner's privacy. Suppose a particular user wants to grant decryption access to a party to all of its Internet traffic logs for all entries on a particular range of dates that had a source IP address from a particular subnet. The user either needs to act as an intermediary and decrypt all relevant entries for the party or must give the party its private decryption key, and thus let it have access to all entries. Neither of these options is particularly appealing. In this paper, we present a new encryption cryptosystem based on the proposed scheme in Ref.[5] and a general proxy re-encryption scheme, named by ABRS (attribute-based re-encryption scheme), which allows the proxy to convert ciphertexts labeled with set of attributes to another ciphertexts labeled with another set of attributes. This kinds of encryption cryptosystem can both imply the encrypted data sharing between users and fine-grained selectively access control to encrypted data.

1 Attribute-Based Re-Encryption Scheme's Definition and Its Security Model

1.1 Re-Encryption Scheme

A re-encryption scheme[4] is a 6-tuple of algorithms, which includes the following:

● Setup: Takes a security parameter k and returns params (system parameters). The system parameters include a description of a finite message space M, and a description of a finite ciphertext space C.

● KeyGen[$(1^k) \rightarrow$(pk, sk)]: On inputting the security parameter 1^k, the key generation algorithm KeyGen outputs a public key pk and a secret key sk.

● Encryption[(pk, $m) \rightarrow c$]: This is a randomized algorithm. On inputting a set of attributes γ_1, and a message m in the message space, the encryption algorithm Enc outputs a ciphertext C.

● RKExtract[$(sk_1, pk_2) \rightarrow rk_{1 \rightarrow 2}$]: Input a secret key sk_1 and another set of attributes γ_2, the re-encryption key generation algorithm RKExtract outputs an unidirectional re-encryption key $rk_{1 \rightarrow 2}$.

● Re-Encryption[$(rk_{1 \rightarrow 2}, c_1) \rightarrow c_2$]: On inputting a re-encryption key $rk_{1 \rightarrow 2}$ and a ciphertext c_1, the re-encryption algorithm ReEncryption outputs an re-encrypted ciphertext c_2 or "Reject".

● Decryption[(sk, $c) \rightarrow m$]: On inputting a secret key sk and a ciphertext c, the decryption algorithm Decryption outputs a message m in the message space or "Re-

ject".

1.2 Correctness

The correctness property has two requirements. For any message m in the message space and any key pair $(\gamma, sk), (\gamma', sk') \leftarrow KeyGen(1^k)$ .The following two conditions must hold:

Decryption(sk, Encryption(γ, m))
$= m$, Decryption(sk′, ReEncryption(RKExtract(sk, γ'),
Encryption(γ, m)))
$= m$

1.3 Its Security Model

Let S be a scheme defined as a tuple of algorithms (Setup, KeyGen, Encryption, Decryption, RKExtract, Re-Encryption), its security is defined according to the following game, similar with Ref.[6].

Initialization: The adversary declares the set of attributes γ with access control structure ψ, that he wishes to be challenged upon.

Setup: The challenger runs the Setup algorithm of ABE and gives the public parameters to the adversary.

Phase 1: The adversary is allowed to issue queries for private keys for many access structures ψ_i, for all i, $\gamma \notin \psi_i$.

a) For adversary's query of KeyGen(ψ_i), return D_{ψ_i} =keyGen(params, msk, ψ_i) to the adversary.

b) For adversary's query of RKExtract(γ_1, γ_2), where $\gamma_1 \neq \gamma_2$, return $rk_{\psi_2 \rightarrow \psi_2}$ =RKGen(params,Keygen (params, msk, ψ_1), ψ_1, ψ_2) to the adversary.

c) For adversary's query of Decryption(ψ_1, c), if ATK=CCA, then return m=Decryption(params, Keygen (params, msk), ψ_1, c) to the adversary. Otherwise if ATK = CPA, return $\perp$ to the adversary.

d) For adversary's query of ReEncryption(ψ_1, ψ_2, c), if ATK=CCA then derive a Re-Encryption key $rk_{\psi_1 \rightarrow \psi_2}$ =RKGen(params, Keygen (params, msk, ψ_1), ψ_1, ψ_2) and return c'=Reencrypt (params, $rk_{\psi_1 \rightarrow \psi_2}$, ψ_1, ψ_2, c) to the adversary. If ATK=CPA, return $\perp$ to the adversary.

