CHAPTER 6
CONSIDERING SECURITY AND BIAS IN GROUP DECISION MAKING

Group Decision Making (GDM) involving Consensus Reaching Process (CRP) attempts to achieve a consensus among Decision-Makers (DMs) before coming to a final decision. Computer-based decision support systems are present to support the decision-making process called the Group Decision Support System (GDSS). The traditional GDSS being centralized is subject to security, transparency and trust issues, such as vulnerable to security risks providing attackers with a single target to attack, a single point of failure and biases. In this regard, this paper identifies and discusses such issues. To address these issues, we introduce a novel idea of a decentralized group decision making structure using blockchain technology. For Group Decision Making (GDM), we present a Consensus Reaching Process (CRP) model suitable for the blockchain platform. For validation, we implement the proposed work using the Ethereum blockchain. Furthermore, a theoretical security analysis of the proposed model is also done to validate that the system eliminates possible security attacks. Experiments validate that the proposed work minimizes the gas cost by minimizing the feedback cost. This work is the first step to introducing the idea, and the advanced approaches will be the natural consequences of this work.

6.1 Background

Various platforms, such as online social network platforms [7][8], promote information sharing, active participant communication, and faster information transmission. Using such platforms, the DMs participate in decision-making by providing their opinions without being physically present. A computer-based centralized decision support system

has been developed to support the CRP-based decision-making process called a Group Decision Support System (GDSS) [109]. Palomares et al. proposed a semi-supervised CSS (Consensus Support System) based on the MAS (multi-agent systems) paradigm, characterized by scalability and distributed computing capabilities to support large groups efficiently. The users are facilitated by the set of agents with high autonomy degree, capable of emulating different behavioural patterns based on the requirements of DMs. Several GDM support systems have been found in the literature review. A few of them is as follows: In LaSca [80], DMs can "decide on how to decide". MENTOR [81] is a graphical tool that studies the evolution of the preferences during the GDM process and DeciTrustNet [82] takes into account the trust and reputation in the social networks. However, this support system adopts the traditional models, in which an administration fully controls GDSS.

Several events occur in our daily lives that come with a decision problem where DMs do not trust such support systems. Delegating their local decision to the third party and listening to the feedback suggestion requires participants to trust the third party, believing it would be impartial. Thus, the centralization of these platforms has created a growing number of issues, such as issues of trust, security, and data censorship which causes less confidence for the participants to accept such systems for use [110] [111]. Since the information is gathered, organized and stored through a centralized entity, the information provided can be manipulated or misappropriated without the active control of the data owner. One of the most popular social networks, Facebook, was subject to a major scandal, i.e., the Cambridge Analytica's scandal [112]. An application launched on the Facebook was used by about 87 million users, that aims to collect the profiles of users. The collected data was delivered to Cambridge Analytica for political goals. This example explains the disclosure of privacy when interacting with a third party. The

discussed issue is not the only problem to be considered; censorship is another problem of such platforms. All scenarios presented here are the main forces leading to the decentralization of the GDSS.

In several research fields, Blockchain technology has been considered a revolutionary technology to overcome the issues of centralized systems. However, a trusted environment is needed to develop a decentralized GDM platform. Fortunately, recently emerged blockchain technologies can provide a trusted environment for developing decentralized GDM, addressing the issues of centralized GDM. Blockchain can provide immutability, security and transparency without a central authority. Therefore, it can be considered a potential approach to encouraging the individual's consciousness in participatory decision-making. Blockchain provides a unique address for each registered account, using which the transactions are performed instead of real identity to preserve an individual's privacy. Blockchain is applied to design various decentralized platforms in the field like cloud computing [85], service selection [86], social networks [113], social applications [87] and smart cities [88] to utilize its benefits. Ethereum blockchain provides the feature of smart contracts, popularly used to design decentralized systems. It eliminates the possibility of censorship, trust issues or third-party dependence [89]. However, there is no free lunch, hence the case with the Ethereum blockchain. Ethereum involves the concept of fee measured in terms of gas associated with the computation of every transaction executed. Gas price is the amount of ether the user is willing to spend on every gas unit.

