ABP: Attribute-based Broadcast Proxy Re-encryption with Coalitional Game Theory

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

***Abstract*—This paper proposes Attribute-based Broadcast Proxy Re-encryption with Coalitional Game Theory (ABP) - a scheme to share encrypted cloud data with IoT devices. The scheme reduces the decryption costs on the receiver's side. In the case of sharing data with a subgroup of IoT devices, i.e., identities, where attributes satisfy a particular access policy, and the cardinality of the IoT devices satisfying the access policy is not known to the data owner, the existing encryption techniques such as broadcast proxy re-encryption and attribute-based encryption, cannot solve the problem. Hence, we propose the concept of attribute-based encryption in broadcast proxy re-encryption, named ABP. In ABP, we use the coalitional game theory to find the optimal coalition for minimizing the decryption cost and the total cost of the system. We prove that ABP is a selective security chosen-plaintext attack secure (SS-CPA) under the random oracle model. The performance of ABP is evaluated and compared with the existing broadcast proxy re-encryption schemes. We observe that the ABP reduces the decryption cost. Furthermore, the total cost of the system is reduced using the coalitional game.**

***Index Terms*—Identity-based encryption, Proxy re-encryption, Broadcast encryption, Ciphertext policy, Attribute-based encryption, Coalitional game theory.**

I. Introduction

Proxy re-encryption (PRE) [1]–[5] is profitable to share encrypted data stored in the cloud server [6]. In the presence of multiple IoT devices, the re-encryption key (Rkey) is created for each device, which increases the computational cost. To avoid this, the broadcast proxy re-encryption (BPRE) is proposed by Chu *et al.* [7], where a single Rkey is generated for multiple receivers. The sender must know all the identities in BPRE. On the other hand, attribute-based encryption (ABE) is another concept where encryption is done for an access policy. If the attribute list of a user follows the access policy, s/he can recover the plaintext. Hence, in the case of ABE, the sender has no idea about the receivers. In the presence of multiple identities, the receiver needs to consider all the identities at the decryption time. Hence, the cost of each decryption increases a lot as the decryption cost depends on the count of identities present. If the data owner generates a separate Rkey for each receiver, it violates broadcast proxy re-encryption benefits. Hence, finding the optimal coalition size from the total group is necessary to minimize the overall cost of the system. The BPRE scheme is used to share cloud data among multiple receivers when the identities of the receivers are known to the data owner. We use the ABE scheme to share the encrypted data with receivers, whose attribute lists satisfy a specific access policy. In the presence of a high number of users, if a sender wants to send the data to a subset of users in a group where the attributes satisfy an access policy, the aforementioned problem cannot be solved using either of the schemes, such as BPRE and ABE. It should be noted that the users are considered IoT devices, which are resource-constrained. Therefore, there is a need to use ABE with the BPRE scheme and the computation cost of the receivers should also be reduced.

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)

II. Related work

A. Broadcast Proxy Re-encryption

BPRE is introduced in Ref. [7] to solve the problem of re-creation of the key. When the data owner needs to circulate the cloud data with a set of identities, s/he calculates a Rkey using his/her private key and the set of identities. The derived key is delivered to the proxy server, which generates the Rciphertext and broadcasts it. Each receiver of the set of identities decrypts the Rciphertext. In this work, a specific condition is attached to both the ciphertext and the Rkey. If these two conditions match, the proxy server can re-encrypt the ciphertext. At the receiver side, if the receiver belongs to the set, s/he recovers the plaintext. For cloud email sharing, a chosen-plaintext attack secure conditional identity-based BPRE scheme is proposed in Ref. [12]. In Ref. [18], a dynamic conditional BPRE scheme is proposed. Another revocable BPRE scheme is proposed in Ref [13], where the proxy server does the revocation. A privacy-preserving BPRE scheme is proposed in Ref. [14], where one receiver cannot find the other group members. An optimal coalition size of the BPRE scheme is proposed in Ref. [19]. The proposed scheme balances the payoffs of the data owner and the receivers. A coalitional game-based BPRE scheme is proposed in Ref. [16] to add new receivers to the existing group. Recently, a multi-channel BPRE scheme is proposed in Ref. [20].

B. Attribute-based Proxy Re-encryption

Attribute-based PRE (APRE) is proposed based on the concept of ABE [21]–[23]. In ciphertext policy ABE (CABE) [21], a list of attributes is linked to a user, and the access policy is linked to the ciphertext. If any attribute list matches the access policy, the corresponding user recovers the plaintext. Using the concept of ABE, the initial ciphertext is re-encrypted

TABLE I: Comparisons of ABP with existing schemes

Work	Type	Optimal group size	Key Generation takes		Re-encryption key generation takes		Ciphertext decryption if		Decryption does not take input all the receivers
			Attribute list	Identity	Access policy	Group of identities	Identity must be present in a specific group	Attribute list follows a specific access policy	
Alrawais *et al.* [8]	Key exchange protocol based on CAB.	✗	✓	✗	✗	✗	✗	✓	✗
Wang *et al.* [9]	Leakage avoiding identity-based PRE.	✗	✗	✓	✗	Single identity	Single identity	✗	✗
Tu *et al.* [10]	PRE scheme is used with ciphertext policy ABE to update access policy.	✗	✓	✗	✓	✗	✗	✓	✗
Huang *et al.* [11]	Id-based conditional BPRE.	✗	✓	✓	✗	✓	✓	✗	✗
Xu *et al.* [12]	Id-based conditional BPRE.	✗	✗	✓	✗	✓	✓	✗	✗
Ge *et al.* [13]	Dynamic BPRE.	✗	✗	✓	✗	✓	✓	✗	✗
Maiti *et al.* [14]	Privacy-preserving BPRE to hide identities.	✗	✗	✓	✗	✓	✓	✗	✗
Ge *et al.* [15]	Revocable APRE to revoke users	✗	✓	✗	✓	✗	✗	✓	✗
Maiti *et al.* [16]	Coalitional-game-based BPRE to add new users.	✗	✗	✓	✗	✓	✓	✗	✗
Ge *et al. [17]*	Verifiable APRE to authenticate ciphertext	✗	✓	✗	✓	✗	✗	✓	✗
ABP	Attribute-based BPRE with coalitional game within group.	✓	✓	✓	✓	✓	✓	✓	✓

for access policy. Anonymous ciphertext policy attribute-based PRE scheme is proposed in Ref. [24], where the matching of the re-encryption key and the ciphertext is done before the re-encryption algorithm to protect the privacy of the attribute list and the access policy. A verifiable APRE is proposed in Ref. [17], where the re-encrypted ciphertext is verified by the receivers. A revocable APRE scheme is proposed in Ref. [15], where users are revoked from the initial set of decryption. There are some existing works on APRE and BPRE. TABLE I shows the comparisons of the ABP scheme with the existing schemes. When the data owner has a group of identities and s/he wants to circulate his/her data with a subgroup of that group, which satisfy a particular access policy, s/he does not have the identities of the receivers. The APRE or the BPRE scheme cannot solve the problem alone. We need to consider all the identities of the superset and particular access policy. Considering all the identities of the superset at the time of the Rkey generation, the cost of the system increases hugely. To reduce this, there is a need to find optimal coalition size before generating the Rkey.

C. Motivation

Fig. 1 shows the motivation scenario of ABP. Suppose, in an IoT application, a huge number of IoT devices for different purposes are registered. Initially, they are registered with their identities to identify the device uniquely. The admin of the application stores the encrypted data on the cloud server to maintain its confidentiality. Later, the data owner needs to share the data with the IoT devices, which serve a different purpose. If the data owner uses existing BPRE, each registered IoT device gets data as the Rkey is calculated for all users. On the other hand, if the data owner uses ABE, then any user who is not registered may also get the data, as ABE cannot specifically identify each user. If the user's specific attribute list satisfies the access policy, then only the user can get the data. Another problem is that some members may require

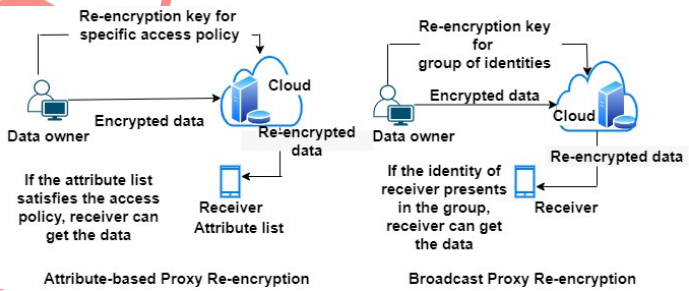

Fig. 1: Motivation Scenario

information about the group. Therefore, if we use BPRE along with ABE scheme, the decryption cost of receivers increases unnecessarily because of the huge number of receivers. Hence, there is a need to find an optimal coalition size to reduce the decryption cost of the receiver without violating the idea of broadcast encryption.