Note that the adversary is not permitted to choose ψ, otherwise trivial decryption is possible using keys extracted during this phase.

Challenge: The adversary submits two equal lengthy messages M_0 and M_1. The challenger flips a random coin b, and encrypts M_b with γ. The ciphertext c^* is passed to the adversary.

Phase 2: Phase 1 is repeated with the following restrictions. Let C be a set of ciphertext Access control structure pairs, initially containing the single pair $\langle c^*, \psi \rangle$, let C' be the set of all possible values derived via (one or more) consecutive calls to Re-encrypt:

(a) Adversary is not permitted to issue any query of Decryption(ψ, c), where $\langle c, \psi \rangle \in (C - C')$.

(b) Adversary is not permitted to issue any queries Keygen(ψ) or RKExtract(γ_1, γ_2) that would permit trivial decryption of any ciphertext in ($C - C'$).

(c) Adversary is not permitted to issue any query of ReEncryption(ψ_1, ψ_2, c) where ψ possesses the keys to trivially decrypt ciphertexts under ψ_2 and $\langle c, \psi_1 \rangle \in (C \cap C')$. On successful execution of any re-encrypt query, let c' be the result and add the pair $\langle c', \psi_2 \rangle$ to the set C.

Guess: The adversary outputs a guess b' of b. The advantage of an adversary ψ in this game is defined as $\Pr[b'=b] - 1/2$.

Definition 1 An attribute-based proxy re-encryption scheme is secure in the selective-set model of security if all polynomial time adversaries have at most a negligible advantage in the selective-set game.

2 Construction for Attribute-Based Proxy Re-Encryption Scheme

There are four parties involved in this attribute-based re-encryption system, delegator, proxy, delegatee and PKG (private key generator). On receiving a ciphertext by delegator's public key, the proxy re-encrypts it into ciphertexts that the delegatee who holds a secret key can decrypt, using a re-encryption key generated by the PKG for a particular delegatee. In addition, the bilinear maps and Lagrange interpolation[5] will be used in this scheme.

Setup: Define the universe of attributes $U = \{1, 2, \cdots, n\}$. Now, for each attribute $i \in U$, choose a number t_i uniformly at random from $\mathbf{Z}_p$. Finally, choose y uniformly at random in $\mathbf{Z}_p$. The public parameters PK are: $T_1 = g^{t_1}, \cdots, T_{|\Omega|} = g^{t_{|\Omega|}}, Y = e(g,g)^y$. The master keys MK are: $t_1, \cdots, t_{|\Omega|}, y$.

Encryption (M, γ, PK): To encrypt a message $M \in G_2$ under a set of attributes γ, choose a random value $r \in \mathbf{Z}_p$ and publish the ciphertext as $E = (\gamma, E' = MY^r, \{E_i = T_i^r\}_{i \in \gamma})$

KeyGen (T, MK): The algorithm outputs a key that enables the user to decrypt a message computed under a set of attributes γ if and only if $T(\gamma) = 1$. The algorithm proceeds as follows. First choose a polynomial q_x for each node x (including the leaves) in the tree T, and these polynomials are chosen in the following way in a top-down manner, starting from the root node r. For each node x in the tree, set the degree d_x of the polynomial q_x to be one less than the threshold value k_x of that node,

that is $d_x = k_x - 1$. Now, for the root node r, set $q_r(0) = y$ and d_r other points of the polynomial q_r randomly to define it completely. For any other node x, set $q_x(0) = q_{\text{parent}(x)}(\text{index}(x))$ (The function index(x) returns such a number associated with the node x. Where the index values are uniquely assigned to nodes in the access structure for a given key in an arbitrary manner) and choose d_x other points randomly to completely define q_x. Once the polynomials have been decided, for each leaf node x, we give the following secret value to the user: $D_x = g^{q_x(0)/t_{i_{q_x}}}$ where $i = \text{att}(x)$ (The function att(x) is defined only if x is a leaf node and denotes the attribute associated with the leaf node x in the tree), The set of above secret values is the decryption key D.

RKExtract (γ_1, γ_2): $\mathrm{RK}_1^{A \to B} = t_1'/t_1, \mathrm{RK}_1^{A \to B} = t_2'/t_2, \cdots, \mathrm{RK}_{|\gamma|}^{A \to B} = t_{|\psi|}'/t_{|\psi'|}$, where A, B's access control structure is ψ and ψ' separately and number of attributes included in the ψ' is no more than number of attributes included in the ψ.