As discussed in previous chapters, the consensus process allows DMs to gradually discuss and modify their opinions. So, to obtain the consensual decision, it involves the computational process in each iteration as such. CRP is supposed to be a time-consuming

process. However, in context to the Ethereum blockchain, for every transaction performed on the blockchain, a fee is charged in terms of gas, known as gas cost, which is limited on the blockchain platform. The DMs submitting their opinions and the moderator generating the feedback suggestions will be charged with the gas cost. Therefore, the existing consensus model [103], [104] will not work well when implemented on the blockchain as it may involve multiple transactions. Designing a GDM model where the consensual solution is achieved with minimal interaction is necessary. In this direction, this study designs a consensus-based GDM model which helps DMs and the moderator reduce their gas cost. This will be conducive to implementing the decentralized GDM structure using blockchain.

6.2 Issues in Centralized Group Decision Making System

In view of the above discussion, this work identifies few weaknesses emerging from the centralized nature of the group decision-making process. The decision support systems provide considerable convenience to users like the DMs. However, the GDSS being centralized has to face various issues such as it is vulnerable to many security risks providing attackers with a single target to attack, thus suffering from a single point of failure. This section discusses the key problems and issues faced by the existing GDSSs, i.e., group decision making using CSSs.

a) ***Single Point failure:*** In the centralized GDSS, all the data provided by the DMs are stored at a central server that coordinates the decision-making process and is vulnerable to a single point of failure. If the central server is compromised or goes down, the system will fail, leading to delays, data loss, and decreased efficiency and trust among group members. It indirectly will undermine the confidence of the DMs. Also, the involvement of a high risk of errors in the decision-making process may

provide inaccurate or incomplete data leading to a poor decision-making outcome. To mitigate this risk of a single point of failure, designing a system with multiple servers is important to add redundancy and hence the fault tolerance. By doing this, the GDSS can be seen as less vulnerable to single points of failure, ensuring the decision-making process is efficient and reliable.

b) ***Security Issues***: Generally, when a system communicates with external entities to provide facilities or services, it is more vulnerable to attack. All facilitation related to communication with external entities becomes a vulnerability exploited by possible threats resulting in the attack. The attacker may exploit the existing vulnerabilities of the GDSS to compromise the system. The centralized GDSS can be flooded with traffic to prevent the system from functioning properly, causing denial-of-service attacks. Also, the system sometimes fails to properly track and log the DMs activity, allowing the occurrence of newer actions by malicious manipulations. Even any of the DMs of the group can misuse the privileges to manipulate or steal the data or manipulate the decision-making process. Hence the GDSS should be secure enough to defend against such attacks.

c) ***Biased behavior of moderator:*** One of the main challenges of the GDM is the presence of a biased moderator, which can significantly challenge the decision result in such a way that stakeholders might not accept the final decision. The biased moderator may significantly impact the decision outcome. A moderator may have pre-existing biases that leads to favor certain opinions or alternatives over others, called as confirmation bias.

d) ***Collusion of Decision Makers:*** From the decision-making perspective, collusion of DMs can significantly impact the decision-making process. Collusion happens when two or more DMs come together to influence the decision process according to their

interest, causing a reduction in the opinion variance irrespective of the number of decision-makers. It can affect decision-making by distorting the information presented to the group members. This leads to incomplete or inaccurate information being used to make decisions that can affect negatively in the long run. Hence, to prevent DMs from colluding, the decision-making platform should hide the real identity of the participants from each other. Therefore, anonymity in blockchain can help prevent the collusion of DMs by making it challenging for malicious DMs to identify the specific DM within the group to collude with.