D. Contribution

In this work, we propose ABP by incorporating the advantages of ABE, broadcast encryption, PRE, and coalitional game theory. ABP generates Rkey and Rciphertext for a specific access policy and a group of identities while considering the system's cost. We consider the Rkey calculation cost, re-encryption cost, and decryption cost to evaluate the minimum cost of the system. We use coalitional game theory to obtain an optimal coalition size from the group of identities so that the decryption cost, as well as the overall cost of the system, are reduced. The contributions of ABP are summarized as follows:

1) The idea of attribute-based encryption is used in broadcast proxy re-encryption to share data to the subset of users of a set whose list of attributes satisfies a particular access policy.

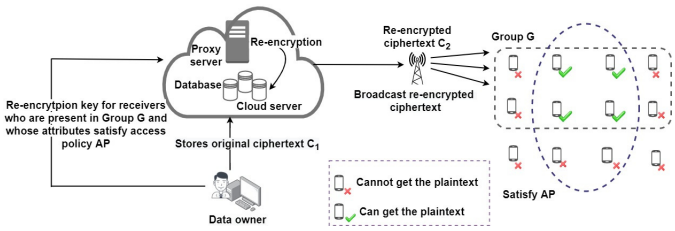

Fig. 2: System model

2) We define the utility functions of the data owner, proxy server, and receiver. We use coalitional game theory to reduce the total cost of the system

3) We model the game for SS-CPA of the ABP scheme and prove that the ABP is SS-CPA secure using the random oracle model [12].

4) We implement the ABP to evaluate the performance concerning different parameters.

III. Problem Statement

A. System Model

Fig. 2 shows the system model, which consists of three entities: data owner, cloud server, and a group of receivers. Let us consider that the data owner $\boldsymbol{id_A}$ encrypts plaintext $\boldsymbol{M}$ with identity $\boldsymbol{id_A}$ and stores the resulting ciphertext $\boldsymbol{C_1}$ at the cloud server. The data owner $\boldsymbol{id_A}$ has a large group of identities $\boldsymbol{G = \{id_1, id_2, ..., id_{n1}\}}$, but the data $\boldsymbol{M}$ needs to be shared with a subgroup $\boldsymbol{S \subseteq G}$, whose attributes satisfy a particular access policy $\boldsymbol{AP}$. The data owner has no idea about the attribute lists of the group $\boldsymbol{G}$. It is also impossible to find the specific identities of the group $\boldsymbol{S}$, whose attributes satisfy the specific access policy. Therefore, the Rkey needs to consider the access policy $\boldsymbol{AP}$ and the group $\boldsymbol{G}$. If any user with identity $\boldsymbol{id_x}$ and attribute list $\boldsymbol{att_x}$, where $\boldsymbol{id_x \notin G}$ or $\boldsymbol{att_x}$ does not satisfy $\boldsymbol{AP}$, cannot discover plaintext from the Rciphertext. We may think that identities can be used as attributes. If the identities can be used as the attributes, the size of the user's attribute list and the access policy increases with the number of identities. On the other hand, each user must know the identities of the other users as the identities must be present in the attribute list of the user in a -ve form, which is not possible in a practical scenario. Therefore, we cannot use the identities as attributes to solve the problem. On the other hand, as the size of the group $\boldsymbol{G}$ is huge, the decryption of the Rciphertext needs extra computations for the other identities of the set $\boldsymbol{G}$. Therefore, finding the optimal coalition size $\boldsymbol{m < |G|}$ is necessary, reducing the decryption cost of Rciphertext. If the sender $\boldsymbol{id_A}$ calculates the Rkey for every member of the group $\boldsymbol{G}$, the cost of Rkey generation and the cost of calculating Rciphertext increase significantly. Therefore, it is necessary to find the optimal coalition size $\boldsymbol{m}$ to reduce the sum of the costs of the system (the decryption cost, Rkey generation cost, and re-encryption cost).

a) System Operations: The specific system operations of different entities of the system model are as follows

Data Owner: Data owner encrypts plaintext M with his/her identity id_A and sends resulting ciphertext C_1 to the cloud server. In the future, if the data owner has to send the data with identities with a subgroup $S \in G$, where $G = \{id_1, id_2, ..., id_{n1}\}$ and the members also satisfy the access policy AP, s/he calculates the Rkey rk and sends to the proxy server.

Cloud Server: Cloud server keeps the original ciphertext C_1 with himself/herself. If s/he receives the Rkey rk, then s/he calculates Rciphertext C_2.

Receivers: If the receiver's identity $id_x \in S$ and his/her attributes att_x satisfy access policy AP, s/he can decrypt C_2.

B. Assumption

The assumptions of the problem statements are as follows:

- The total number of members of group $\boldsymbol{G}$ is fixed.
- The data owner $\boldsymbol{id_A}$ only knows about the identities of the group $\boldsymbol{G}$.
- The data owner has no idea about the corresponding attribute list of each member of the group $\boldsymbol{G}$.

C. Design Goals

1) The group of identities should be divided into coalitions of optimal size, which reduces the cost on the receiver side, as the receivers may be resource-constrained devices.

2) A receiver recovers the plaintext from the Rciphertext if his/her identity is present in a specific coalition and his/her attributes match the specific access policy.

3) For any outside attacker, it should be hard to decrypt the Rciphertext. 4) Any inside attacker cannot find the secret key of the sender.

D. Preliminaries

1) Bilinear Map: If $\mathbb{G}$ and $\mathbb{G}_\mathbb{T}$ be two cyclic groups of same prime order q, then e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\mathbb{T}$ is called a bilinear map [12], if two generators g_1, $g_2 \in \mathbb{G}$, a,b $\in \mathbb{Z}_q$, and it holds the following properties.

- e($g_1{}^a$,$g_2{}^b$) = e(g_1,g_2)^{ab}, where $(g_1, g_2) \in \mathbb{G}$, and $(a, b) \in \mathbb{Z}_q{}^*$.
- If $u = e(g_1, g_2)$, then u is the generator of $\mathbb{G}_T$.

We use Decisional Bilinear Diffie Hellman Assumption (DBDH) [25] and Discrete Logarithm Problem Assumption (DLP) [26] in security analysis of the proposed scheme

2) Defining attributes for attribute-based encryption: If the system consists of total $N = \{1, 2, \ldots, n\}$ attributes, then the public element $\{F_i\}_{1 \leqslant i \leqslant n}$ corresponds to attribute i, when attribute i is present in the attribute list. Similarly, $\{F_i\}_{n+1 \leqslant i \leqslant 2n}$ corresponds to attribute i, when i is not present in the attribute list and $\{F_i\}_{2n+1 \leqslant i \leqslant 3n}$ corresponds to attribute i where its presence or absence does not matter [27].

IV. ABP:The Proposed Scheme

A. Game formulation

1) Justification for choice of game formulation: We aim to divide the receivers into separate coalitions and find the optimal coalition size to minimize the cost of the system. The single Rkey is generated for a single coalition. Therefore,

the number of Rkeys increases with the number of coalitions present. Similarly, the number of re-encryption also depends on the number of coalitions present. On the other hand, each decryption of the Rciphertext depends on the size of each coalition. Therefore, if we only consider the cost of one side, it may reduce the cost of that side, but the cost of the other side increases, which leads to an increase in the overall cost of the system. Therefore, it needs to consider the cost of each entity to reduce the total cost. Therefore, we use the coalitional game theory to find out the optimal coalition size among all possible group sizes so that the total cost of the system is reduced.

2) Coalitional game model: We consider the problem of group formulation of receivers as a coalitional game [28]. The game consists of

Players: The data owner id_A, the proxy server, and the receiver group G act as the players of the game. It is to be noted that the data owner has the information about the receivers and acts as a coordinator. Hence, there is no need to exchange any information among the players.

Utility function: For each possible group size m, where $1 \leq m \leq |G|$, there is a utility function $U(m, G)$. The optimal coalition size is m, for which the utility $U(m, G)$ is maximum among all $U(m, G)$, $1 \leq m \leq |G|$. The utility function depends on the decryption cost, the Rkey generation cost, and the re-encryption cost. The optimal value of m ensures that the utilities of all the players increase without any bias.

