CBP: Coalitional Game-Based Broadcast Proxy Re-encryption in IoT

Sumana Maiti, Sudip Misra, *Fellow, IEEE*, and Ayan Mondal, *Member, IEEE*

***Abstract*—This paper proposes Coalitional Game-Based Broadcast Proxy Re-encryption in IoT — a broadcast proxy re-encryption for adding new IoT devices. The proxy re-encryption is extended to broadcast proxy re-encryption to prevent re-calculation of the re-encryption key. However, the group of recipients needs to be pre-determined before the calculation of the re-encryption key. If any new IoT device requires the same data, an individual re-encryption key is generated for him/her. Hence, generating individual re-encryption key is an overhead for the organization. We propose a re-encryption key updation for the broadcast proxy re-encryption method. If excessive IoT devices want to join the group of the existing recipient, then updation needs to be done learnedly as an excessive number of recipients in a group increases the computation cost of the decryption extremely for all the members of the group. Therefore, we use the coalitional game theory to estimate the optimal number of new recipients from all new recipients. We update the re-encryption key for the optimal number of members and a separate re-encryption key is calculated for other recipients. We prove the correctness of the CBP. We prove that if any recipient behaves maliciously, s/he cannot get the secret key of the organization.**

***Index Terms*—Proxy Re-encryption, Broadcast encryption, Cloud, Coalitional Game theory, Identity-based encryption.**

I. Introduction

DIfferent organizations and institutes store their data to the cloud server. The data may contain confidential information for some applications like healthcare, military, and vehicular application. The data owner stores his/her encrypted data to avoid data leakage. Whenever the data is required, it is downloaded and decrypted using a secret key. However, the data needs to be shared with another user with maintaining confidentiality. Proxy Re-encryption (P-RE) [1] is a proficient scheme to share confidential data with another user. P-RE is extended to the Broadcast Proxy Re-encryption (B-RE) [2] to delegate confidential cloud data with more than one recipients. The data owner requires to compute a single re-encryption key (ReKey) for more than one user and the ReKey is used to transform the original ciphertext to the re-encrypted ciphertext (ReText). The recipient of the set uses his/her secret key to recover the data. There are different existing B-RE schemes [2]–[4]. However, the group of recipients of ReText is predefined in the existing B-RE schemes. Some applications require adding a new recipient to the existing group after the ReText is calculated. If the data owner calculates a separate ReKey for each new user, it becomes an overburden for him/her. The reduction of the computation cost of the data owner is possible if we update the existing ReKey instead to re-compute it. However, if an excessive number of users wish to be added to the existing group, the decryption cost of the recipient grows linearly as the decryption cost is dependent on the count of recipients present in the group. Therefore, it needs to be decided on how many new recipients can be joined with the existing recipient to avoid the huge decryption cost.

A. Motivation

The owner of the cloud data shares his/her sensitive data with a user using P-RE. To reduce the cost of the data owner, a single ReKey is generated for more than one recipient using the concept of B-RE. However, the existing B-RE schemes consider that the set of the recipient is fixed at the time of ReKey computation. In a real-life scenario of mobile applications, the IoT devices may join the existing group of IoT devices after the ReKey and ReText computation. Fig. 1 shows a motivation scenario of the CBP scheme. Suppose, an organization provides a particular service to different IoT devices. Therefore, the devices should be registered to the organization. Initially, the service data is shared with a few numbers of recipients. Later, some new IoT devices want to join the group. If the organization generates ReKey for the addition of each new recipient, it incurs a high cost. It is better to update the ReKey instead of re-calculating the ReKey. However, in B-RE, the computation cost of the recipient depends on the number of members of the group. Hence, adding multiple IoT devices to the group becomes an overburden for the recipient. Therefore, it is needed to find

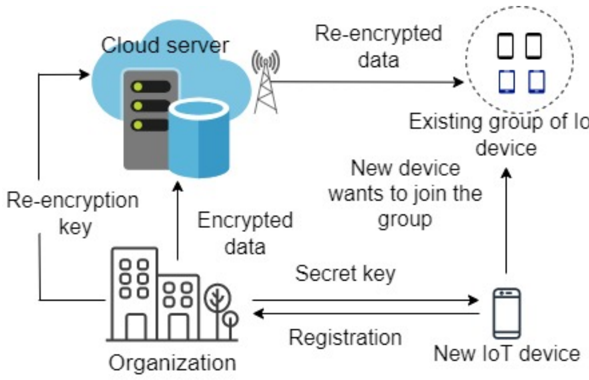

Fig. 1: Motivation scenario

Sumana Maiti is with the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala 147004 India, (E-mail: sumana.maiti@thapar.edu)

Sudip Misra is with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, India (Email:smisra@sit.iitkgp.ac.in)

Ayan Mondal is with the Department of Computer Science and Engineering, Indian Institute of Technology, Indore 453552, India (Email:ayanm@iiti.ac.in)

the optimal number of recipients who can join the group. Depending on the result, it updates ReKey and ReText.

B. Contribution

In CBP, we use coalitional game theory to decide on the joining of the new users. We find out the optimal number of new recipients who can join the existing recipient group based on the decision. We update the existing ReKey and ReText. The recipients of the existing group use the same secret key to decrypt the updated ReText. We analyze the proposed scheme and prove that the scheme is correct. Additionally, we prove that CBP is secure against the inside attacker. Finally, we implement CBP to show its efficiency over other existing schemes. The contributions of the proposed scheme are listed as follows.

1) We use coalitional game theory to find the optimal number of new IoT devices that can join the existing group.
2) We update the existing ReKey and ReText to add new IoT devices to the existing group of recipients.
3) We prove the correctness of the proposed scheme.
4) We prove that if any malicious recipient is present in the group of recipients, s/he cannot get the secret key of the organization.
5) We implement CBP and show its efficiency over other existing schemes.

II. Related work

A. Proxy Re-encryption

The concept of P-RE is initially introduced in Ref. [1], where a semi-trusted entity transforms the ciphertext of one user to a ciphertext of another user, without revealing any information of the plaintext. Various P-RE algorithms [5] [6] [7] [8] are proposed. A unidirectional P-RE scheme is proposed in Ref. [5], where more than one proxy server is required to transform the ciphertext. All the proxy servers hold a part of ReKey. Another unidirectional scheme is proposed in Ref. [6]. Here, one part of the secret key is given to the proxy, and another part is given to the recipient. PKI (Public Key Infrastructure) schemes need expensive certificates to verify the public key of the recipients. Identity-based P-RE scheme is proposed in Ref. [7], which is proven secure in the random-oracle model. Secure Identity-based P-RE is proposed in Ref. [9]. Here, the secret keys of the data owner, the recipients, and the identity of the recipient are not revealed. Chosen-ciphertext secure P-RE scheme is proposed in Ref. [10] based on the Decisional Bilinear Diffie-Hellman assumption.

B. Broadcast Proxy Re-encryption

P-RE is used to re-encrypt for a single recipient. The ReKey needs to be re-calculated, when multiple recipients are present, which is a headache for the data owner. Therefore, the conditional B-RE scheme is proposed in Ref. [2] using broadcast encryption [11], [12], where the data owner generates a ReKey for a set of recipients. The data owner specifies different conditions to control the re-encryption. A time-based P-RE scheme is proposed in Ref. [13]. Here, a time-based algorithm is used to control the time of re-encryption of the initial ciphertext. The data owner needs to know the recipients of the initial ciphertext. Conditional B-RE is proposed in Ref. [3] to forward the email in the cloud from one set of recipients to another set of recipients. The scheme is a chosen-plaintext attack secure. A conditional dynamic B-RE scheme is proposed in Ref. [14]. Here, another entity the broadcast center generates the ReKey on behalf of the data owner. A fine-grained access control conditional B-RE scheme is proposed in Ref. [15], where the ReKey is computed using a set of attributes. The recipients having matching attributes with the original ciphertext, decrypt the ReText. A chosen-ciphertext secure, collusion-resistant B-RE scheme is introduced in Ref. [16]. A revocable B-RE algorithm is introduced in Ref. [4]. The power of the revocation is given to the cloud server. The data owner outputs the list of the intended revoked users and delegates it to the cloud server, which updates the existing ReKey based on the revocation list to revoke some recipients from the group of existing recipients. A conditional B-RE scheme is proposed in Ref. [17], where the condition is used as a set of strings. A set of a string is used in the original ciphertext generation and another set of a string is used in the ReKey generation. If the similarity between these two sets of strings is greater than the threshold value, then the proxy server can re-encrypt the original ciphertext. Privacy-protected B-RE algorithm is introduced in Ref. [18], where a recipient cannot discover the identities of the other recipients of the group. An optimal group size of the B-RE system is calculated in Ref. [19]. The scheme balances the utilities of the data owner and the recipients.