e) ***Lack of transparency:*** In the current decision-making structure, a single authority controls the decision-making process, leading to the issue of transparency in group decision-making. DMs need access to information related to the decision-making process, thus lacking trust in the decision-making process. Because of the trust issue, there will be a lack of confidence in the DMs while interacting with such systems. Furthermore, this may also lead to a lack of engagement of DMs. This is because the DMs do not have direct control over their preference and verify that their preference is considered without manipulation. Thus, we need a transparent system that can ensure the DMs' trust by providing enough transparency of the system that holds all the control.

f) ***Data Integrity:*** In a group decision making structure, the DMs provide their opinions in the form of preferences to the moderator in a decision support system. Since the preferential data provided by the DMs are analyzed, managed, and stored by the moderator, the chance of some intentional modification or accidental corruption either in the communication channel or at endpoints is possible. Hence doubt about the integrity of the data provided by the DMs arises.

g) ***Censorship*****:** Central authority holding the data can potentially censor it. They can prevent a part of the data from interacting with the rest. Like on social media platforms, say Twitter, any account or tweets can be censored, abandoning them from participating in decision-making. In this way, a quality decision cannot be made when the data provided is not utilized in the decision-making process.

6.3 Blockchain for Group Decision Making

The blockchain, with promising features such as decentralization, transparency, immutability, anonymity, etc., has the power to overcome most of the challenges faced by the present decision-making system. Using blockchain technology, participants can communicate with each other by sharing relevant information in the form of transactions without the involvement of any third party. The idea of blockchain for group decision making introduce this novel approach towards making the system secure, transparent and trustful for the participants. Nevertheless, the advanced approaches are likely to be natural consequence of this work. Here we investigate the requirements of blockchain for the group decision-making system.

a) Security requirements of CSS

Integrity of Transactions: When a computer-based support system is used to facilitate the group decision-making process, the information provided by the DMs in the form of opinions can be managed by either a dedicated system or by different intermediaries in the case of other existing platforms. This brings the risk of deliberately forging the information, and therefore the false or incorrect decision result can be obtained. Thus, the system must guarantee the integrity of the transactions and also it should prevent transactions from being tampered with. Blockchain is an efficient

approach to address the integrity of transactions since it is highly secured using cryptography.

Availability of System and Data: The availability here refers to both the availability at the system and transaction levels. System-level availability means that the decision-support system should run reliably even under network attack or failure. Furthermore, the transaction level availability means the DMs should access the data without being corrupted or inconsistent. The blockchain system is distributed and decentralized, eliminating the need for a central authority. This promises that the system is more resilient and available as there is no chance of a single point of failure. Blockchain supports the creation of smart contract that has the potential to automate the decision-making processes in the GDSS, thereby enhancing the availability of the GDSS.

b) Anonymity of Decision Makers' Identity

Some decision problems require the privacy of the DMs to be maintained, and their identities need to be hidden from the crowd or one another. To deal with such decision problems, blockchain, which is well known for its anonymity, could be employed to accomplish the anonymity of DMs in group decision making. The anonymity can prevent the collusion of DMs and thus prevent the blockchain from compromising its integrity. In the blockchain, each participant is recognized by their unique public address, which is not linked to any real identity. Thus, whenever a DM makes a transaction on the blockchain, that transaction is signed with a private key (associated with the sender's public key). This helps protect the DMs anonymity and ensures the integrity and authenticity of the transaction.

c) Resistance to DDoS attack

A Denial of Service (DoS) attack is a cyber-attack that aims to disrupt the host machine or its resources by making it unavailable to its intended user. Furthermore, when the incoming traffic comes from various distributed sources across the internet, it becomes a distributed denial-of-service (DDoS) attack. They attempt to overload the system by flooding it with continuous requests [114]. These attacks can be observed in a large-scale decision-making environment where many users can interact with the system. However, the blockchain, being a fully decentralized system where the consensus protocol is used for the generation and addition of new blocks, ensures the availability of blockchain even if some node goes offline. Thus, using a blockchain-based decision support system would be very helpful for emergencies where the time to make a decision is limited. With blockchain, the transaction processing will be continued even after some nodes go offline.

d) Transparency and Trust

In either case, the decision-making system requires DMs to sincerely provide opinions and accept the solutions. Two features of blockchain technology can improve the decision-making process. First, the system's trust can be gained when implemented on the blockchain, where validity, security, and auditability can be achieved inherently. Next, the decision-making is distributed across the blockchain network instead of controlled by a single authority. Smart contracts can also automate the decision-making process. These blockchain features ensure GDSS is transparent and trustworthy, ensuring fairness for all DMs when implemented. For example, a DM can use anonymous accounts in order to check their preferences and can also verify that their opinions are considered without manipulation.