3) Cost evaluation: The total group of receivers is $G = \{id_1, id_2, ..., id_N\}$. We have adopted the cost evaluation of the Rkey generation and decryption from [19]. In [19], the objective was to balance the Rkey and decryption costs. Therefore, non-cooperative game theory is used to solve the problem. Here, the objective is to minimize the overall cost of the system. If we keep all the N identities in one group, then cost of each decryption is written as $T_{dc}(N) = \beta_1 + \beta_2 \times N$ [19], where β_1 and β_2 are constants. If we keep single identity in a group, then each decryption cost is written as $T_{dc}(1) = \beta_1 + \beta_2$. If we keep m identities in one group, then cost of each decryption is written as $T_{dc}(m) = \beta_1 + \beta_2 \times m$ [19]. If $1 < m < N$, then $T_{dc}(1) < T_{dc}(m) < T_{dc}(N)$. On the other hand, if we keep N identities in a single group, then total Rkey generation cost is $TC_{rk}(N) = \alpha_1 + N \times \alpha_2$. Here, α_1, α_2 are constants. If we keep single element in one group, then total Rkey generation cost is $TC_{rk}(1) = N \times (\alpha_1 + \alpha_2)$ [19]. If we keep N identities in a single group, then the total re-encryption cost is constant θ. $T_{rc}(N) = \theta$. If the proxy server re-encrypts for every receiver, then $T_{rc}(1) = N * \theta$. Hence, when we keep m members in a group then total re-encryption cost is $T_{rc}(m) = \frac{N}{m} \times \theta$. Therefore, for $1 < m < N$, $T_{rk}(N) < T_{rk}(m) < T_{rk}(1)$ and $T_{rc}(N) < T_{rc}(m) < T_{rc}(1)$. Hence, we need to find an optimal coalition size m such that the total cost of the system is minimized.

B. Utility function

The total group of receivers $G = \{id_1, id_2, ..., id_N\}$ is divided into coalitions (each of size m, where $m < |G|$), such that the total cost of decryption, Rkey generation, and re-encryption is reduced than it's maximum value.

The utility function $U(m, G)$ depends on Rkey generation, re-encryption, and decryption costs.

a) Utility of re-encryption key generation: The cost of Rkey generation is $TC_{rk}(m) = \frac{N}{m} \times (\alpha_1 + \alpha_2 \times m)$, when the coalition size is m. Here $|G| = N$ and α_1, α_2 are constants. The Rkey generation cost becomes maximum when for every receiver, a single Rkey is generated. The maximum possible Rkey generation cost is $TC_{rk}(1) = N \times (\alpha_1 + \alpha_2)$. Therefore, the utility of Rkey generation is $U_{rk}(m) = \frac{TC_{rk}(1) - TC_{rk}(m)}{TC_{rk}(1)} = \frac{\alpha_1 - \frac{\alpha_1}{m}}{\alpha_1 + \alpha_2}$.

b) Utility of re-encryption: The cost of re-encryption is $TC_{rc}(m) = \frac{N}{m} \times \theta$, when group size is m. Here, θ is constant. The maximum possible re-encryption cost is $TC_{rc}(1) = N \times \theta$. Therefore, the utility of re-encryption is $U_{rc}(m) = \frac{TC_{rc}(1) - TC_{rc}(m)}{TC_{rc}(1)} = 1 - \frac{1}{m}$.

c) Utility of decryption: The cost of each decryption of Rciphertext is $T_{dc}(m) = \beta_1 + \beta_2 \times m$, when group size is m. Here, β_1 and β_2 are constants. The maximum possible decryption cost is $T_{dc}(N) = \beta_1 + \beta_2 \times N$. Therefore, the utility of decryption is $U_{dc}(m) = \frac{TC_{dc}(N) - TC_{dc}(m)}{TC_{dc}(N)} = \frac{\beta_1 + \beta_2 \times N - \beta_1 - \beta_2 \times m}{\beta_1 + \beta_2 \times N} = \frac{\beta_2 \times (N - m)}{\beta_1 + \beta_2 \times N}$.

d) Total utility: The total utility $U(m, G)$ depends on the utility of Rkey generation, re-encryption, and decryption of the Rciphertext. The data owner stores encrypted data on the cloud server. The data is downloaded and decrypted when s/he needs it. The data owner runs the Rkey generation algorithm to share the data with the receiver without performing tasks such as downloading, decrypting, and encrypting. Additionally, the Rkey is sent to the proxy server. The proxy server converts it to the Rciphertext using Rkey. The receiver does the decryption of the Rciphertext. In real life, the proxy server has more resources than the data owner or receiver. The receivers may be resource-constrained devices like mobile devices. Therefore, the impact of utilities of Rkey generation and decryption of Rciphertext should be more than the impact of the utility of re-encryption. The total utility when each group size is m is written as $U(m, G) = \delta_x \times \Big(U_{rk}(m) + U_{dc}(m)\Big) + \delta_y \times U_{rc}(m) = \delta_x \times \Big(\frac{\alpha_1 - \frac{\alpha_1}{m}}{\alpha_1 + \alpha_2} + \frac{\beta_2 \times (N - m)}{\beta_1 + \beta_2 \times N}\Big) + \delta_y \times \Big(1 - \frac{1}{m}\Big)$.

e) Objective function: The utility functions are calculated for all possible group sizes m. The optimal value of m is obtained for which the utility function $U(m, G)$ is the maximum among all possible values of m. Therefore, the objective function is written as

The objective function should satisfy the following constraints:

- $1 < m < |G|$.
- $\delta_x + \delta_y = 1$.
- $\delta_x > \delta_y$.

C. Equilibrium analysis

Given a set of identities $G = \{id_1, id_2, ..., id_N\}$, the Nash equilibrium is defined as $\{i^*, G\}$, where $U(i^*, G) \geq U(j, G)$, $\forall j, j \neq i^*$.

Lemma 1. *There exists a Nash equilibrium in ABP.*

Proof. We know – $\boldsymbol{U(i,G) = \delta_x \times \Big(U_{rk}(i) + U_{dc}(i)\Big) + \delta_y \times U_{rc}(i)}$ for strategy $\{i, G\}$. If $\boldsymbol{U(i,G) \geq U(j,G)}$ and $\boldsymbol{i \neq j}$, we argue that $\boldsymbol{i \succcurlyeq j}$. In ABP, a strategy $\{i, G\}$, where $\boldsymbol{i \succcurlyeq j}$, $\boldsymbol{1 \leq j \leq |G|}$ and $\boldsymbol{j \neq i}$, is considered to be the Nash equilibrium. We also argue that the utility of the game cannot be improved by any player by changing their decision in ABP. Hence, we argue that there exists Nash equilibrium. □

D. Definition of ABP and its security

1) Scheme Definition: The ABP scheme is composed of the following algorithms.

SU ($\boldsymbol{\lambda, n_1, n_2}$): It takes a security parameter $\boldsymbol{\lambda}$, number of possible receivers $\boldsymbol{n_1}$, and the number of attributes in universe $\boldsymbol{n_2}$ as inputs. It results in system public parameter $\boldsymbol{spp}$ and system secret key $\boldsymbol{ssk}$.

KG($\boldsymbol{id_A, att_A, ssk, spp}$): It takes identity $\boldsymbol{id_A}$, corresponding attribute list $\boldsymbol{att_A}$, system secret key $\boldsymbol{ssk}$, and system public parameter $\boldsymbol{spp}$ as inputs. It outputs secret key $\boldsymbol{sk_A}$.

ENC ($\boldsymbol{id_A, M, spp}$): It takes identity $\boldsymbol{id_A}$, plaintext $\boldsymbol{M}$, and $\boldsymbol{spp}$ as inputs and generates initial ciphertext $\boldsymbol{C_1}$.

GroupFormulation($\boldsymbol{G, \delta_x, \delta_y}$): It takes group of receivers $\boldsymbol{G}$ and impact factors $\boldsymbol{\delta_x}$, $\boldsymbol{\delta_y}$ and divides $\boldsymbol{N}$ into optimal coalition size $\boldsymbol{m}$. The number of coalitions is $\frac{N}{m}$, where $\boldsymbol{N = |G|}$.

RKG ($\boldsymbol{sk_A, m, AP, spp}$): It takes secret key $\boldsymbol{sk_A}$ of sender, $\boldsymbol{m}$ number of identities from group $\boldsymbol{G}$, access policy $\boldsymbol{AP}$, and system public parameter $\boldsymbol{spp}$ as inputs and generates Rkey $\boldsymbol{rk_{A->G,AP}}$. The Rkey needs to be generated for $\frac{N}{m}$ coalitions.

RENC ($\boldsymbol{C_1, rk_{A->G,AP}, spp}$): It takes initial ciphertext $\boldsymbol{C_1}$, Rkey $\boldsymbol{rk_{A->G,AP}}$, and system public parameter $\boldsymbol{spp}$ as inputs and it generates Rciphertext $\boldsymbol{C_2}$. The re-encryption needs to be computed for $\frac{N}{m}$ coalitions.