C. Encrypted data Sharing in IoT devices

There are different existing works [20]–[24] on secure data sharing in IoT devices. An IoT-based proxy re-encryption scheme is proposed in Ref. [20]. The proposed scheme is bidirectional and it supports the multi-hop functionalities. The drawback of the scheme is that the sender needs the private key of the recipient to generate the re-encryption key. An attribute-based encryption scheme is proposed in Ref. [21], where partial encryption and decryption operations are done at the edge nodes. Hence, the computation costs of encryption and decryption are reduced for resource-constrained devices. A multi-user searchable encryption scheme is proposed in Ref. [22] to retrieve query-related encrypted cloud data for IoT devices. The drawback of the scheme is that the size of the master public key increases linearly with the total number of users. An attribute-based data sharing scheme is proposed in [23], where the data are collected by different vehicles and the data is stored in the blockchain.

D. Game theory in Broadcast Proxy Re-encryption

In B-RE algorithms [2]–[4], the game is defined between a challenger and an adversary and it is proved that the scheme is secure against chosen-plaintext attack and chosen ciphertext attack. There are different co-operative [25], [26] and non-cooperative game [27]–[29] in the existing literature, which can be used to decide security algorithms. However, these algorithms are rarely used in B-RE algorithms. A game-based security protocol is introduced in Ref. [30] in the social network, where any user can join or exit from the contact list of another user. In this scheme, a game is played between a good user and the service provider and between a bad user and the service provider. Nash bargaining is used to show that the players gain more utility of privacy and advertising if they are good users. To find the group size of recipients, non-cooperative bargaining is used in Ref. [19], which balances the utilities of the data owner and the recipients. Additionally, it increases the total utility of the system, while the total utility is calculated based on the utilities of the data owner, the proxy server, and the recipient.

E. *Inference*

The existing B-RE schemes allow the generation of a single ReKey for multiple recipients but it considers that the group of the recipients is predefined, which is not the case in real life. There are some existing B-RE algorithms for user revocation [4], [14]. However, in Ref. [14], a separate entity generates ReKey for the existing group of recipients and in Ref. [4], the power of revocation is given to the proxy server, which is not fully trusted. None of the existing schemes discuss about the joining of the new recipients. Hence, there is a need for a B-RE algorithm, where a new recipient can easily join the existing group of recipients. On the other hand, if we keep adding a new user to the existing group, the decryption cost of the recipients increases linearly. Therefore, our objective is to decide on whether adding the new recipients to the existing group is beneficial, or creating a new group is beneficial. We update or re-generate the ReKey based on the decision. Table I shows the differences between the existing B-RE schemes and the proposed CBP scheme. In our proposed scheme, we use new algorithms *JoinDecision* and *UpdateReKey* to add new recipients to the existing group. None of the existing B-RE scheme supports adding new recipients to the existing group of recipients. In *JoinDecision* algorithm, using coalitional game theory, we decide the number of optimal recipients who can join the existing group and then the existing ReKey is updated using *UpdateReKey* algorithm based on the identities of the new recipients.

III. System Model

The system model consists of three entities, namely the organization, the proxy server, and the group of IoT devices, which act as recipients. The organization generates the original ciphertext and stores it on the cloud server. Here, the organization also acts as the secret key generator, who generates a secret key for individual IoT devices. If the organization wishes to share the ciphertext with a group of recipients, it generates the ReKey and sends it to the proxy server, who converts the original ciphertext to ReText using the ReKey. The recipient can use his/her secret key to decrypt the ReText.

A. *Problem area*

Fig.2 shows the system model of the CBP scheme. Let us consider that in an organization, different IoT devices are registered with their identities for a particular service. The service manager of the organization authorizes the IoT devices with their identities and provides the secret keys for the identities. The organization initially stores its original ciphertext to the cloud server. Whenever s/he needs the data, s/he downloads it and decrypts it with the corresponding secret key. To share the data, s/he generates a ReKey for a group of IoT devices, and after receiving the ReKey, the proxy server converts the original ciphertext to the ReText. The receivers of the group can decrypt the ReText with their corresponding secret key. However, new IoT devices may want to request the same data from the organization. If the organization generates a separate ReKey for each new user, it becomes expensive for the organization as the computation cost increases linearly. If the organization updates the ReKey instead of re-generating the same, it is cost-efficient. However, if we keep adding a new IoT device to the existing group, it increases the cost of each receiver as they need to compute extra computation, which increases with the increase in the number of members in the group. It may also reach the limit of maximum allowable receivers for each re-encryption. Therefore, the organization should make a decision, on whether new members can be joined in the existing group or a new ReKey should be generated based on the computation cost of the system. In reality, only a few of the recipients decrypt the ReText because of the associated computation overhead of the decryption.

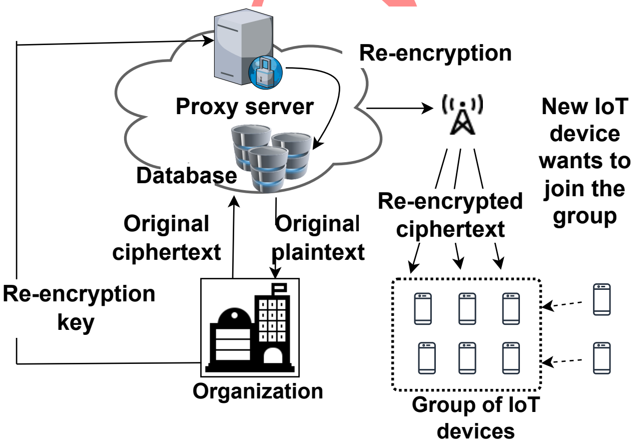

Fig. 2: System model

Assumptions: The assumptions of the proposed scheme are listed as follows.

1) Any recipient needs to be registered with the organization with his/her identity.

2) No separate key generation center is considered.

3) The organization himself/herself generates the parameters of the system and the secret key of the recipient.

4) We consider that a single data of the organization is stored in the cloud server.

Design goals: The design goals of the CBP scheme is summarized as follows. 1) Update the existing ReKey to avoid the regeneration of the whole ReKey.

2) Proxy server should not have any idea about the new users from the updated ReKey.

TABLE I: Analysis of existing B-RE schemes

Scheme	Unidirectional	Proxy server needs identities of recipients	Dynamic	Third party has the power to modify the existing group of recipients	Modify the existing group after taking care of utility of each entity.
Chu et al. [2]	Yes	Yes	No	Not relevant	Not relevant
Liang et al. [13]	Yes	No	No	Not relevant	Not relevant
Xu et al. [3]	Yes	No	No	Not relevant	Not relevant
Jiang and Guo [14]	Yes	No	Yes	Yes	No
Ge et al. [4]	Yes	No	Yes	Yes	No
CBP	Yes	No	Yes	No	Yes

3) There is a need to find the optimal number of new users who can join the existing group.

4) The existing user uses his/her same secret key to recover the plaintext the updated ReText.

5) If the proxy server and the intended recipients collide, they cannot find out the master key or the secret key of the organization. 6) The proxy server should not obtain the identities of the recipients.

IV. Preliminaries

We discuss some of the required preliminaries as follows.

1) Bilinear map:: Our proposed scheme is built from bilinear maps [31]. Let $\mathcal{G}_1$, and $\mathcal{G}_2$ be two cyclic groups of prime order p. A bilinear map e: $\mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ such that two generators $g_1, g_2 \in \mathcal{G}_1$ and $a, b \in \mathbb{Z}_p$

- e(${g_1}^a$,${g_2}^b$) = e(g_1,g_2)^{ab}, where $(g_1, g_2) \in \mathcal{G}_1$, and $(a, b) \in \mathbb{Z}_p$, and holds the following properties.
- If $u = e(g, g)$, then u is the generator of $\mathcal{G}_2$.