The present decision-making system, as discussed above, has some weaknesses. The situations that arise in group decision making are the following: (i) A group of DMs

frequently communicates relevant information, such as preferences over the set of alternatives; (ii) the system obtains the collective value and provides feedback suggestions to the inconsistent DMs; (iii) Depending upon the obtained suggestions, DMs modify their preferences. This process of generating feedback suggestions and modifying preferences will run until the predefined consensus threshold is reached. However, there are challenges in dealing with such systems: (i) The system lacks transparency because of which the third party cannot be trusted and hence questions the authenticity of information communication, (ii) For some sensitive applications, the validation of transmitted information is a priority, and so is data authenticity and integrity, and (iii) DMs can form collusion and favor a certain opinion or alternative. These challenges not only limit the safety of information provided by the DMs but also lacks transparency and so the lack of confidence of the participating DMs in the decision-making system. To overcome the issues of the centralized decision-making structure and provide DMs with a more secure and trustful platform, a decentralized GDM system is required leveraging the features of blockchain. Blockchain would facilitate the distributed decision-making and consensus building process with the use of self-executing smart contract.

6.4 Proposed Decentralized Group Decision Making

To improve the decision-making systems' performance and cope with previously mentioned challenges, we propose a decentralized group decision-making structure leveraging the advantages of blockchain. The proposed blockchain-based GDM structure achieves an automation degree (leveraging the features of blockchain) by means of smart contracts as well as making distributed CRP possible. In this direction, blockchain technology provides a tool that enforces security in these distributed decision processes. This section describes the various key architectures and design details of our proposed

blockchain-based group decision making. An Ethereum smart contract will be used to manage the process of group decision making.

6.4.1 Overall System Architecture

The proposed system architecture is shown in Fig. 6.1. The architecture comprises three main participants with access to the Ethereum smart contract through the internet: Decision-makers, miners, and a moderator. Decision-makers have unique Ethereum addresses (with public keys and private keys) and connect with the smart contract through a front-end application. The key role of the different participants has been summarized here as follows.