DC1 ($\boldsymbol{C_1, sk_A}$): It takes initial ciphertext $\boldsymbol{C_1}$ and secret key $\boldsymbol{sk_A}$ as inputs and recovers $\boldsymbol{M}$.

DC2 ($\boldsymbol{C_2, id_x, sk_x, G, att_x, AP, att}$): It takes $\boldsymbol{C_2}$, identity $\boldsymbol{id_x}$, corresponding secret key $\boldsymbol{sk_x}$, group of $\boldsymbol{m}$ users $\boldsymbol{G_m \in G}$, and attribute list $\boldsymbol{att_x}$ as inputs. If $\boldsymbol{id_x \in G_m}$ and $\boldsymbol{att_x}$ satisfies $\boldsymbol{AP}$, then it outputs plaintext $\boldsymbol{M}$; otherwise, it outputs $\perp$.

2) Re-encrypted ciphertext selective security chosen-plaintext attack model: ABP is said to be Rciphertext selective security chosen-plaintext attack (SS-CPA) secure if no Polynomial time adversary (PTA) has a non-negligible advantage in the security game. The initial ciphertext is obtained from identity-based encryption, which is not shared with any receiver. The Rciphertext is the resulting ciphertext of the attribute-based BPRE algorithm, which is shared with receivers. Hence, in this work, we show the security analysis of the Rciphertext only.

Init: The adversary $\boldsymbol{Ad}$ selects a set of receivers $\boldsymbol{G^*}$ and access policy $\boldsymbol{AP^*}$ and sends these to challenger $\boldsymbol{Ch}$.

Setup: The $\boldsymbol{Ch}$ executes SU and outputs $\boldsymbol{spp}$ and $\boldsymbol{ssk}$. $\boldsymbol{Ch}$ sends $\boldsymbol{ssp}$ to $\boldsymbol{Ad}$ and keeps $\boldsymbol{ssk}$.

Phase$_1$: $\boldsymbol{Ad}$ sends the following queries to $\boldsymbol{Ch}$ i) *Secret key query* $\boldsymbol{\mathbb{Q}_{sk}(id, att)}$*:* $\boldsymbol{Ad}$ sends query $\boldsymbol{\mathbb{Q}_{sk}(id, att)}$ to get the secret key. If $\boldsymbol{id \in G^*}$ and $\boldsymbol{att}$ satisfies $\boldsymbol{AP^*}$, then outputs $\perp$ else runs KG algorithm to get the secret key $\boldsymbol{sk_{id}}$. $\boldsymbol{Ch}$ responses this to $\boldsymbol{Ad}$.

ii)*Rekey query* $\boldsymbol{\mathbb{Q}_{rk}(id_A, att, G, AP)}$*:* $\boldsymbol{Ad}$ sends query $\boldsymbol{\mathbb{Q}_{rk}(id_A, att, G, AP)}$ to get Rkey. $\boldsymbol{Ad}$ cannot query both $\boldsymbol{\mathbb{Q}_{rk}(id_A, G, AP)}$ and $\boldsymbol{\mathbb{Q}_{sk}(id, att)}$ for any $\boldsymbol{G}$, where $\boldsymbol{id_A \in G^*}$, $\boldsymbol{id \in G}$ and $\boldsymbol{att}$ satisfies $\boldsymbol{AP}$.

Challenge: $\boldsymbol{Ad}$ chooses two plaintexts $\boldsymbol{M_0}$ and $\boldsymbol{M_1}$. For the initial ciphertext, $\boldsymbol{Ad}$ selects $\boldsymbol{id'}$ and $\boldsymbol{att'}$. It should be noted that the secret keys of $\boldsymbol{id'}$ and $\boldsymbol{att'}$ should not be queried. $\boldsymbol{Ad}$ sends $\boldsymbol{\{M_0, M_1\}}$ and $\boldsymbol{\{id', att'\}}$ to adversary $\boldsymbol{Ch}$. $\boldsymbol{Ch}$ randomly chooses $\boldsymbol{z \in \{0,1\}}$, and runs ENC $\boldsymbol{(id', att', M_z, spp)}$. The resulting initial ciphertext $\boldsymbol{C_1}$ is again re-encrypted to $\boldsymbol{C_2^*}$ for $\boldsymbol{G^*}$ and $\boldsymbol{AP^*}$. $\boldsymbol{Ch}$ sends $\boldsymbol{C_2^*}$ to adversary $\boldsymbol{Ad}$.

Phase$_2$: This is the same as Phase$_1$.

Guess: Adversary $\boldsymbol{Ad}$ guesses $\boldsymbol{z'}$. If $\boldsymbol{z' = z}$, it is said that $\boldsymbol{Ad}$ wins the game.

E. Methodology

We use the concept of attribute-based encryption from Ref. [27] and BPRE from Ref. [12]. Additionally, we use the coalitional game theory to find the optimal number of members in a coalition. The workflow of the ABP scheme is shown in Fig. 3. ABP comprises the algorithms — SU, KG, ENC, GroupFormulation, RKG, RENC, DC1, and DC2. The key generation center (KGC) runs SU to calculate the system public parameter and secret key. The data owner $\boldsymbol{id_A}$ runs the ENC algorithm to generate initial ciphertext $\boldsymbol{C_1}$ of data, which the DC1 algorithm can decrypt. KGC runs the KG algorithm to calculate the secret key of any user. When the data needs to be shared with a group of receivers $\boldsymbol{G}$, the GroupFormulation algorithm calculates the optimal coalition size $\boldsymbol{m}$ from group $\boldsymbol{G}$. The data owner calculates Rkeys by the RKG algorithm for each group $\boldsymbol{rk_{A->G,AP}}$ based on the identities of $\boldsymbol{G_m \in G}$ and access policy $\boldsymbol{AP}$. The third-party proxy server runs the RENC algorithm to calculate the Rciphertexts for all the groups. The DC2 algorithm can decrypt the Rciphertext $\boldsymbol{C_2}$ if the identity in the group $\boldsymbol{G_m}$ and the corresponding attribute list satisfy access policy $\boldsymbol{AP}$.