2) Discrete log problem: Let two elements $g_1, g_2 \in \mathcal{G}_1$ and $\mathcal{G}_1$ is a cyclic group of prime order p. Here, g_1 is the generator of the group $\mathcal{G}_1$ then the advantage of any polynomial time adversary to find x, such that $g_1^x = g_2$, is negligible [32].

V. The proposed framework

In this section, we define the proposed scheme, game formulation, and discuss the methodology of CBP.

A. Definition

The proposed scheme consists of nine algorithms. All of the algorithms are defined as follows.

1) *ParamGen:* The inputs of the *ParamGen* are security parameter ς and maximum possible number of recipient in one re-encryption $\mathcal{M}$. It outputs public parameter $\mathcal{PP}$ and master key $\mathcal{MK}$.
2) *SecKeyGen:* The inputs of the *SecKeyGen* are master key $\mathcal{MK}$ and the identity of user α_i. It gives secret key of user κ_i
3) *OriginalEnc:* The inputs of the *OriginalEnc* are $\mathcal{PP}$, identity of the recipient of original recipient α_i, and plaintext $\mathcal{P}$. It outputs original ciphertext $\mathcal{OC}$.
4) *OriginalDec:* The inputs of the *OriginalDec* are $\mathcal{PP}$, $\mathcal{OC}$, and the secret key of original ciphertext recipient κ_i. It gives plaintext $\mathcal{P}$ as output.
5) *ReEncKeyGen:* The inputs of the *ReEncKeyGen* are $\mathcal{PP}$, secret key of organization κ_o, and the identities of the recipients $\mathcal{R} = \{\alpha_1, \alpha_2, ..., \alpha_x\}$. It outputs the ReKey rky.
6) *ReCipherGen:* The inputs of the *ReCipherGen* are $\mathcal{PP}$, rky, and $\mathcal{OC}$. It outputs the ReText $\mathcal{RC}$.
7) *JoinDecision:* This algorithm takes $\mathcal{M}$, existing number of recipients in the group $\mathcal{R}$, which is x, and the number of new recipients want to join n as inputs. It outputs an optimal value of new recipient k, who can join the group $\mathcal{R}$.
8) *UpdateReKey:* This algorithm inputs the optimal value of new recipient k, the ReKey rky. It outputs the updated ReKey rky'. It adds the k members to the existing group $\mathcal{R}$. The proxy server inputs the updated ReKey rky' and $\mathcal{RC}$. It outputs the updated ReText $\mathcal{RC}'$.
9) *ReDec:* The inputs of the *ReDec* are initial ReText $\mathcal{RC}$ or updated ReText $\mathcal{RC}'$, the identity of the recipient α_i, and secret key κ_i. If $\alpha_i \in \mathcal{R}$, then the algorithm outputs $\mathcal{P}$, else it outputs $\perp$.

B. Game formulation

1) Justification of using coalitional game: Our objective is to decide whether new members can join the existing group of recipients or not. If we keep on adding the new member to the group of existing recipients, it would be beneficial for the organization as s/he does not require to generate separate ReKey for all the new recipients. However, the computation cost of the decryption increases with the count of recipients in the group. We aim to reduce the cost of the system in terms of size and computation cost. Therefore, we need to consider the utility function of both the organization and the recipient to find out the optimal number of new recipients who can join the group. The coalitional game is a game between a group of players where the goal of the players is to find the utilities. The players can get in a group though they are competitive in nature. In the scenario, our goal is to find the best scenario for the system in terms of size and computation cost, where the players are the organization and the recipients, and they are performing in a group. The problem to find the optimal number of recipients can be mapped to the Knapsack problem which is NP hard problem. Hence, to provide an incremental heuristic solution, coalition game is one of the promising methods to adopt the dynamics of the proposed system. Hence, we use

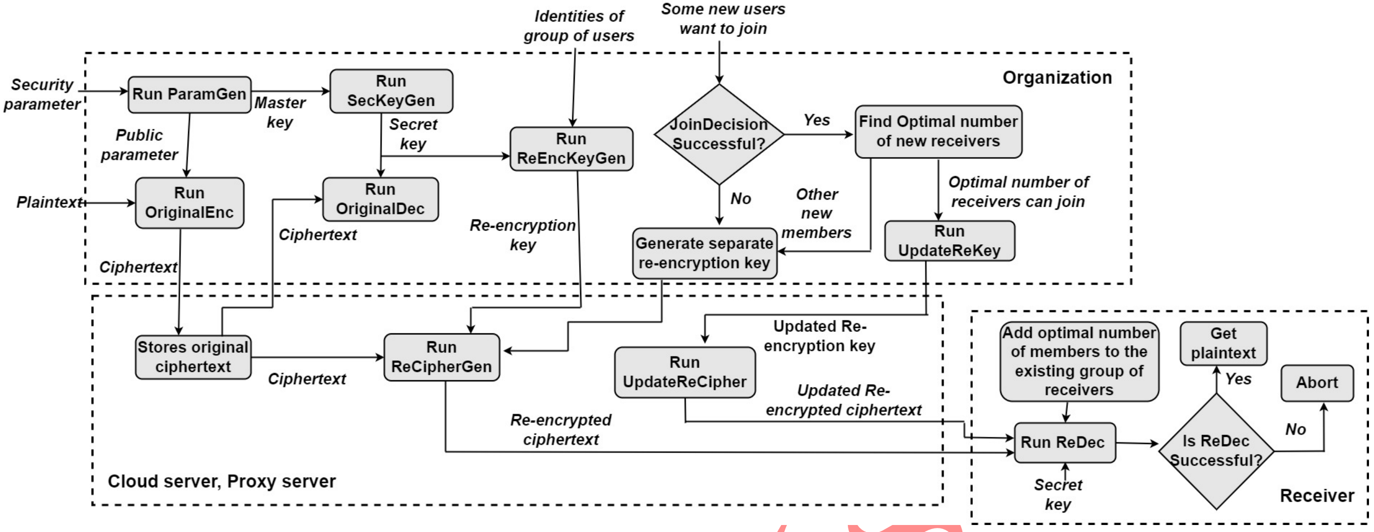

Fig. 3: Workflow of CBP

coalitional game theory to formulate the game and solve the problem of finding the optimal number of new recipients who can join the existing group.

2) Game model: We formulate the coalitional game as follows.

i) The organization and the recipient group are the players of the coalitional game.

ii) The utility functions of the organization and the recipients are U_O and U_R respectively. We consider that the ReKey is initially generated for x number of users. Later, another n number of users want to join the group. Let the maximum number of allowable recipients in a group is N. Therefore, the first condition to add n members in the group of x members is $(x + n) \leq N$. If the n number of receivers is added to the existing group, it is beneficial for the organization as the organization does not require to calculate a separate ReKey for each newly joined receiver. It also reduces the size of the required ReKey. However, it incurs the computation cost of the decryption of the ReText as it grows with the increase of the count of the recipients in the group.