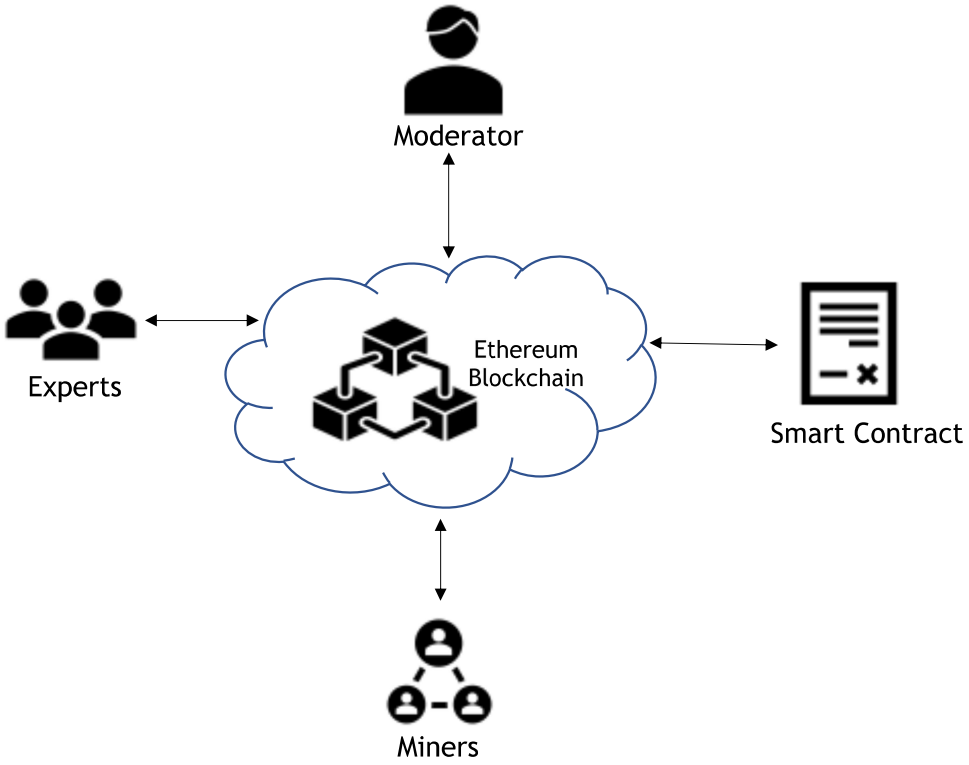

Fig. 6.1: Architecture of Proposed GDM

Moderator: The moderator is responsible for managing the DMs' access control list and permissions for participating in the group decision making. The moderator is the owner and the creator of the smart contract. The owner of the smart contract, i.e., moderator, can

add DMs. The main task of the moderator is to register and de-register the DMs in the system. Furthermore, the moderator gives feedback to the DMs through the smart contract to change their opinions for reaching the consensus threshold.

Decision Makers: In group decision making, two or more decision-makers come together to achieve a common solution to a given problem consisting of various alternative solutions to such a problem [115]. DMs may come from different background, knowledge, expertise but has the same goal. DMs provide their preference about an alternative with respect to other ones, using an adequate preference structure and expression domain. After initially providing their opinions as a preference, they negotiate among themselves to reach a consensus.

Smart Contract: The smart contract is used in our proposed solution for the whole system. The smart contract contains the list of all the authenticated DMs and their provided preferences. Moreover, it evaluates the consensus measure of each DM and then generates a global consensus value and the modified DM preferences. Consensus measure, feedback mechanism, and selection process are decentralized through the smart contract.

Miners: Mining is the process of creating a block of transactions to be added to the Ethereum blockchain. Miners solve computationally difficult puzzles to produce the blocks and legitimate transactions in the block and publish the block in the network. Others validate the block and add a valid block in the local blockchain.

6.4.2 Proposed Decentralized Group Decision Making Model

To ensure the transparency and the security of the opinion exchange (or the information exchange), we propose blockchain-based GDM defined by the smart contract for decision making, which can be used for the group decision making discussed in section 2. As

already discussed in Ethereum public blockchain, interacting with the smart contract must be done by sending the transaction to the blockchain. Some computation is required to process the transaction by the miner. Thus, the participant has to pay a fee also known as gas for this computation.

In general, the CRP is an iterative and dynamic process that involves multiple rounds of discussions. Many authors have proposed consensus models in order to deal with CRP [23], [109], [115], [116]. Processes in these models are coordinated by a moderator responsible for supervising and guiding decision-makers in the whole process and giving them the advice to modify their preferences [104][12]. This requires a high computational effort and the generation of more transactions by the participants, which can incur a high transaction fee, making the process costly. To leverage the benefits of blockchain technology, we provide a cost-efficient consensus-based group decision making model that aims to obtain the consensual solution at once. To reduce the transaction fees to be paid by the DMs, we provide a consensus reaching model that deals with minimizing the transaction fees and provides a quick decision based on the predefined consensus threshold. We here illustrate the proposed blockchain-based model for the group decision making process in a secured and trusted environment.