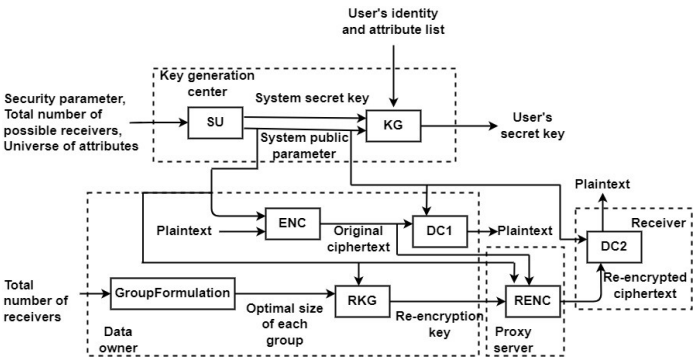

Fig. 3: Block diagram

1) **SU(λ, n_1, n_2):** It generates a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. Then, it selects α, β, $f_i (1 \leqslant i \leqslant 3n_2) \in \mathbb{Z}_p$, and three generators g, h, $u \in \mathbb{G}$. Then, it calculates $v_1 = e(g, h)$, $v_2 = {v_1}^{\beta}$, $w = g^{\alpha}$, and $F_i = g^{f_i}$ for each $(1 \leqslant i \leqslant 3n_2)$. Finally, it outputs system public parameter spp and system secret key ssk, where $spp = \langle e, h, h^{\alpha}, \ldots h^{\alpha^{n_1}}, u^{\alpha}, \ldots u^{\alpha^{n_1}}, \{F_i\}_{1 \leqslant i \leqslant 3n_2}, w, v_1, v_2, H, H^{'} \rangle$ and $ssk = \langle \alpha, \beta, g, \{f_i\}_{1 \leqslant i \leqslant 3n_2} \rangle$. Here, $H : \mathbb{G} \rightarrow \mathbb{Z}_p$, and $H^{'} : \mathbb{G}_T \rightarrow \mathbb{G}$.
2) **KG (id_A, att_A, ssk, spp):** This algorithm calculates $d_1 = g^{\frac{1}{\alpha + H(id_A)}}$. It randomly chooses $t_1, t_2, \ldots t_{n_2} \in \mathbb{Z}_p$ and calculates $t = t_1 + t_2 + \ldots + t_{n_2}$ and $d_2 = g^{\beta - t}$. For each $i \in n_2$, if $i \in att_A$, it calculates $d_{3,i} = h^{\frac{t_i}{f_i}}$ and $d_{4,i} = h^{\frac{t_i}{f_{2n_2+i}}}$. If $i \notin att_A$, then it calculates $d_{3,i} = h^{\frac{t_i}{f_{n_2+i}}}$ and $d_{4,i} = h^{\frac{t_i}{f_{2n_2+i}}}$. This algorithm outputs secret key $sk_A = \langle d_1, d_2, \{d_{3,i}, d_{4,i}\}_{i \in n_2} \rangle$.
3) **ENC (id_A, M, spp):** This algorithm randomly chooses $l \in \mathbb{Z}_p$. To encrypt data M, this algorithm calculates $e_1 = w^{-l}$, $e_2 = h^{l \cdot (\alpha + H(id_A))}$, $e_3 = {v_1}^{l} \cdot M$, and $e_4 = u^{l \cdot \frac{(\alpha + H(id_A))}{H(id_A)}}$. Finally, it outputs the initial ciphertext $C_1 = \langle e_1, e_2, e_3, e_4 \rangle$.
4) **GroupFormulation(G, δ_x, δ_y):** Let $N = |G|$. For each m, where $1 \leq m \leq N$ it calculates $U_{rk}(m) = \frac{\alpha_1 - \frac{\alpha_1}{m}}{\alpha_1 + \alpha_2}$. Then it calculates $U_{rc}(m) = \frac{\theta_1 - \frac{\theta_1}{m}}{\theta_1 + \theta_2}$ and $U_{dc}(m) = \frac{\beta_1 \times (1 - \frac{1}{m}) + \beta_2 (N-1)}{\beta_1 + \beta_2}$.
Finally, it calculates $U(m, G) = \delta_x \times \Big(U_{rk}(m) + U_{dc}(m)\Big) + \delta_y \times U_{rc}(m)$. It outputs optimal coalition size m, where $\max\{U(m, G), \forall\, 1 < m < N\}$.
5) **RKG (sk_A, m, AP, spp):** This algorithm randomly chooses s, s_1, $s_2 \in \mathbb{Z}_p$. It calculates $R = h^s$. For $i \in n_2$, if i is $+$ve in AP, then it calculates $R_i = {F_i}^s$. If i is $-$ve in AP, then it calculates $R_i = {F_{n_2+i}}^s$; Else it calculates $R_i = {F_{2n_2+i}}^s$. After that it calculates $rk_1 = w^{-s_1}$, $rk_2 = h^{s_1 \cdot \prod_{id_i \in G} (\alpha + H(id_i))}$, $rk_3 = d_1 \cdot u^{\frac{s_2}{H(id_A)}}$, and $rk_4 = H^{'}({v_2}^s \cdot {v_1}^{s_1}) \cdot h^{s_2}$. Finally, it outputs a Rkey $rk_{A->G,AP} = \langle rk_1, rk_2, rk_3, rk_4, (R_i)_{i \in n_2}, R \rangle$.
6) **RENC (C_1, $rk_{A->G,AP}$, spp):** This algorithm calculates $RE_1 = e_3 \cdot e(rk_3, e_2)^{-1}$, $RE_2 = rk_1$, $RE_3 = rk_2$, $RE_4 = rk_4$, $RE_5 = e_4$, and $RE_6 = rk_3$. It outputs the Rciphertext $C_2 = \langle RE_1, RE_2, RE_3, RE_4, RE_5, (R_i)_{i \in n_2} \rangle$.
7) **DC1 (C_1, sk_A):** This algorithm calculates $e(d_1, h^{l \cdot (\alpha + H(id_A))})$. It outputs plaintext M if C_1 is an encrypted ciphertext for id_A, else it outputs $\perp$.
8) **DC2 (C_2, id_x, sk_x, G, att_x, AP, att):** If $id_x \in G$, then s/he calculates $X_2 = \frac{1}{\big(e(RE_2, h^{\varnothing(id_x, G)}) \cdot e(d_1, RE_3)\big)^{\prod_{id_i \in G, id_i \neq id_x} H(id_i)}}$. Here $\varnothing(id_x, G) = \alpha^{-1} \cdot \Big(\prod_{id_i \in G, id_i \neq id_x} (\alpha + H(id_i)) - \prod_{id_i \in G, id_i \neq id_x} H(id_i)\Big)$.
For each $i \in n_2$, if $i \in att_x$, then receiver calculates $H_i = e(R_i, d_{3,i})$. If $i \notin att_x$, then it calculates $H_i = e(R_i, d_{3,i})$; otherwise, $H_i = e(R_i, d_{4,i})$. Then, it calculates $X_1 = \prod_{id_i \in N_2} H_i \cdot e(d_2, R)$. Finally, the receiver calculates $RE_1 \cdot e\Big(\frac{RE_4}{H^{'}(X_1 \cdot X_2)}, RE_6\Big)$.
If $id_x \in G$ and att_x satisfies AP, then it outputs plaintext M; otherwise, it gives $\perp$ as output.

Lemma 2. *The coalition formation optimizes the decryption cost of recipients.*

Proof. The GroupFormulation algorithm takes a group of receivers G and calculates optimal coalition size m, where $m \leq |G|$. The utility of decryption $U_{dc}(m)$ is negatively correlated with m, where $U_{dc}(m) = \frac{\beta_2 \times (N-m)}{\beta_1 + \beta_2 \times N}$ and $N = |G|$. Hence, if $m_1 \leq m_2$, then $U_{dc}(m_1) \geq U_{dc}(m_2)$. In ABP, the objective function needs to satisfy the constraint — $1 \leq m \leq |G|$. Therefore, we can write $U_{dc}(m) \geq U_{dc}(|G|)$. Hence, it proves that the coalition formation optimizes the decryption cost. □

F. Correctness

The following theorems prove the correctness of the proposed scheme. Theorem 1 proves the correctness of the initial ciphertext, which is adapted from [12] and [13], where the idea of Theorem 2 is modified from [27] and [12].

Theorem 3. *If the initial cipheretxt $C_1 = \langle e_1, e_2, e_3, e_4 \rangle$ is the result of ENC (id_A, M, spp) and the secret key of user id_A is $sk_A = \langle d_1, d_2, \{d_{3,i}, d_{4,i}\}_{i \in n_2} \rangle$, then DC1 ($C_1$, sk_A) always give the correct plaintext M.*

Proof. To decrypt $C_1 = \langle e_1, e_2, e_3, e_4 \rangle$, the user calculates $e(d_1, h^{l \cdot (\alpha + H(id_A))}) = e(g^{\frac{1}{\alpha + H(id_A)}}, h^{l \cdot (\alpha + H(id_A))}) = {v_1}^{l}$. Here, $v_1 = e(g, h)$. The user calculates $\frac{e_3}{v^l} = \frac{v^l \cdot M}{v^l} = M$. □

Theorem 4. *If the Rciphertext $C_2 = \langle RE_1, RE_2, RE_3, RE_4, RE_5, (R_i)_{i \in n_2} \rangle$ is the result of RENC (C_1, $rk_{A->G,AP}$, spp), the initial ciphertext C_1 is the result of ENC (id_A, M, spp), and the Rkey $rk_{A->G,AP} = \langle rk_1, rk_2, rk_3, rk_4, (R_i)_{i \in n_2}, R \rangle$ is the result of RKG (sk_A, G, AP, spp), then the DC2 (C_2, sk_x, G, att_x, AP) algorithm always gives the correct plaintext M.*

Proof. To decrypt the Rciphertext $C_2 = \langle RE_1, RE_2, RE_3, RE_4, RE_5, (R_i)_{i \in n_2} \rangle$, user

calculates $H_i = e(R_i, d_{3,i})$, where $sk_x = \langle d_1, d_2, \{d_{3,i}, d_{4,i}\}_{i\in n_2}\rangle$. If $i \in att_x$, then $H_i = e(R_i, d_{3,i}) = e(F_i{}^s, h^{\frac{t_i}{f_i}}) = e(g^{f_i\cdot s}, h^{\frac{t_i}{f_i}}) = e(g,h)^{s\cdot t_i}$. If $i \notin att_x$, then $H_i = e(R_i, d_{3,i}) = e(F_{n_2+i}{}^s, h^{\frac{t_i}{f_{n_2+i}}}) = e(g^{f_{n_2+i}\cdot s}, h^{\frac{t_i}{f_{n_2+i}}}) = v_1{}^{s\cdot t_i}$. Otherwise, $H_i = e(R_i, d_{4,i}) = e(F_{2n_2+i}{}^s, h^{\frac{t_i}{f_{2n_2+i}}}) = e(g^{f_{2n_2+i}\cdot s}, h^{\frac{t_i}{f_{2n_2+i}}}) = v_1{}^{s\cdot t_i}$. Then, it calculates $X_1 = \prod_{id_i\in n_2} H_i \cdot e(d_2, R) = e(g,h)^{s\cdot t} \cdot e(g^{\beta-t}, h^s)$ and $X_2 = e(RE_2, h^{\varnothing(id_x,G)}) \cdot e(d_1, RE_3)^{\frac{1}{\prod_{id_i\in G, id_i\neq id_x} H(id_i)}} = e(w^{-s_1}, h^{\varnothing(id_x,G)}) \cdot e(d_1, rk_2)^{\frac{1}{\prod_{id_i\in G, id_i\neq id_x} H(id_i)}}$ Then, the user calculates $RE_1 \cdot e\Big(\frac{RE_4}{H'(X_1\cdot X_2)}, RE_6\Big) = v_1{}^l \cdot M \cdot e(rk_3, e_2)^{-1} \cdot e\Big(\frac{H'(v_2{}^s \cdot v_1{}^{s_1})\cdot h^{s_2}}{H'(X_1\cdot X_2)}, d_1 \cdot u^{\frac{s_2}{H(id_A)}}\Big) = M$. Therefore, it is proved that if $id_x \in G$ and att_x satisfies AP, it outputs the correct plaintext M. □

V. Security Analysis

Theorem 5. *The re-encrypted ciphertext of the ABP scheme is SS-CPA secure.*

Proof. ABP scheme is considered ciphertext selective security chosen-plaintext attack (SS-CPA) secure if no PTA has a non-negligible advantage in the security game.