3) Cost and size evaluation: We do not consider the proxy server as the player because it does not have much role in the updating process. The utility function of the organization considers the reduction of computation cost and size of ReKey in the update process than the re-generation of ReKey. The receiver's utility depends on how much extra computation needs to be done for the newly joined user. Therefore, we consider both computation cost and size in this scheme. Based on this, we make a decision, whether updating the ReKey is better or separate ReKey generation is better. The computation cost of the sender to generate a separate ReKey for n number of receivers is $\zeta_2 n$. If we add only k users from n users, then the ReKey generation cost for n new receivers is $\zeta_1 k + \zeta_2 (n-k)$. Here, $0 \leq k \leq n$. Here, ζ_1 and ζ_2 are the constants. ζ_1 denotes the computation cost to update the ReKey for a single recipient and ζ_2 denotes the computation cost to calculate the ReKey for a single recipient. Here, $\zeta_2 > \zeta_1$ as the calculation cost of ReKey is always greater than the update cost of ReKey. It should be noted that the computation cost depends on expensive mathematical operations like bilinear pairing and modular exponentiation. Similarly, if we generate a separate ReKey for all n new recipients, then the size of the security elements is $\zeta_3 n$. If we add k new users from n new users to the existing group, then the size is $\zeta_3(n-k) + \zeta_4$. Here, ζ_3 and ζ_4 are the constants. ζ_3 is the size of the ReKey for a single recipient and ζ_4 is the size of the security element, which is created for a single update of a ReKey. It should be noted that to update ReKey for k members, the size of security elements is constant. The size grows linearly with the count of recipients for $(n-k)$ members as a separate ReKey needs to be generated. On the other hand, if k members are added to the existing group of x members, then the cost of decryption for each member is $\eta_1 + \eta_2(x+n)$ as the count of members in the group is $x + n$. Here, η_1 and η_2 are constants, where η_1 is a constant amount of expensive operations for decryption, which does not depend on the number of recipients and η_2 is the number of expensive operations for decryption for a single recipient, which linearly increases with the increase of the count of recipients present in the group.

4) Utility function: The system's utility function depends on both the players' utility functions. Here, the utility function of the organization is inversely proportional to the utility function of the existing receiver. The utility function of the organization is denoted as $U_O(x, n, k)$ and the utility function of the receiver is denoted by $U_R(x, n, k)$. Here, The utility function of the organization $U_O(x, n, k)$ is directly proportional to k and the utility function of the receiver $U_R(x, n, k)$ is proportional to $-k$, where, $0 \leq k \leq n$. Hence, we can say $U_R(x, n, k)$ and k are negatively correlated to each other. Here, n is the number of users that want to join the existing group, where the initial number of members is x.

a) Utility function of the organization: The utility function of the organization depends on the cost of ReKey generation and the sizes of the ReKey. Therefore, we can write

$$U_O(x, n, k) = U_{O_{cost}}(x, n, k) + U_{O_{size}}(x, n, k) \qquad (1)$$

The cost-utility function of the organization is denoted as $U_{O_{cost}}(x, n, k)$. It is calculated based on how much cost is reduced to add n new users to the existing group. If n users are added to the existing group, it is efficient for the organization as the existing ReKey only needs to be updated. We calculate how much cost is reduced to add k members from n members and for other $(n-k)$ members, a separate ReKey is generated. Hence, the cost-utility function of the organization is calculated as follows.

$$U_{O_{cost}}(x, n, k) = \frac{(\zeta_2 n) - (\zeta_1 k + \zeta_2(n-k))}{\zeta_2 n} = \frac{(\zeta_2 - \zeta_1)k}{\zeta_2 n} \tag{2}$$

Similarly, the size utility of the organization is denoted as $U_{O_{size}}(x, n, k)$. If total n members are added to the existing recipient group, it increases the size of security elements as it does not need to generate the whole ReKey. We calculate how much size is reduced to add k members from n members and for other $(n-k)$ members, a separate ReKey is generated. Therefore, the size utility function of the organization is calculated as follows.

$$U_{O_{size}}(x, n, k) = \frac{(\zeta_3 n) - (\zeta_3(n-k) + \zeta_4)}{\zeta_3 n} = \frac{(\zeta_3 k) - \zeta_4}{\zeta_3 n} \tag{3}$$

The utility function of the organization can be written as a combination of the cost-utility function and the size-utility function of the organization. Here, the impact factors of the cost-utility and the size-utility are denoted as δ_c and δ_s respectively. Therefore, the utility function of the organization can be written as follows.

$$U_O(x, n, k) = (\delta_c U_{O_{cost}}) + (\delta_s U_{O_{size}}) = \left(\delta_c \frac{(\zeta_2 - \zeta_1)k}{\zeta_2 n}\right) + \left(\delta_s \frac{(\zeta_3 k) - \zeta_4}{\zeta_3 n}\right) \tag{4}$$

b) Utility function of the recipient: The utility function of the recipient is calculated based on the computation cost of the existing recipient to decrypt the ReText. The cost of decryption increases linearly with the increase in the number of members of the existing group. Therefore, if we add n members to the existing group of x users, the computation cost of decryption of the ReText is maximum. In this situation, the decryption cost is $\eta_1 + \eta_2(x+n)$ as the number of recipients present in the group is $x+n$. On the other hand, if only k number of recipients are added to the existing group, the decryption cost is $\eta_1 + \eta_2(x+k)$. The utility function of the recipient is calculated based on how much the computation cost of decryption of the ReText is reduced after adding k members to the group of x members. For the other $(n-k)$ users, a separate ReKey is generated. Hence, the recipient does not need to consider these users. The utility function of the recipient in the group is calculated as follows.

$$U_R(x, n, k) = \frac{(\eta_1 + \eta_2(x+n)) - (\eta_1 + \eta_2(x+k))}{(\eta_1 + \eta_2(x+n))} = \frac{\eta_2(n-k)}{(\eta_1 + \eta_2(x+n))} \tag{5}$$

c) System utility function: The system utility function is denoted as $U_{System}(x, n, k)$. It is a combination of the utility functions of the organization and the existing recipients. The k number of members is added to the existing group of x. Hence, the total decryption cost of all the $x+k$ members is considered in the system utility function. Therefore, we can write the system utility function as follows.

$$U_{System}(x, n, k) = \Delta_O\Big(U_O\Big) + \Delta_R\Big(U_R\Big)\Big(x + k\Big) \tag{6}$$

Here, Δ_O is the impact factor of the organization and Δ_R is the impact factor of the receiver. Therefore, we can write the total utility function of the system as follows.

$$U_{System}(x, n, k) = \Delta_O\left(\left(\delta_c \frac{(\zeta_2 - \zeta_1)k}{\zeta_2 n}\right) + \left(\delta_s \frac{(\zeta_3 k) - \zeta_4}{\zeta_3 n}\right)\right) + \Delta_R\left(\frac{\eta_2(n-k)(x+k)}{(\eta_1 + \eta_2(x+n))}\right) \tag{7}$$

5) Objective function: We calculate the utility function $U_{System}(x, n, k)$ for $0 \leq k \leq n$. The optimal value of k is calculated for which U_{System} is the maximum among all possible values. Therefore, we can write the objective function as follows.

$Max\{U_{System}(x, n, k), \ \forall 0 \leq k \leq n\}$. Here, initially, the ReKey is generated for the x number of users. Later n new members want to join the group. The function must satisfy the constraints as follows.

1) $x+n \leq N$, Here N is the maximum number of allowable receivers in the receiver group.
2) $0 \leq \delta_c, \delta_s \leq 1$.
3) $0 \leq \Delta_c, \Delta_s \leq 1$.
4) The n number of receivers who want to join the same group.

6) Equilibrium analysis: We select the strategy $\Big(k, n\Big)$ over $\Big(l, n\Big)$, if $U_{System}(x, n, k) \geq U_{System}(x, n, l)$. It is not possible to increase one player's utility without decreasing the other player's utility. The strategy $\Big(k^*, n\Big)$ is in nash equilibrium, if $U_{System}(x, n, k^*)$ is maximum among all $U_{System}(x, n, k)$, where $0 \leq k \leq n$.