(i) Consensus Measure

In GDM, the DMs must participate in the discussion process to reach a consensus solution. In the consensus process, a method for obtaining the consensus solution for DMs is introduced based on the degree of deviation between the individual DMs' decision matrix and the group decision matrix [15]. In the proposed GDM model, there is a problem to solve for which the opinions of the DMs are required. For the specified problem, a set of feasible alternatives, $x = \{x_1, x_2, \ldots . x_n\}$ where $(n \geq 2)$, and the group of two or more DMs, $d = \{d_1, d_2, \ldots . d_m\}$ where $(m \geq 2)$ characterized by their own

experience, ideas, backgrounds and knowledge who express their opinion about the provided set of alternatives. Let $P^k = (p_{ij}^k)_{n\times n},\ for\ i < j$ be the preference vector of the decision maker $d_k \in D$. Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ be the weight vector of the decision maker, where $\lambda_k \geq 0$ is the weight associated with the DM d_k and $\sum_{k=1}^{m} \lambda_k = 1$. The collective preference $P^c = (p_{ij}^c)_{n\times n},$ for $i < j$, is obtained as:

$$p_{ij}^c = \sum_{k=1}^{m} \lambda_k . p_{ij}^k \tag{6.1}$$

Based on the distance between the individual decision matrix and collective preference matrix denoted as $D(P^k, P^c)$, the individual consensus degree of DM d_k is obtained as given in Eq. (6.2).

$$D(P^k, P^c) = \frac{1}{n \times (n-1)} \sum_{i=1}^{n} \sum_{j=1, i\neq j}^{n} \left| p_{ij}^k - p_{ij}^c \right|$$

$$ICD(d_k) = 1 - D(P^k, P^c) \tag{6.2}$$

And then, the group consensus degree (GCD) can be calculated as:

$$GCD = \sum_{k=1}^{m} \lambda_k \times ICD(d_k),\ \ for\ i, j = 1, \ldots, n \tag{6.3}$$

Obviously, $GCD \in [0, 1]$. The larger the value of GCD, the higher the consensus among experts. In general, it is impossible to achieve a full consensus among experts. Therefore, soft consensus is adopted in the consensus reaching process. In order to achieve soft consensus in GDM, a consensus threshold $cl \in [0, 1]$ is predefined to measure the consensus degree among the experts. It is the minimum required consensus among the experts. If the obtained GCD is greater than or equal to the threshold, i.e., $GCD \geq cl$, then the expected soft consensus is achieved. Whereas if $GCD < cl$ the feedback process is carried out to modify the resulting soft consensus. The feedback process to guide DMs below the consensus level is discussed below.

(ii) Feedback Mechanism

In the general CRP framework described in Chapter 2, the collective preference relation P^c is of direct use to produce feedback advice to the decision-makers with lower consensus degree. The decision-makers then modify their opinion with reference to the advice provided. However, this interactive process achieves the required level of agreement in several rounds, which will be time consuming and costly. DMs, when submitting their opinions again and again based on the feedback advice provided by the moderator, will add on the transaction fees (measured in terms of gas) at the DMs end, while computing and generating the feedback advice will charge the fees at the moderator's end until the consensus is reached. Thus, it is required to provide appropriate feedback advice to burden the participants with minimum fees. For this, we propose a feedback mechanism based on the predefined consensus threshold that determines the level of agreement among the DMs.

In general, the minimum required consensus level, also called consensus threshold, is the minimum required degree of agreement ($cl \in [0,1]$) among the DMs. That means, if the value of threshold $cl = 0$, it implies no agreement among the DMs and if $cl = 1$, it implies full agreement among them over the decision, which is ideally not possible. Therefore, there will always be the possibility of some degree of disagreement among DMs, which is allowed in every decision-making situation. Hence, provided the minimum level of consensus, we can obtain the maximum degree of disagreement $(1 - cl)$ which will be used in this work to formulate the feedback process. Thus, the proposed feedback mechanism provides personalized advice to DMs so that the required consensus level can be achieved in a single round. Usually, the feedback mechanism consists of two rules [103] [61]:

Identification rule used to identify the DMs with consensus degree lower than the consensus threshold

Direction rule aims to provide advice to the inconsistent DMs to achieve the group consensus level based on the predefined threshold. The second rule is elaborated below.