Init: Ad chooses a group of users G^* and access policy AP^* and sends G^* and AP^* to challenger Ch.

Setup: Ch runs the SU algorithm. Ch takes inputs λ, $N_1 = \{1, 2, \ldots, n_1\}$, and $N_2 = \{1, 2, \ldots, n_2\}$. It generates a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Here, α, β, $f_i (1 \leqslant i \leqslant 3n_2) \in \mathbb{Z}_p$ and $g, h \in \mathbb{G}$. It calculates $y \in \mathbb{Z}_p$. It calculates $u = h^y$ and $v_1 = e(g,h)$, $v_2 = v_1{}^\beta$, $w = g^\alpha$, $F_i = g^{f_i}$ for each $(1 \leqslant i \leqslant 3n_2)$. Finally, Ch generates the system public parameter $spp = \langle e, h, h^\alpha, \ldots h^{\alpha^{n_1}}, u^\alpha, \ldots u^{\alpha^{n_1}}, \{F_i\}_{1\leqslant i\leqslant 3n_2}, w, v_1, v_2, H, H'\rangle$ and system secret key $ssk = \langle \alpha, \beta, g, \{f_i\}_{1\leqslant i\leqslant 3n_2}\rangle$. Here, $u = h^y$, $H : \mathbb{G} \to \mathbb{Z}_p$, and $H' : \mathbb{G}_T \to \mathbb{G}$. Ch sends spp to Ad and keeps ssk.

Phase$_1$: Ad can issue the following queries to Ch

a) *Secret key query* $\mathbb{Q}_{sk}(id_x, att_x)$*:* Ad can query $\mathbb{Q}_{sk}(id, att)$ to get the secret key. If $id_x \in G^*$ or att_x satisfies AP^*, then Ch outputs $\perp$; otherwise, it runs KG algorithm to calculate the secret key $sk_x = \langle d_1, d_2, \{d_{3,i}, d_{4,i}\}_{i\in n_2}\rangle$. Challenger Ch sends sk_x to Ad and stores the tuple $\langle id_x, att_x, sk_x\rangle$ into table T_{sk}.

b) *Rekey query* $\mathbb{Q}_{rk}(id_x, att_x, G, AP)$*:* Ad can query $\mathbb{Q}_{rk}(id_x, att_x, G, AP)$ to get Rkey.

i) If $id_x \notin G^*$ and at least one tuple is present in table T_{sk} for group G and access policy AP, then Ch randomly chooses $s, s_1, s_2 \in \mathbb{Z}_p$. Then, it calculates $R = h^s$. Ch calculates $\mathbb{Q}_{sk}(id_x, att_x)$, where $sk_x = \langle d_1, d_2, \{d_{3,i}, d_{4,i}\}_{i\in n_2}\rangle$. For $i \in n_2$, if i is $+$ve in AP, then it calculates $R_i = F_i{}^s$. If i is $-$ve in AP, then it calculates $R_i = F_{n_2+i}{}^s$, otherwise $R_i = F_{2n_2+i}{}^s$. After that it calculates $rk_1 = w^{-s_1}$, $rk_2 = h^{s_1\cdot\prod_{id_i\in G}(\alpha+H(id_i))}$, $rk_3 = d_1 \cdot u^{\frac{s_2}{H(id_A)}}$, and $rk_4 = H'(v_2{}^s \cdot v_1{}^{s_1}) \cdot h^{s_2}$. Challenger Ch sends $rk_{x->G,AP} = \langle rk_1, rk_2, rk_3, rk_4, (R_i)_{i\in n_2}, R\rangle$ to Ad.

ii) If $id_x \in G^*$ and at least one tuple is present in table T_{sk} for group G and access policy AP, Ch responds with $\perp$.

iii) If $id_x \in G^*$ and no tuples are present in table T_{sk} for group G and access policy AP, then Ch randomly chooses $s, s_1, s_2 \in \mathbb{Z}_p$. For $i \in N_2$, if i is $+$ve in AP, then it calculates $R_i = F_i{}^s$. If i is $-$ve in AP, then it calculates $R_i = F_{n_2+i}{}^s$; otherwise, $R_i = F_{2n_2+i}{}^s$. Thereafter, it calculates $rk_1 = w^{-s_1}$, $rk_2 = h^{s_1\cdot\prod_{id_i\in G}(\alpha+H(id_i))}$. Ch randomly chooses $rk_3 \in \mathbb{G}$ and $rk_4 = H'(v_2{}^s \cdot v_1{}^{s_1}) \cdot h^{s_2}$. Challenger Ch sends $rk_{x->G,AP} = \langle rk_1, rk_2, rk_3, rk_4, (R_i)_{i\in n_2}, R\rangle$ to Ad.

Challenger Ch stores the tuple $\langle id_x, n_2, G, AP, rk_{x->G,AP}\rangle$ to table T_{rk}.

Challenge: Ad chooses two plaintexts M_0 and M_1. For the initial ciphertext Ad selects id' and att'. It should be noted that the secret keys of id' and att' should not be queried. Ad sends $\{M_0, M_1\}$ and $\{id', att'\}$ to adversary Ch. Ch randomly chooses $z \in \{0,1\}$ and runs EC (id', att', M_z, spp). Challenger Ch randomly chooses $l \in \mathbb{Z}_p$ and then it calculates $e_1 = w^{-l}$, $e_2 = h^{l\cdot(\alpha+H(id'))}$, $e_3 = v_1{}^l \cdot M$, and $e_4 = u^{l\cdot\frac{(\alpha+H(id'))}{H(id')}}$. The resulted initial ciphertext $C_1 = \langle e_1, e_2, e_3, e_4\rangle$ is re-encrypted for G^* and AP^*. Ch executes $\mathbb{Q}_{rk}(id', att', G^*, AP^*)$ and the resulted Rkey $rk_{id'->G^*,AP^*}$ is used to re-encrypt C_1 to C_2^*. The result C_2^* is sent to Ad.

Phase$_2$: This is same as Phase$_1$.

Guess: Adversary Ad guesses z'. If $z' = z$, we can say that Ad wins the game.

To guess the plaintext correctly, there are two possibilities for an adversary when in Rkey query $\mathbb{Q}_{rk}(id_x, att_x, G, AP)$, $id_x \in G^*$ and no tuples are present in table T_{sk} for group G and access policy AP.

1) Ad has to guess whether $rk_3 = g^{\frac{1}{\alpha+H(id_x)}} \cdot u^{\frac{s_2}{H(id_x)}}$ or not. Based on the DBDH assumption [25], the possibility of Ad to guess whether rk_3 is equal to $g^{\frac{1}{\alpha+H(id_x)}} \cdot u^{\frac{s_2}{H(id_x)}}$ is negligible.
2) To break the security, Ad must has to calculate s from the value of h and h^s. Based on the DLP assumption [26], the advantage of Ad to calculate the value of s is negligible.

TABLE II: Experimental Setup

Hardware	Intel Core i3-10110U CPU@2.10GHz
OS	Ubuntu 16.04 LTS
Compiler	gcc-5.4.0
Program Library	pbc-0.5.14 [29]

TABLE III: Simulation Parameters

Parameter	Value
Number of attributes (n_2)	10, 20, 30, 40, 50
Number of receivers (n_1)	50, 100, 150, 200
Impact factor δ_x	0.8
Impact factor δ_y	0.2
α_1	$n_2 + 1$
α_2	1
θ	1
β_1	$n_2 + 1$
β_2	2

Therefore, it is proved that the re-encrypted ciphertext of the ABP scheme is RC-SS-CPA secure □

VI. Performance Evaluation

A. Experimental Setup

The experimental setup, including hardware, Operating system, Compiler, and Program library, is shown in Table II. We implement the proposed scheme, ABP, to show its performance.