C. Methodology

We use the coalitional game theory [26] in the B-RE [3] scheme. The concept of the B-RE scheme is borrowed from the schemes proposed in Refs. [3], [4]. However, in Refs. [3], [4], the initial ciphertext is shared with multiple recipients. The proposed scheme considers that the initial ciphertext is only shared with the identity of the organization. A separate key generation center is present in the schemes Refs. [3] and [4]. In our case, the organization provides services to the user. Hence, the users need to register with the organization at the time of key generation. After authentication of the identity of the recipient, it generates a secret key for the registered users. Additionally, we propose the decision-making process to find out the optimal number of members from the intended newly joined members, and we update the ReKey and ReText based on the resulting decision. The proposed scheme consists of nine algorithms namely *ParamGen*, *SecKeyGen*, *OriginalEnc*, *OriginalDec*, *ReEncKeyGen*, *ReCipherGen*, *JoinDecision*, *UpdateReKey*, and *ReDec*. Fig. 3 shows the workflow of the proposed scheme. The organization runs *ParamGen* algorithm to generate the public parameter and the master key. The public parameter is given to the other entities in the system. The organization runs *SecKeyGen* algorithm to register a user. The user, who wants to register himself/herself to the organization, sends his/her identity and the organization runs *SecKeyGen* algorithm to generate a secret key for the user. The organization runs *OriginalEnc* algorithm to compute the original ciphertext for himself/herself before storing it on the cloud server. Whenever, s/he requires the data, s/he downloads the data and runs *OriginalDec* algorithm to get the original data back. If the organization wants to share the data with a group of recipients, s/he runs *ReEncKeyGen* algorithm to calculate the ReKey and the key is sent to the proxy server. The proxy server runs *ReCipherGen* algorithm to calculate the ReText for the group of recipients. If later, some new members want to join the existing group of the recipient, the organization runs *JoinDecision* algorithm to find out the optimal number of recipients who can join the existing group. The new ReKey is generated for the other new members. After finding the optimal number of recipients, who can join, the organization runs *UpdateReKey* to generate the updated ReKey. The organization runs *ReEncKeyGen* algorithm to generate a ReKey for the remaining new recipients. The proxy server updates the existing ReText based on the updated ReKey and generates a new ReText for the remaining recipients using the newly calculated ReKey. The original ReText and the updated ReText both can be decrypted by the *ReDec* using the secret key of the recipient if the identity of the recipient is present in the group of recipients.

1) *ParamGen:* The inputs are security parameter ς and maximum possible number of recipient in one re-encryption $\mathcal{M}$. It computes a bilinear map $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$. Three random elements $g, y, z \in \mathcal{G}_1$ and a random element $\rho \in \mathbb{Z}_p$ are chosen. Two functions $\mathcal{F}_1 : \{0,1\}^* \to \mathbb{Z}_p$ and $\mathcal{F}_2 : \mathcal{G}_2 \to \mathcal{G}_1$ are defined. Here, $\mathcal{X}_1 = \{y, y^{\rho}, ..., y^{\rho^{\mathcal{M}}}\}$ and $\mathcal{X}_2 = \{z, z^{\rho}, ..., z^{\rho^{\mathcal{M}}}\}$. Here, $u = e(g, y)$ and $\chi = g^{\rho}$. *ParamGen* outputs the public parameter $\mathcal{PP} = \{e, \mathcal{G}_1, \mathcal{G}_2, u, \chi, \mathcal{X}_1, \mathcal{X}_2, \mathcal{F}_1, \mathcal{F}_2\}$ and the master key $\mathcal{MK} = \{g, \rho\}$.

2) *SecKeyGen:* The inputs are master key $\mathcal{MK}$, public parameter $\mathcal{PP}$, and identity α_i. It calculates $\kappa_i = g^{\frac{1}{\mathcal{F}_1(\alpha_i)+\rho}}$. It outputs secret key κ_i.

3) *OriginalEnc:* The inputs of *OriginalEnc* are plaintext $\mathcal{P}$, identity α_i, and public parameter $\mathcal{PP}$. It selects $\beta \in \mathbb{Z}_p$ and calculates $\mathcal{OC}_1 = \chi^{-\beta}$, $\mathcal{OC}_2 = y^{\beta(\rho+\mathcal{F}_1(\alpha_i))}$, $\mathcal{OC}_3 = u^{\beta}\mathcal{P}$, and $\mathcal{OC}_4 = z^{\beta\left(\frac{\rho+\mathcal{F}_1(\alpha_i)}{\mathcal{F}_1(\alpha_i)}\right)}$. It gives the original ciphertext $\mathcal{OC} = \{\mathcal{OC}_1, \mathcal{OC}_2, \mathcal{OC}_3, \mathcal{OC}_4\}$ as output.

4) *OriginalDec:* The inputs are $\mathcal{OC}$, $\mathcal{PP}$, α_i, and κ_i. It calculates $\mathcal{K} = e(\kappa_i, \mathcal{OC}_2)$. Then it calculates the plaintext as $\mathcal{P} = \frac{\mathcal{OC}_3}{\mathcal{K}}$.

5) *ReEncKeyGen:* On input of $\mathcal{PP}$, secret key of organization κ_o, and intended group of recipients $\mathcal{R} = \{\alpha_1, \alpha_2, ..., \alpha_x\}$, it selects $\gamma, \sigma \in \mathbb{Z}_p$ and calculates $rky_1 = \chi^{-\gamma}$, $rky_2 = y^{\gamma \prod_{\alpha_j \in \mathcal{R}} (\rho+\mathcal{F}_1(\alpha_i))}$, $rky_3 = \mathcal{F}_2(u^{\gamma})y^{\sigma}$, and $rky_4 = \kappa_o z^{\frac{\sigma}{\mathcal{F}_1(\alpha_o)}}$. It outputs the ReKey $rky = \{rky_1, rky_2, rky_3, rky_4\}$.

6) *ReCipherGen:* It inputs $\mathcal{OC}$ and rky. It calculates $\mathcal{RC}_1 = rky_1$, $\mathcal{RC}_2 = rky_2$, $\mathcal{RC}_3 = rky_3$, $\mathcal{RC}_4 = \mathcal{OC}_4$, and $\mathcal{RC}_5 = \mathcal{OC}_3 e(rky_4, \mathcal{OC}_2)^{-1}$. The resulted ReText is $\mathcal{RC} = \{\mathcal{RC}_1, \mathcal{RC}_2, \mathcal{RC}_3, \mathcal{RC}_4, \mathcal{RC}_5\}$.

7) *JoinDecision:* This algorithm takes $\mathcal{M}$, $x = |\mathcal{R}|$, the number of new recipients n, the constants of organization $(\zeta_1, \zeta_2, \zeta_3, \zeta_3)$, the constants of the receiver (η_1, η_2), the cost impact factor of the organization δ_c, the size impact factor of the organization δ_s, the impact factor of the organization Δ_O, and the impact factor of the recipient Δ_R as inputs. It checks whether $x + n \leq \mathcal{M}$ or not. If $x + n > \mathcal{M}$, then it aborts. It runs separate *ReEncKeyGen* for the n new members. Else, it calculates $U_{System}(x, n, k)$ for $0 \leq k \leq n$ as follows and selects the $k^* = k$, for which $U_{System}(x, n, k)$ is maximum among all possible values. $U_{System}(x, n, k) = \Delta_O\left(\left(\delta_c \frac{\left(\zeta_2-\zeta_1\right)k}{\zeta_2 n}\right)+\left(\delta_s \frac{\left(\zeta_3 k\right)-\zeta_4}{\zeta_3 n}\right)\right)+\Delta_R\left(\frac{\eta_2\left(n-k\right)\left(x+k\right)}{\left(\eta_1+\eta_2\left(x+n\right)\right)}\right)$. It outputs the optimal number of new recipients k^*. The organization runs *ReEncKeyGen* algorithm for $n - k^*$ new members to generate a separate ReKey. Therefore, if n number of new recipients want to join the existing group of x members, the *JoinDecision* algorithm calculates the optimal number of new recipients k^* from n, who can join the existing group of x elements. The organization runs *ReEncKeyGen* algorithm and generates separate ReKey for remaining $n - k^*$ members. Then, a separate ReText is generated using the newly calculated ReKey. The $n - k^*$ new members can use an individual secret key to decrypt the ReText.