ReEncryption: $C_b = (\gamma', E' = MY^{rs}, \{E_i = ((T_i^r)^{\mathrm{RK}_i^{A \to B}})^s\}_{i \in \gamma})$, where γ' is a finite set with user B's complete attributes and $s \in \mathbf{Z}_p$

Decryption (E, D): do some Decryption procedure like Ref.[1].

Correctness: Correctness for first-level ciphertexts (i.e., those produced by Encryption) has been shown[1].

The correctness under re-encryption is shown as follows: Given a first-level ciphertext $c_{\psi_1} = (\gamma_1, E' = MY^r, \{E_i = T_i^r\}_{i \in \gamma_1}$ and a correctly-formed re-encryption key ($\mathrm{RK}_1^{A \to B} = t_1'/t_1, \mathrm{RK}_1^{A \to B} = t_2'/t_2, \cdots, \mathrm{RK}_{|\gamma_1|}^{A \to B} = t_{|\psi_1|}'/t_{|\psi_1'|}$), we obtain the "second-level" ciphertext:

$$C_{\psi_2} = (\gamma', E' = MY^{rs}, \{E_i = (T_i^r)^{\mathrm{RK}_i^{A \to B}}\}_{i \in \gamma}).$$

By computing

$$C_{\tau_2} = (\gamma', E' = MY^{rs}, \{E_i = (T_i^{rs})^{\mathrm{RK}_i^{A \to B}}\}_{i \in \gamma})$$

$$= (\gamma', E' = MY^{rs}, \{E_i = (T_i'^{rs})\}),$$

Then use the defined recursive function[1] like：
Decrypt (E, D, x) =

$$\begin{cases} e(D_x, E_i) = e(g^{q_x(0)/t_i}, g^{rs \cdot t_i}) = e(g,g)^{rs \cdot q_x(0)}, & \text{if } i \in \gamma \\ \perp, & \text{otherwise} \end{cases}$$

We now consider the recursive case when x is a non-leaf node. The algorithm DecryptNode(E, D, x) then proceeds as follows: For all nodes z that are children of x, it calls DecryptNode(E, D, z). In addition, stores the output as F_z. Let S_x be an arbitrary k_x sized set of child nodes z such that $F_z \neq \perp$. If no such set exists then the node was not satisfied and the function returns $\perp$. Oth-

erwise, we compute:

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,s'_x}(0)}$$

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

$$= \prod_{z \in S_x} (e(g,g)^{rs \cdot q_{\mathrm{parent}(z)}(\mathrm{index}(z))})^{\Delta_{i,s'_x}(0)}$$

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

$$= e(g,g)^{rs \cdot q_x(0)}$$

where $i = \mathrm{index}(z), s'_x = \{\mathrm{index}(z) : z \in S_x\}$.

Now since the decryption algorithm calls the recursive function on the root of the tree, Y^{rs} can be easily computed, then use Y^{rs} to divide $E' = MY^{rs}$, and we will get the original message m.

Security: Our security definition has two parts: standard security, requiring that the cryptosystem remain semantically-secure[7] even when all re-encryption keys are public, and master secret security, where a delegator's master secret key is not recoverable. We next show that ABRS scheme defined above meets them in the attribute-based selective-set model if the decisional bilinear Diffie-Hellman assumption holds in G_1, G_2[8].

3   Proof of Security

We will use the similar method used in Ref.[5] to prove that the security of our scheme in the attribute-based selective-set model reduces to the hardness of the decisional BDH assumption,

Theorem 1   If an adversary can break our scheme in the attribute-based selective-set model, then a simulator can be constructed to play the decisional BDH game with a non-negligible advantage.

Proof: Suppose that a polynomial-time adversary A exists, which can attack our scheme in the Selective-Set model with advantage ε. A simulator B is built that can play the decisional BDH game with advantage $\varepsilon / 2$. The simulation proceeds as follows:

We first let the challenger set the groups G_1 and G_2 with an efficient bilinear map, e and generator g. The challenger flips a fair binary coin μ, outside of B' s view. If $\mu = 0$, the challenger sets $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{abc})$; otherwise it sets $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ for random a, b, c, z. We assume the universe U is defined.