B. Simulation parameter

TABLE III shows the simulation parameter. We analyze the performance by varying the number of receivers from $\mathbf{50}$ to $\mathbf{200}$. We also vary the number of attributes from $\mathbf{6}$ to $\mathbf{50}$. The impact factors δ_x and δ_y are chosen as $\mathbf{0.8}$ and $\mathbf{0.2}$ respectively. The chosen constants α_1, α_2, θ, β_1, and β_2 are shown in TABLE III. We show the effects of these parameters on the performance of ABP. We compare the performance of ABP with some recent existing BPRE schemes —CIBPRE [12], RIB-BPRE19 [13], P2B20 [14], and CBP23 [16] schemes. Our proposed scheme is similar to BPRE schemes in most of the algorithms. The ENC algorithm generated identity-based encrypted ciphertext in both existing BPRE schemes and ABP schemes. On the other hand, in attribute-based proxy re-encryption, the initial ciphertext is attribute-based encrypted ciphertext. In the case of RKG and RENC algorithms, the existing BPRE converts the identity-based encrypted ciphertext to identity-based encrypted ciphertext for multiple users. In our scheme, the RKG algorithm converts the identity-based encrypted ciphertext to identity-based encrypted ciphertext, which also supports a specific access policy. On the other hand, the APRE converts the attribute-based encrypted ciphertext to ciphertext which supports a specific access policy. Moreover, the BPRE schemes and proposed scheme can specifically identify each user, which is not possible in the case of APRE. Therefore, it is clear that our proposed scheme is very similar to broadcast proxy re-encryption schemes, and there is not enough similarity between our scheme and attribute-based encryption schemes. Hence, we cannot compare ABP with existing attribute-based proxy re-encryption schemes.

TABLE IV: Required time for GroupFormulation

Total number of receivers	50	100	200	300	400	500
Time (microsecond)	1	1	2	2	4	4

C. Performance Metrics

In the section, we use the following performance metrics to show the performance of ABP.

a) Communication Overhead of the data owner to cloud server: The communication overhead of the data owner to cloud server is calculated as the size of initial ciphertext $\boldsymbol{C_1}$ and Rkey $\boldsymbol{rk_{A->G,AP}}$. The single initial ciphertext $\boldsymbol{C_1}$ contains four group elements $\langle \boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3}, \boldsymbol{e_4} \rangle$ and the single Rkey $\boldsymbol{rk_{A->G,AP}} = \langle \boldsymbol{rk_1}, \boldsymbol{rk_2}, \boldsymbol{rk_3}, \boldsymbol{rk_4}, (\boldsymbol{R_i})_{i \in att}, \boldsymbol{R} \rangle$. Depending on the number of elements, the number of Rkeys varies. The size of initial ciphertext $\boldsymbol{C_1}$ and the total size of Rkey $\boldsymbol{rk_{A->G,AP}}$ are evaluated as the communication overhead (bytes) from the data owner to the cloud server.

b) Communication Overhead of cloud server to receiver: We evaluate the communication overhead of cloud server to the receiver as the size of Rciphertext $\boldsymbol{C_2}$, where $\boldsymbol{C_2} = \langle \boldsymbol{RE_1}, \boldsymbol{RE_2}, \boldsymbol{RE_3}, \boldsymbol{RE_4}, \boldsymbol{RE_5}, (\boldsymbol{R_i})_{i \in att} \rangle$. The size of $\boldsymbol{C_2}$ is evaluated as the communication overhead (bytes) from the cloud server to the receiver.

c) Time of re-encryption key generation: We evaluate the required time to generate all Rkeys with varying the count of receivers and the count of attributes. To generate a single Rkey, the data owner needs to run a single RKG algorithm.

d) Time of decryption of re-encrypted ciphertext: We evaluate the required time to decrypt a Rciphertext with varying the count of receivers and attributes. To decrypt the Rciphertext, the receiver has to run the DC2 algorithm.

e) Total cost of the system: We evaluate the total cost of the system as follows:
Total Cost = Total Rkey calculation cost + Total Rciphertext calculation cost + Decryption cost. Here, the total cost is evaluated as the required time to run RKG, RENC, and DC2 algorithms

Here, we assumed that the $\mathbf{30\%}$ of the total users decrypting the Rciphertext as all the receivers may not decrypt the Rciphertext.

D. Results and Discussions

Fig. 4 shows the comparisons of the required time to generate Rkey of ABP scheme with other existing schemes. We consider the number of attributes to be $\mathbf{6}$. It shows that the required time to calculate the Rkey of the ABP scheme is more than CIBPRE16 [12], RIB-BPRE19 [13], and CBP23 [16] schemes and less than P2B20 [14] scheme as in ABP; we need to generate a Rkey for each coalition. In ABP, we consider the sum of time taken to generate Rkey for all the coalitions, whereas, for other schemes, a single Rkey is generated for all

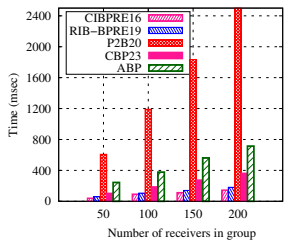

Fig. 4: Comparisons of RKG time of ABP with existing schemes.

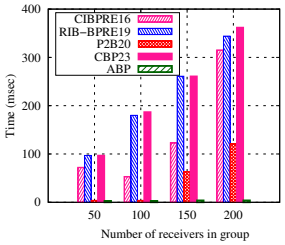

Fig. 5: Comparisons of DC2 time of ABP with existing schemes.

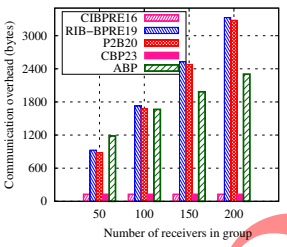

Fig. 6: Comparisons of communication overhead of ABP with existing schemes.

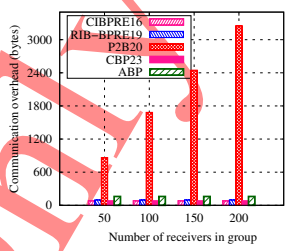

Fig. 7: Comparisons of communication overhead of ABP with existing schemes.

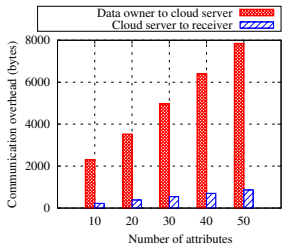

Fig. 8: Communication Overhead with varying number of attributes.

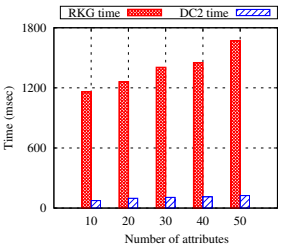

Fig. 9: Required time to execute RKG and DC2 with varying attributes.

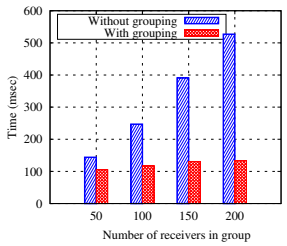

Fig. 10: Comparisons of DC2 time of ABP without grouping and with grouping protocol.

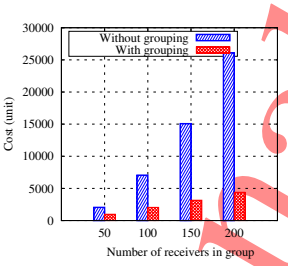

Fig. 11: Comparisons of total cost of ABP without grouping and with grouping protocol.