8) *UpdateReKey:* This algorithm takes k^* new members of group $\mathcal{R}' = \{\alpha'_1, \alpha'_2, ..., \alpha'_{k^*}, \}$, existing ReKey $rky = \{rky_1, rky_2, rky_3, rky_4\}$, and master key $\mathcal{MK}$ as inputs. It generates updated ReKey as follows $rky'_2 = rky_2^{\left(\rho+\mathcal{F}_1(\alpha'_1)\right)\left(\rho+\mathcal{F}_1(\alpha'_2)\right)...\left(\rho+\mathcal{F}_1(\alpha'_{k^*})\right)}$. Finally, the updated ReKey is $rky = \{rky_1, rky'_2, rky_3, rky_4\}$. After getting the updated ReKey, the new ReText is calculated. Here, $\mathcal{RC}'_2 = rky'_2$. Hence, the new ReText is $\mathcal{RC}' = \{\mathcal{RC}_1, \mathcal{RC}'_2, \mathcal{RC}_3, \mathcal{RC}_4, \mathcal{RC}_5\}$. It updates the group $\mathcal{R} = \mathcal{R} \cup \mathcal{R}'$. The cardinality of the group is updated to $(x + k^*) = |\mathcal{R}|$

9) *ReDec:* *ReDec* takes initial ReText $\mathcal{RC}$ or updated ReText $\mathcal{RC}'$, the identity of the recipient α_i, and secret key κ_i as inputs. It calculates $\mathcal{K}' = \left(e(\mathcal{RC}_1, y^{\Phi(\alpha_i,\mathcal{R})})e(\kappa_i, \mathcal{RC}_2)\right)^{\frac{1}{\prod_{\alpha_j \in \mathcal{R}, \alpha_j \neq \alpha_i} \mathcal{F}_1(\alpha_j)}}$ if it is initial ReText, where $x = |\mathcal{R}|$.
If it is updated ReText, then it calculates $\mathcal{K}' = \left(e(\mathcal{RC}_1, y^{\Phi(\alpha_i,\mathcal{R})})e(\kappa_i, \mathcal{RC}'_2)\right)^{\frac{1}{\prod_{\alpha_j \in \mathcal{R}, \alpha_j \neq \alpha_i} \mathcal{F}_1(\alpha_j)}}$. Here, $(x + k^*) = |\mathcal{R}|$. The function $\Phi(\alpha_i, \mathcal{R})$ can be written as $\Phi(\alpha_i, \mathcal{R}) = \rho^{-1}\left(\prod_{\alpha_j \in \mathcal{R}, \alpha_j \neq \alpha_i} (\rho + \mathcal{F}_1(\alpha_j)) - \prod_{\alpha_j \in \mathcal{R}, \alpha_j \neq \alpha_i} (\mathcal{F}_1(\alpha_j))\right)$. It calculates $\mathcal{P}' = \mathcal{RC}_5 e\left(\frac{\mathcal{RC}_3}{\mathcal{F}_2(\mathcal{K}')}, \mathcal{RC}_4\right)$. If $\alpha_i \in \mathcal{R}$, then $\mathcal{P}' = \mathcal{P}$, else the algorithm aborts.

VI. Analysis

In this Section, motivated by the work proposed in Ref. [3], we analyze the proposed algorithm in terms of correctness and security analysis of the algorithm. Theorem 1 proves that the CBP scheme is secure against an inside attacker. Theorem 2 proves the correctness of CBP.

Theorem 1. *If there exists an insider attacker $\mathcal{A}$, who holds a secret key κ_i, where $\alpha_i \in \mathcal{R}$ and $\mathcal{R}$ is the existing group after adding the new members, then $\mathcal{A}$ can output organization's secret key κ_o with probability ε, then the attacker can solve discrete log problem with the same probability.*

Proof. Here, adversary $\mathcal{A}$ is any member from the recipient's group, who behaves maliciously. The adversary $\mathcal{A}$ can holds one secret key of any recipient κ_i, where $\alpha_i \in \mathcal{R}$. The ReText is $\mathcal{RC} = \{\mathcal{RC}_1, \mathcal{RC}_2, \mathcal{RC}_3, \mathcal{RC}_4, \mathcal{RC}_5\}$, where $\mathcal{RC}_1 = \chi^{-\gamma}$, $\mathcal{RC}_2 = y^{\gamma \prod_{\alpha_j \in \mathcal{R}} (\rho + \mathcal{F}_1(\alpha_i))}$, $\mathcal{RC}_3 = \mathcal{F}_2(u^\gamma) y^\sigma$, $\mathcal{RC}_4 = z^{\beta\left(\frac{\rho+\mathcal{F}_1(\alpha_i)}{\mathcal{F}_1(\alpha_i)}\right)}$, and $\mathcal{RC}_5 = u^\beta \mathcal{P} e(\kappa_o z^{\frac{\sigma}{\mathcal{F}_1(\alpha_o)}}, y^{\beta(\rho+\mathcal{F}_1(\alpha_i))})^{-1}$. At the time of decryption, s/he calculates $\mathcal{K}' = \left(e(\mathcal{RC}_1, y^{\Phi(\alpha_i, \mathcal{R})}) e(\kappa_i, \mathcal{RC}_2)\right)^{\frac{1}{\prod_{\alpha_j \in \mathcal{R}, \alpha_j \neq \alpha_i} \mathcal{F}_1(\alpha_j)}}$. Then s/he calculates $y^\sigma = \frac{\mathcal{RC}_3}{\mathcal{F}_2(\mathcal{K}')}$ After decrypting $\mathcal{RC}$ with the secret key κ_i, adversary $\mathcal{A}$ can easily find y^σ. To find the secret key of the organization κ_o, it is clear that adversary $\mathcal{A}$ has to know the value of σ. Adversary has y^σ and y, as y is a master public key parameter. Therefore, it is clear that to find out the secret key of the organization, adversary $\mathcal{A}$ has to break the discrete log problem. So we can say that the advantage of adversary $\mathcal{A}$ to break the proposed scheme is ε. Here, ε is negligible. Hence, we can say that for any adversary $\mathcal{A}$, the probability to break the proposed scheme under inside attack is negligible. □

Theorem 1 proves if any user belongs to the existing group, s/he cannot access the organization's secret key using his/her secret key, re-encrypted ciphertext, and public parameter. Therefore, any outside attacker, who has only access to the public parameter, and does not have any knowledge about the existing recipients, new recipients, or any secret key, cannot discover the organization's secret key. Moreover, if we talk about the re-encrypted ciphertext, if the user belongs to the group, s/he can decrypt the corresponding re-encrypted ciphertext as s/he has his/her secret key. However, for any outside attacker, it is impossible to decrypt any re-encrypted ciphertext as s/he is not a group member and does not have any secret key.

Theorem 2. *If the organization decides to add a new user $\alpha_i^{'}$ to the group of existing recipients $\mathcal{R} = \{\alpha_1, \alpha_2, ..., \alpha_x\}$, the existing user can recover plaintext $\mathcal{P}$ from the updated re-encrypted ciphertext.*

Proof. If new user $\alpha_i^{'}$ joins the group of existing recipients $\mathcal{R} = \{\alpha_1, \alpha_2, ..., \alpha_x\}$, then the existing group is $\mathcal{R} = \mathcal{R} \cup \alpha_i^{'}$. Let $\alpha_{x+1} = \alpha_i^{'}$ The updated ReText is $\mathcal{RC}^{'} = \{\mathcal{RC}_1, \mathcal{RC}_2^{'}, \mathcal{RC}_3, \mathcal{RC}_4, \mathcal{RC}_5\}$. Here, $\mathcal{RC}_1 = \chi^{-\gamma}$, $\mathcal{RC}_2^{'} = y^{\gamma \prod_{\alpha_j \in \mathcal{R}} (\rho + \mathcal{F}_1(\alpha_i))}$, (Here, $\mathcal{R}$ is the updated group). $\mathcal{RC}_3 = \mathcal{F}_2(u^\gamma) y^\sigma$, $\mathcal{RC}_4 = z^{\beta\left(\frac{\rho+\mathcal{F}_1(\alpha_i)}{\mathcal{F}_1(\alpha_i)}\right)}$, and $\mathcal{RC}_5 = u^\beta \mathcal{P} e(\kappa_o z^{\frac{\sigma}{\mathcal{F}_1(\alpha_o)}}, y^{\beta(\rho+\mathcal{F}_1(\alpha_i))})^{-1}$. It runs *ReDec* algorithm as follows. $\mathcal{K}' = \left(e(\mathcal{RC}_1, y^{\Phi(\alpha_i, \mathcal{R})}) e(\kappa_i, \mathcal{RC}_2^{'})\right)^{\frac{1}{\prod_{\alpha_j \in \mathcal{R}, \alpha_j \neq \alpha_i} \mathcal{F}_1(\alpha_j)}}$

$= \left(e(\chi^{-\gamma}, y^{\Phi(\alpha_i, \mathcal{R})}) e(\kappa_i, y^{\gamma \prod_{\alpha_j \in \mathcal{R}} (\rho + \mathcal{F}_1(\alpha_i))})\right)^{\frac{1}{\prod_{\alpha_j \in \mathcal{R}, \alpha_j \neq \alpha_i} \mathcal{F}_1(\alpha_j)}}$

$= \left(e(g^{\rho-\gamma}, y^{\Phi(\alpha_i, \mathcal{R})}) e(g^{\frac{1}{\mathcal{F}_1(\alpha_i)+\rho}}, y^{\gamma \prod_{\alpha_j \in \mathcal{R}} (\rho + \mathcal{F}_1(\alpha_i))})\right)^{\frac{1}{\prod_{\alpha_j \in \mathcal{R}, \alpha_j \neq \alpha_i} \mathcal{F}_1(\alpha_j)}}$

$= e(g, y)^\gamma = u^\gamma$.