Initialization: The simulator B runs adversary A. If Alice with γ_{Alice} wants to delegate her decryption rights

to Bob with γ_{Bob}, γ_{Alice} and γ_{Bob} are selected by adversary A, it wishes to be challenged upon.

Setup: The simulator sets the parameter $Y = e(A, B) = e(g, g)^{ab}$. For all $i \in U$, it sets T_i as follows: if $i \in \gamma_{\mathrm{Alice}}$, it chooses a random $r_i \in \mathbf{Z}_p$ and sets $T_i = g^{r_i}$ (thus $t_i = r_i$). Otherwise, it chooses a random $\beta_i \in \mathbf{Z}_p$ and sets $T_i = g^{b\beta_i} = B^{\beta_i}$ (thus, $t_i = b\beta_i$). The similar things will be done with T'_j and γ_{Bob}. Then the public parameters are given to A,

Phase 1: adversary A adaptively makes requests for the keys corresponding to any access structures ψ such that the challenge set γ does not satisfy ψ. Suppose adversary A makes a request for the secret key for an access structure ψ, where $\psi(\gamma) = 0$.

To generate the secret key, simulator B needs to assign a polynomial Q_x of degree d_x for every node in the access tree ψ. We will use two procedures, named PolySat and PolyUnsat, which are defined in[1].

To give keys for access structure ψ, simulator first runs PolyUnsat(ψ, γ, A) to define a polynomial q_x for each node x of ψ. Notice that for each leaf node x of ψ, we know q_x completely if x is satisfied; if x is not satisfied, then at least $g^{q_x(0)}$ is known (in some cases q_x might be known completely). Furthermore $q_r(0) = a$.

Simulator now defines the final polynomial $Q_x(\cdot) = bq_x(\cdot)$ for each node x of ψ. Notice that this sets $y = Q_x(0) = ab$. The key corresponding to each leaf node is given using its polynomial as follows. Let

$i = \mathrm{att}(x)$,

$$D_x = \begin{cases} g^{Q_x(0)/t_i} = g^{bq_x(0)/r_i} = B^{q_x(0)/r_i}, & \text{if } \mathrm{att}(x) \in \gamma \\ g^{Q_x(0)/t_i} = g^{bq_x(0)/b\beta_i} = B^{q_x(0)/\beta_i}, & \text{otherwise} \end{cases}$$

When adversary A submits (RKExtract, γ_1, γ_2), where γ_1 and γ_2 do not satisfy ψ and ψ', simulator B selects:

$$\mathrm{RK}_{i,j}^{A \to B} = \begin{cases} t'_j / t_i, & i \in \lambda \text{ and } j \in \lambda' \\ \beta^2 t'_j / t_i, & i \notin \lambda \text{ and } j \notin \lambda' \\ \beta t'_j / t_i, & i \notin \lambda \text{ and } j \in \lambda' \\ t'_j / t_i, & i \in \lambda \text{ and } j \notin \lambda' \end{cases}$$

Given an encryption for Alice, $E(m) = (\gamma, E' = mY^r, \{E_i = T_i^r\}_{i \in \gamma})$, the proxy can transform it into an encryption for Bob by releasing: $(\gamma', E' = mY^{rs}, \{E_i = ((T_i^r)^{\mathrm{RK}_i^{A \to B}})^s\}_{i \in \gamma'})$

Therefore, the simulator is able to construct a private key for the access structure ψ. Furthermore, the distribution of the private key for ψ is identical to that in the original scheme.

Challenge: The adversary A, will submit two chal-

lenge messages m_0 and m_1 to the simulator. The simulator flips a fair binary coin b, and returns an encryption of m_b. The ciphertext is output as: $E = (\gamma,\ E' = m_b Z,\ \{E_i = C^{r_i}\}_{i \in \gamma})$

If $u=0$, then $Z=\mathrm{e}(g, g)^{abcd}$. If we let $rs = c$, then we have $Y^{rs}=\mathrm{e}(g, g)^{abc}$, and $E_i = (g^{r_i})^c = C^{r_i}$, Therefore, the ciphertext is a valid random encryption of message m_b.