the identities. We consider that the count of attributes is **6** in Fig. 5. The required time for decryption of Rciphertext in the ABP scheme is much less than existing schemes. In the ABP scheme, we divide the identities among separate coalitions, and each decryption only considers the number of receivers in each coalition. In contrast, for CIBPRE16, RIB-BPRE19, P2B20, and CBP23 schemes, it depends on how many receivers are present, as for all of these a single Rciphertext is generated. Therefore, the decryption costs of existing schemes also increase. Fig. 6 shows the comparisons of the communication overhead from the data owner to the cloud server of the ABP scheme with other existing schemes. The communication overhead of the ABP scheme is less than the RIB-BPRE19 and P2B20 schemes and greater than the CIBPRE16 and CBP23 schemes. In ABP, the GroupFormulation algorithm generates a separate Rkey for each group. Therefore, the total count of Rkeys increases with increasing the count of groups. Fig. 7 shows the comparisons of the communication overhead from the cloud server to the receiver of the ABP scheme with other existing schemes. The communication overhead from the cloud server to the receiver of ABP scheme is almost similar to CIBPRE16, RIB-BPRE19, and CBP23 schemes and much less than the P2B20 scheme. In ABP, each Rciphertext only depends on the size of each coalition, which does not change with the increase in the number of receivers. Fig. 8 shows the communication overhead of ABP scheme with a varying count of attributes. We consider that the count of receivers is **100** in Fig. 8. The size of each Rkey increases with the count of attributes, and for each coalition, a separate Rkey is generated. Therefore, the total size of all the Rkeys increases. The communication overhead from the cloud server to the receiver increases as the size of each Rciphertext increases with the count of attributes. Fig. 9 shows how the required time to generate Rkey and decryption changes with a varying count of attributes. We consider that the total number of receivers is **100**. Here, The rate of increase in Rkey generation time is more than decryption time because we consider the total time to generate Rkeys for all coalitions. In contrast, the DC2 algorithm only depends on the number of attributes for a single coalition. Fig. 10 shows that the required time of decryption of Rciphertext of ABP is much less when we use grouping protocol than without grouping protocol. When the grouping protocol is not used, each decryption cost depends on all the group members as all the receivers are considered members of a single group. On the other hand, when the grouping protocol is used, the receivers are divided among different coalitions. Each decryption cost depends on the number of receivers present in each coalition, which is much less than the total number of receivers. Fig. 11 shows that the total cost of the system without a grouping protocol is much more than that of the system with a grouping protocol. In grouping protocol, the decryption cost only depends on the number of members of the particular coalition or the optimal coalition size, and the optimal coalition size is much less than the total number of receivers. Therefore, the total cost of the system is reduced when a grouping protocol is used in ABP. TABLE IV shows the required time to run the GroupFormulation algorithm with varying total numbers of receivers. For example, if the total number of receivers is 100, the required time is 1 microsecond which is negligible if we compare it with the time to generate the Rkey. Therefore, the GroupFormulation algorithm does not have much overhead in the Rkey generation process.

VII. Conclusion

This paper presents the ABP scheme and its SS-CPA security definition. We use both AB and BPRE concepts in the ABP scheme. Using ABP, an identity-based encrypted ciphertext is re-encrypted for some receivers from a particular group of IoT devices whose attributes match a specific access policy. Additionally, we used coalitional game theory to find the optimal coalition size from a group of identities of IoT devices to reduce the decryption cost, where the total cost of the system is also reduced. Using the random oracle model, we proved that the Rciphertext is secure under SS-CPA. Finally, we implemented the scheme using the pairing-based cryptographic library to show how the performance of the system varies with the number of attributes and identities. We also compared the system performance with the existing broadcast proxy re-encryption schemes.

In this work, we assumed that the total number of receivers is constant. We plan to extend the work in the future to dynamic groups, where new IoT devices can join the group and the existing IoT devices can leave the group.

References

[1] K. Liang, Q. Huang, R. Schlegel, D. S. Wong, and C. Tang, "A Conditional Proxy Broadcast Re-Encryption Scheme Supporting Timed-Release," *in Proc. the 9^{th} Int. Conf. Inf. Security Practice Experience*, pp. 132–146, 2013, doi :10.1007/978-3-642-38033-4_10.

[2] K. O.-B. O. Agyekum, Q. Xia, E. B. Sifah, C. N. A. Cobblah, and H. Xia, "A proxy re-encryption approach to secure data sharing in the internet of things based on blockchain," *IEEE Systems Journal*, 2022, doi: 0.1109/JSYST.2021.3076759.

[3] C.-I. Fan, J.-C. Chen, S.-Y. Huang, J.-J. Huang, and W.-T. Chen, "Provably secure timed-release proxy conditional reencryption," *IEEE Systems Journal*, 2017, doi: 10.1109/JSYST.2014.2385778.

[4] Y. Zhan, B. Wang, Z. Wang, T. Pei, Y. Chen, and Q. Qu, "Improved proxy re-encryption with delegatable verifiability," *IEEE Systems Journal*, 2019, doi: 10.1109/JSYST.2019.2911556.

[5] J. Gao, H. Yu, X. Zhu, and X. Li, "Blockchain-based digital rights management scheme via multiauthority ciphertext-policy attribute-based encryption and proxy re-encryption," *IEEE Systems Journal*, 2021, doi: 10.1109/JSYST.2021.3064356.

[6] S. Misra, A. Singh, S. Chatterjee, and M. S. Obaidat, "Mils-cloud: A sensor-cloud-based architecture for the integration of military tri-services operations and decision making," *IEEE Systems Journal*, 2016, doi: 10.1109/JSYST.2014.2316013.

[7] C. K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional Proxy Broadcast Re-Encryption," *ACISP, Springer*, vol. 5594, 2009, doi: 10.1007/978-3-642-02620-1_23.

[8] A. Alrawais, A. Alhothaily, C. Hu, X. Xing, and X. Cheng, "An Attribute-Based Encryption Scheme to Secure Fog Communications," *IEEE Access*, 2017, doi: 10.1109/ACCESS.2017.2705076.

[9] Z. Wang, "Leakage resilient ID-based proxy re-encryption scheme for access control in fog computing," *Future Generation Computer Systems, Elsevier*, vol. 87, pp. 379–385, 2018, doi: 10.1016/j.future.2017.12.001.

[10] Y. Tu, G. Yang, J. Wang, and Q. Su, "A secure, efficient and verifiable multimedia data sharing scheme in fog networking system," *Cluster Computing, Springer*, 2020, doi: 10.1007/s10586-020-03101-6.

[11] 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, doi: 10.1016/j.future.2017.05.026.

[12] 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 Trans. on Computers*, vol. 65, no. 1, pp. 66–79, Jan 2016, doi: 10.1109/TC.2015.2417544.

[13] C. Ge, Z. Liu, J. Xia, and F. Liming, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Trans. on Dependable and Secure Computing*, 2019, doi: 10.1109/TDSC.2019.2899300.

[14] S. Maiti and S. Misra, "P2b: Privacy preserving identity-based broadcast proxy re-encryption," *IEEE Trans. on Vehicular Technology*, 2020, doi: 10.1109/TVT.2020.2982422.

[15] C. Ge, W. Susilo, Z. Liu, J. Baek, X. Luo, and L. Fang, "Attribute-based proxy re-encryption with direct revocation mechanism for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, 2023, doi: 10.1109/TDSC.2023.3265979.

[16] S. Maiti, S. Misra, and A. Mondal, "Cbp : Coalitional game-based broadcast proxy re-encryption in iot," *IEEE Internet of Things Journal*, Apr 2023, doi: 10.1109/JIOT.2023.3265028.

[17] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, 2022, doi: 10.1109/TDSC.2021.3076580.

[18] 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, doi: 10.1109/ACCESS.2017.2726584.

[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, doi: 10.1016/j.comcom.2020.03.052.

[20] S. Maiti, S. Misra, and A. Mondal, "Mbp : Multi-channel broadcast proxy re-encryption for cloud-based iot devices," *Computer Communications*, 2024, doi: 10.1016/j.comcom.2023.11.020.

[21] Y. Zhang, R. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng, "Attribute-based encryption for cloud computing access control: A survey," *ACM Computing Surveys*, 2020, doi: 10.1145/3398036.

[22] A. Manzoor, A. Braeken, S. S.Kanhere, M. Ylianttila, and M. Liyanage, "Proxy re-encryption enabled secure and anonymous iot data sharing platform based on blockchain," *Journal of Network and Computer Applications, Elsevier*, 2021, doi: 10.1016/j.jnca.2020.102917.

[23] O. A. Khashan, "Hybrid lightweight proxy re-encryption scheme for secure fog-to-things environment," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2984317.

[24] Y. Zhang, J. Li, X. Chen, and H. Li, "Anonymous attribute-based proxy re-encryption for access control in cloud computing," *Security and Communication Networks, Wiley Online Library*, 2016, doi: 10.1002/sec.1509.

[25] J.-S. Coron, "A variant of Boneh-Franklin IBE with a tight reduction in the random oracle model," *Designs, Codes and Cryptography*, 2008, doi: 10.1007/s10623-008-9218-2.

[26] C. Meshram, "An efficient ID-based cryptographic encryption based on discrete logarithm problem and integer factorization problem," *Information Processing Letters, Elsevier*, vol. 115, pp. 351–358, 2015, doi: 10.1016/j.ipl.2014.10.007.

[27] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," *in Proc. the 14^{th} ACM conference on Computer and communications security*, pp. 456–465, 2007, doi: 10.1145/1315245.1315302.

[28] A. Chakraborty, A. Mondal, and S. Misra, "Cache-enabled sensor-cloud: The economic facet," *in Proc. the WCNC, IEEE*, 2018, doi: 10.1109/WCNC.2018.8377069.

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