Then it calculates $\mathcal{RC}_5 e\left(\frac{\mathcal{RC}_3}{\mathcal{F}_2(\mathcal{K}')}, \mathcal{RC}_4\right)$

$= u^\beta \mathcal{P} e(\kappa_o z^{\frac{\sigma}{\mathcal{F}_1(\alpha_o)}}, y^{\beta(\rho+\mathcal{F}_1(\alpha_i))})^{-1} e\left(\frac{\mathcal{F}_2\left(u^\gamma\right) y^\sigma}{\mathcal{F}_2(u^\gamma)}, z^{\beta\left(\frac{\rho+\mathcal{F}_1(\alpha_i)}{\mathcal{F}_1(\alpha_i)}\right)}\right)$

$= \mathcal{P}$. Hence, the existing recipient can recover the plaintext $\mathcal{P}$ from the updated ReText. □

TABLE II: Experimental setup and benchmarks

Hardware	Intel Core i3-10110U CPU@2.10GHz
Operating system	Ubuntu 16.04 LTS
Compiler	gcc-5.4.0
Virtual machine	Oracle VirtualBox 6.1
Program Library	pbc library (version:0.5.14) [33]
Benchmarks	Conditional B-RE, Revocable B-RE

TABLE III: Simulation parameter

Parameter	Value
Impact factors of the cost and size	$\delta_c = 0.5$, $\delta_s = 0.5$
Impact factors of the organization and recipient	$\Delta_O = 0.5$, $\Delta_R = 0.5$
Constants of the organization	$\zeta_1 = 1$, $\zeta_2 = 4$, $\zeta_3 = 4$, $\zeta_4 = 1$
Constants of the recipients	$\eta_1 = 2$, $\eta_2 = 4$
Number of initial recipients	50
Number of new recipients	100 to 350
Coalition size	20 to 110

VII. Performance Analysis

A. Experimental Setup and benchmarks

The experimental setup and the benchmarks are shown in TABLE II. We use the pbc library [33] to implement the proposed scheme and compare it with the existing schemes Conditional B-RE [3] and Revocable B-RE [4]. The Conditional B-RE scheme is a B-RE scheme to forward the email which is stored in the cloud server. This scheme does not support any mobility of the recipients. The Revocable B-RE scheme is used to revoke existing users. Revocable B-RE scheme is the extension of the Conditional B-RE scheme, but the revocation power is given to the proxy server, which is not fully trusted. The objective of our proposed B-RE method is to find the optimal number of new recipients who can join the existing group of recipients when a large number of new recipients want to join the group. Therefore, we compare the proposed scheme with these algorithms. The proposed scheme is similar to the conditional B-RE and revocable B-RE schemes. In the proposed scheme, we consider that the data owner has the access to the master key and we update the ReKey instead of re-generating it. Additionally, the proposed scheme uses coalitional game theory to find the number of members who can join the existing group.

B. Simulation parameters

TABLE III shows the parameters used in the simulation parameter. The cost and the size impact factors, impact factors of the organization and the recipients, which are used in the simulation are mentioned. We also mention the value of the constants, the number of initial recipients, and the number of new recipients, which are used in the simulation. The system's utility depends on the utility of the organization and the utility of the recipients. The utility of the organization depends on the cost utility and size utility. The impact factors of the cost and

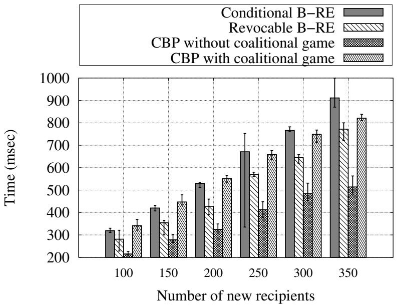

Fig. 4: Re-encryption key generation time

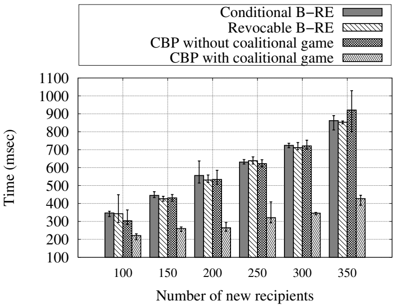

Fig. 5: Decryption time

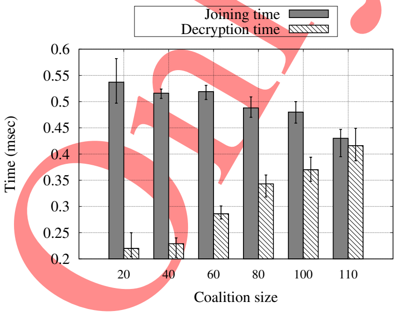

Fig. 6: Re-encryption key generation and decryption time vary with coalitional size

size are represented as δ_c and δ_s respectively. Here, $0 \leq \delta_c \leq 1$ and $0 \leq \delta_s \leq 1$. We have chosen $\delta_c = 0.5$ and $\delta_s = 0.5$. On the other hand, the impact factors of the organization and the recipients are represented as Δ_O and Δ_R respectively. Here, $0 \leq \Delta_O \leq 1$ and $0 \leq \Delta_R \leq 1$. We have chosen $\Delta_O = 0.5$ and $\Delta_R = 0.5$. The values of the constants of the organization and recipients depend on the number of expensive operations and the size of security elements, which are different in different proposed schemes. Referring to the scheme proposed by Xu *et al.* [3], the values of the constants of the organization and recipients ζ_1, ζ_2, ζ_3, ζ_4, η_1, and η_2 are taken as 1, 4, 4, 1, 2, and 4 respectively. The number of initial recipients is considered as 50 in Fig. 4 and Fig. 5. We have computed the time of ReKey generation and the time of the decryption in Fig.4 and Fig. 5 for 100 to 350 number of new recipients. In Fig. 6, the joining and decryption time are calculated for coalition size 20 to 110. In the experiment, we consider that 40% of the recipients decrypt the ReText as all the recipients may not perform the decryption operation because of the associated computation overhead. The coalition size is the number of recipients who can join the existing group of recipients.

C. *Performance metrics*

The performance metrics, which are used in the simulation are discussed as follows.

1) *Re-encryption key generation time:* The ReKey generation time is measured by the required time to generate/update the ReKey for the new recipients. In the proposed scheme if n number of recipients want to join and the optimal value is k, then for k number of recipients, it updates the ReKey and for $(n-k)$ number of recipients, it generates a new ReKey.

2) *Decryption time:* It is measured by the required time to decrypt the ReText for the existing users. It should be noted, we only consider the decryption time of a single recipient.

D. *Discussion*

Fig. 4 shows the comparisons of the ReKey generation time of the proposed scheme with existing schemes. The ReKey generation cost is almost similar in all the schemes when we use coalitional game theory to minimize the total cost of the system in CBP.

We update the existing ReKey for the optimal number of members and the other members, we calculate a separate ReKey. The Conditional B-RE and Revocable B-RE generate separate ReKeys for the newly joined user. The ReKey generation cost of CBP is less if the coalitional game theory is not used than the other existing schemes. Here, the ReKey needs to be updated for all the newly joined members. No separate ReKey is generated. Hence, the cost is reduced in this case.