Otherwise, if $u =1$, then $Z=\mathrm{e}(g, g)^z$ We then have $E' = m_b \mathrm{e}(g, g)^z$, since z is random, E' will be a random element of G_2 from the adversaries view and the message contains no information about m_b

Phase 2: The simulator acts exactly as it does in Phase 1. Let C be a set of ciphertext/Access control structure pairs, initially containing the single pair $\langle c^*, \psi \rangle$, let C' be the set of all possible values derived via (one or more) consecutive calls to Re-encryption:

(a) Adversary is not permitted to issue any query of the form Decryption(ψ, c), where $\langle c, \psi \rangle \in (C-C')$.

(b) Adversary is not permitted to issue any queries Keygen(ψ) or RKExtract(γ_1, γ_2)that would permit trivial decryption of any ciphertext in $(C-C')$.

(c) Adversary is not permitted to issue any query of ReEncryption(ψ_1,ψ_2,c) where ψ possesses the keys to trivially decrypt ciphertexts under ψ_2 and $\langle c, \psi_1 \rangle \in (C \cap C')$. On successful execution of any re-encrypt query, let c' be the result and add the pair $\langle c', \psi_2 \rangle$ to the set C.

Guess: A will submit a guess b' of b. If $b' = b$ the simulator will output $u' =0$ to indicate that it was given a valid BDH-tuple otherwise it will output $u'=1$ to indicate it was given a random 4-tuple.

As shown in the construction the simulator's generation of public parameters and private keys is identical to that of the actual scheme.

In the case where $u=1$ the adversary gains no information about b. Therefore, we have $\Pr[b'\neq b|u=1]=1/2$. Since the simulator guesses $u'=1$ when $b'\neq b$, we have $\Pr[b'=b|u=1]=1/2$.

If $u = 0$. then the adversary sees an encryption of m_b. The adversary's advantage in this situation is ε by definition. Therefore, we have $\Pr[b'=b|u=0]=1/2+\varepsilon$, since the simulator guesses $u'=0$ when $b'=b$, we have $\Pr[b'=b|u=0]=1/2+\varepsilon$.

The overall advantage of the simulator in the decisional BDH game is that: $1/2 \times \Pr[b'=b|u=0]+1/2 \times \Pr[b'=b|u=1]-1/2=1/2 \times (1/2+\varepsilon)+1/2 \times 1/2-1/2=\varepsilon/2$, which is a non-negligible advantage.

4 Conclusion

In this paper, we propose attribute-based re-encryption system based on the attribute-based encryption scheme and general proxy re-encryption scheme, and we also propose its security proof in the standard model based on decisional bilinear Diffie-Hellman assumption. Under this scheme, the semi-trusted proxy, with some additional information, can transform a ciphertext encrypted under a set of attributes into a new ciphertext under another set of attributes on the same message, but not vice versa. This scheme can be used to devise access control model with encrypted data, which has an important application in the field of network storage. In the future, we will focus on its effective implementation to secure network storage.

References

[1] Mambo M, Okamoto E. Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts[J]. *IEICE Trans on Fundamentals*, 1997, **E80-A**(1): 54-63.

[2] Blaze M, Bleumer G, Strauss M. Divertible Protocols and Atomic Proxy Cryptography[C]//*Proceedings of Eurocrypt*'98 (*LNCS* 1403). Heidelberg: Springer-Berlin, 1998: 127-144.

[3] Dodis Y, Ivan A. Proxy Cryptography Revisited[C]// *Proceedings of the* 10*th Annual Network and Distributed System Security Symposium* (*NDSS*). San Diego: the Internet Society, 2003

[4] Ateniese G, Fu K, Green M, *et al*. Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage[J]. *ACM Transactions on Information and System Security* (*TISSEC*), 2006, **9**(1): 1-30.

[5] Goyal V, Pandeyy O, Sahaiz A, et al. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data [C]//*Proceedings of the* 13*th ACM conference on Computer and communications security*. New York：ACM Press, 2006: 89-98.

[6] Boneh D, Boyen X. Efficient Selective-ID Secure Identity-Based Encryption without Random Oracles[C]// *Proceedings of Eurocrypt*'2004 (*LNCS* 3027). Berlin: Springer- Verlag, 2004: 223-238.

[7] Goldwasser S, Micali S. Probabilistic Encryption[J]. *Journal of Computer and System Sciences*, 1984, **28**(2): 270-299.

[8] Boneh D, Franklin M K. Identity-Based Encryption from the Weil Pairing[C]//*CRYPTO*'2001 (*LNCS* 2139), Santa Barbara: Springer-Verlag, 2001: 213-229.

□