Fig. 5 shows the comparisons of the decryption cost of the ReText. In the proposed scheme, if coalition game is used, the optimal number of new members is added to the existing group. On the other hand, a separate group is formed for the new recipients. Hence, the number of members present in the group considers the optimal number of members and the existing members. The other schemes consider all the newly joined members in a single group, Hence, the decryption cost is less in the proposed scheme, if the coalitional game is used than the other existing schemes. The decryption cost of the proposed scheme is more than the conditional B-RE and revocable B-RE if the coalitional game is not used. In that case, we keep on adding all the newly joined members to the existing group. It should be noted that the decryption cost depends on the number of members of the group. Here, in our case, the group consists of all the newly joined members and the existing members. For the other existing scheme, only the newly joined members are present in the group as a separate ReKey is calculated for new members.

Fig. 6 shows how the increase in coalition size affects the joining time and decryption time. The coalitional size indicates the number of members which are added to the group of existing recipients. In this figure, we want to show how the variation of the coalitional size affects the joining time and decryption time. It can be seen that the joining time decreases with the increase of the coalition size as the ReKey needs to be updated for the members equal to the coalition size. On the other hand, the decryption time increases with the increase in the size of the coalition as the number of members in the group increases.

VIII. Conclusion

In this paper, we proposed a B-RE for adding new recipients to the existing group of recipients. We formulated the utility functions of the organization and the recipient based on the current members of the group and new recipients, who want to join the group. We used the coalitional game theory to determine the optimal number of new recipients, whose addition would be beneficial for the system. The organization updated the ReKey for the optimal new recipients to reduce the cost of him/her. The existing recipients can decrypt the updated ReText using his/her secret key. We proved the correctness of the proposed scheme. Additionally, CBP is secure against an inside attacker. Finally, we implemented CBP to show the effectiveness of the scheme over other existing schemes. We showed that using the coalitional game reduces the decryption cost and the total cost of the system.

In the proposed scheme, we consider that the organization has its key generation center. Therefore, it has access to the master key. In the future, we can extend the work where a separate key generation center will be there to produce the secret key for the users. The work also can be extended to support the revocation of the existing recipients.

References

[1] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," *in Proc. Springer EUROCRYPT*, pp. 127–144, 1998.

[2] C. K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional Proxy Broadcast Re-Encryption," *in Proc. Springer ACISP 2009*, vol. 5594, pp. 327–342, 2009.

[3] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional Identity-Based Broadcast Proxy Re-Encryption and Its Application to Cloud Email," *IEEE Transactions on Computers*, vol. 65, no. 1, pp. 66–79, Jan 2016.

[4] G. Chunpeng, Z. Liu, J. Xia, and F. Liming, "Revocable Identity-Based Broadcast Proxy Re-encryption for Data Sharing in Clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, pp. 1214–1226, 2019.

[5] M. Jakobsson, "On Quorum Controlled Asymmetric Proxy Re-encryption," *in Proc. Springer PKC*, pp. 112–121, 1999.

[6] A. Ivan and Y. Dodis, "Proxy Cryptography Revisited," *in Proc. NDSS*, 2003.

[7] M. Green and G. Ateniese, "Identity-Based Proxy Re-Encryption," in *in Proc. Springer ACNS*, 2007, pp. 288–306.

[8] V. Kirtane and C. P. Rangan, "RSA-TBOS Signcryption with Proxy Re-encryption," in *Proc. the 8^{th} ACM workshop on Digital rights management*, ACM, Ed., 2008, pp. 59–66.

[9] G. Ateniese, K. Benson, and S. Hohenberger, "Key-Private Proxy Re-Encryption," in *in Proc. Springer CT-RSA*, vol. 5473, 2009, pp. 279–294.

[10] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," *in Proc. ACM CCS*, pp. 185–194, 2007.

[11] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys," *in Proc. Springer CRYPTO*, vol. 3621, pp. 258–275, 2005.

[12] C. Gentry and B. Waters, "Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts)," *in Proc. Springer EUROCRYPT*, vol. 5479, pp. 171–188, 2009.

[13] K. Liang, Q. Huang, R. Schlegel, D. S. Wong, and C. Tang, "A Conditional Proxy Broadcast Re-Encryption Scheme Supporting Timed-Release," *in Proc. Springer ISPEC*, vol. 7863, pp. 132–146, 2013.

[14] L. Jiang and D. Guo, "Dynamic Encrypted Data Sharing Scheme Based on Conditional Proxy Broadcast Re-Encryption for Cloud Storage," *IEEE Access*, vol. 5, pp. 13 336 – 13 345, 2017.

[15] Q. Huang, Y. Yang, and J. Fu, "PRECISE: Identity-based private data sharing with conditional proxy re-encryption in online social networks," *Future Generation Computer Systems, Elsevier*, vol. 86, p. 1523–1533, 2018.

[16] M. Sun, C. Ge, L. Fang, and J. Wang, "A proxy broadcast re-encryption for cloud data sharing," *Multimedia Tools and Applications, Springer*, vol. 77, no. 9, pp. 10 455—-10 469, 2018.

[17] Y. Liu, Y. Ren, C. Ge, J. Xia, and Q. Wang, "A CCA-secure multi-conditional proxy broadcast re-encryption scheme for cloud storage system," *Journal of Information Security and Applications*, vol. 47, pp. 125–131, 2019.

[18] S. Maiti and S. Misra, "P2B: Privacy Preserving Identity-Based Broadcast Proxy Re-Encryption," *IEEE Transactions on Vehicular Technology)*, 2020.

[19] S. Maiti and S. Misra, "GROSE: Optimal group size estimation for broadcast proxy re-encryption," *Computer Communications, Elsevier*, vol. 157, pp. 369–380, 2020.

[20] Y.-M. H. Han-Yu Lin, "An improved proxy re-encryption scheme for iot-based data outsourcing services in clouds," *Sensors, MDPI*, 2021.

[21] H. Zhong, Y. Zhou, Q. Zhang, Y. Xu, and J. Cui, "An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare," *Future Generation Computer Systems, Elsevier*, 2021.

[22] J. Cui, J. Lu, H. Zhong, Q. Zhang, C. Gu, and L. Liu, "Parallel key-insulated multiuser searchable encryption for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 18, 2022.

[23] J. Ma, T. Li, J. Cui, Z. Ying, and J. Cheng, "Attribute-based secure announcement sharing among vehicles using blockchain," *IEEE Internet of Things Journal*, vol. 8, 2021.

[24] H. Zhong, W. Zhu, Y. Xu, and J. Cui, "Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage," *Soft Computing, Springer*, 2022.

[25] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, "Coalitional game theory for communication networks," *IEEE Signal Processing Magazine*, vol. 26, pp. 77–97, 2009.

[26] A. Chakraborty, A. Mondal, and S. Misra, "Cache-Enabled Sensor-Cloud: The Economic Facet," *in Proc. IEEE WCNC*, 2018.

[27] S. Brahma and M. Chatterjee, "Spectrum sharing in secondary networks: A bargain theoretic approach," *in Proc. IEEE WCNC*, 2012.

[28] S. Brahma and M. Chatterjee, "Spectrum Bargaining: A Model for Competitive Sharing of Unlicensed Radio Spectrum," *IEEE Transactions on Cognitive Communications and Networking, 2015,*, pp. 257–272.

[29] L. Y. Njilla and N. Pissinou, "Dynamics of data delivery in mobile ad-hoc networks: A bargaining game approach," *in Proc. IEEE CISDA*, 2015.

[30] Y.-H. Lin, C.-Y. Wang, and W.-T. Chen, "A content privacy-preserving protocol for energy-efficient access to commercial online social networks," *in Proc. IEEE ICC*, 2014.

[31] C. . D. ee, "Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys," *in Proc. 13^{th} Int. Conf. Theory Appl. Cryptol. Inf. Security: Adv. Cryptol.*, pp. 200–215, 2007.

[32] I. F. Blake, X. Gao, R. C. Mullin, S. A. Vanstone, and T. Yaghoobian, "The discrete logarithm problem," *Menezes A.J. (eds) Applications of Finite Fields, the Springer International Series in Engineering and Computer Science (Communications and Information Theory)*, vol. 199, 1993.

[33] B. Lynn. (2013) Pbc library. [Online]. Available: https://crypto.stanford.edu/pbc/