For $d_k \in d$, $D(P^k, P^c) > (1 - cl)$, then d_k will be called an inconsistent decision maker and the feedback mechanism should be carried out for the decision maker d_k.We propose to compute customized advice for the decision maker that varies according to the group preference matrix and the consensus threshold cl . To do so a feedback matrix FM^k is computed for an inconsistent DM d_k at a maximum disagreement degree from the initial group preference matrix P^c. The method for obtaining the feedback matrix in the feedback process is introduced based on two factors: the degree of deviation between an individuals' decision matrix P^k and the group decision matrix P^c, i.e., $D(P^k, P^c)$ and the predefined degree of disagreement $(1 - cl)$. Based on these two factors, a feedback matrix $FM^k = (fm_{ij}^k)_{n\times n}$ will be generated to advice the d_k. And this received advice if properly considered, the group consensus would reach threshold without undergoing any other round for negotiation. The feedback matrix FM^k is calculated using Eq. (6.4).

$$FM^k = P^c + \frac{(1 - cl)}{D(P^k, P^c)}(P^k - P^c) \tag{6.4}$$

FM^ksatisfies the following:

$$D(FM^k, P^c) = \frac{1}{n \times (n-1)} \sum_{i=1}^{n} \sum_{i=1, i\neq j}^{n} \left| fm_{ij}^k - p_{ij}^c \right| = (1 - cl)$$

Specifically, replacing $\frac{(1-cl)}{D(P^k,P^c)}$ by δ_k in (6.4), we have

$$fm_{ij}^k = (1 - \delta_k).p_{ij}^c + \delta_k.p_{ij}^k \tag{6.5}$$

which guarantees that $fm_{ij}^k + fm_{ji}^k = 1$ and $fm_{ij}^k \in [min(p_{ij}^c, p_{ij}^k), max(p_{ij}^c, p_{ij}^k)] \in [0,1]$. Thus, to improve the consensus level of the identified DM, the advised adjustment direction is:

$$\begin{cases} \overline{p}_{ij}^k \in [min(p_{ij}^c, fm_{ij}^k), max(p_{ij}^c, fm_{ij}^k)], & i \geq j \\ \overline{p}_{ji}^k = 1 - \overline{p}_{ij}^k, & i < j \end{cases} \tag{6.6}$$

Notice that because of $fm_{ij}^k \in [min(p_{ij}^c, p_{ij}^k), max(p_{ij}^c, p_{ij}^k)]$, this adjustment direction ensures that the adjusted preference $\overline{p}_{ij}^k$ will be closer to the collective preference than the preference p_{ij}^k that gets replaced and that too in the range of consensus threshold. After the DM modify their preference in the advised range, they certainly will achieve group consensus. Then the updated group decision matrix $\overline{P}^c = (\overline{p}_{ij}^c)_{n\times n}$ can be obtained using the weighted average operator given in Eq. (2.1). The aim of the feedback mechanism based on the consensus threshold is to carry out different adjustment suggestions, which respects the decision makers' initial preferences too. It is to note in this study we assume that decision makers being cooperative accept the feedback recommendations and do not present non-cooperative behaviors.

6.5 Interactions and Message Sequence

The interaction among the system entities to carry out the group decision making process is illustrated in Fig 6.2. Fig. 6.2 shows a sequence diagram for a successful group decision making process defined by a consensus reaching process. The moderator first creates the smart contract and registers the decision-makers through the *addDecisionMaker()* function. Each decision-maker is provided with a unique Ethereum Address (EA). The decision-maker provides opinions to the smart contract specifying the preference relations. The smart contract will check the saved list of authorized lists of authorized DMs. If the one sending (or providing) the preference by invoking function *addOpinion()*

is not authorized to participate in the GDM process, a reject event will be issued. Otherwise, the smart contract will accept the preference provided if the DM is authorized.

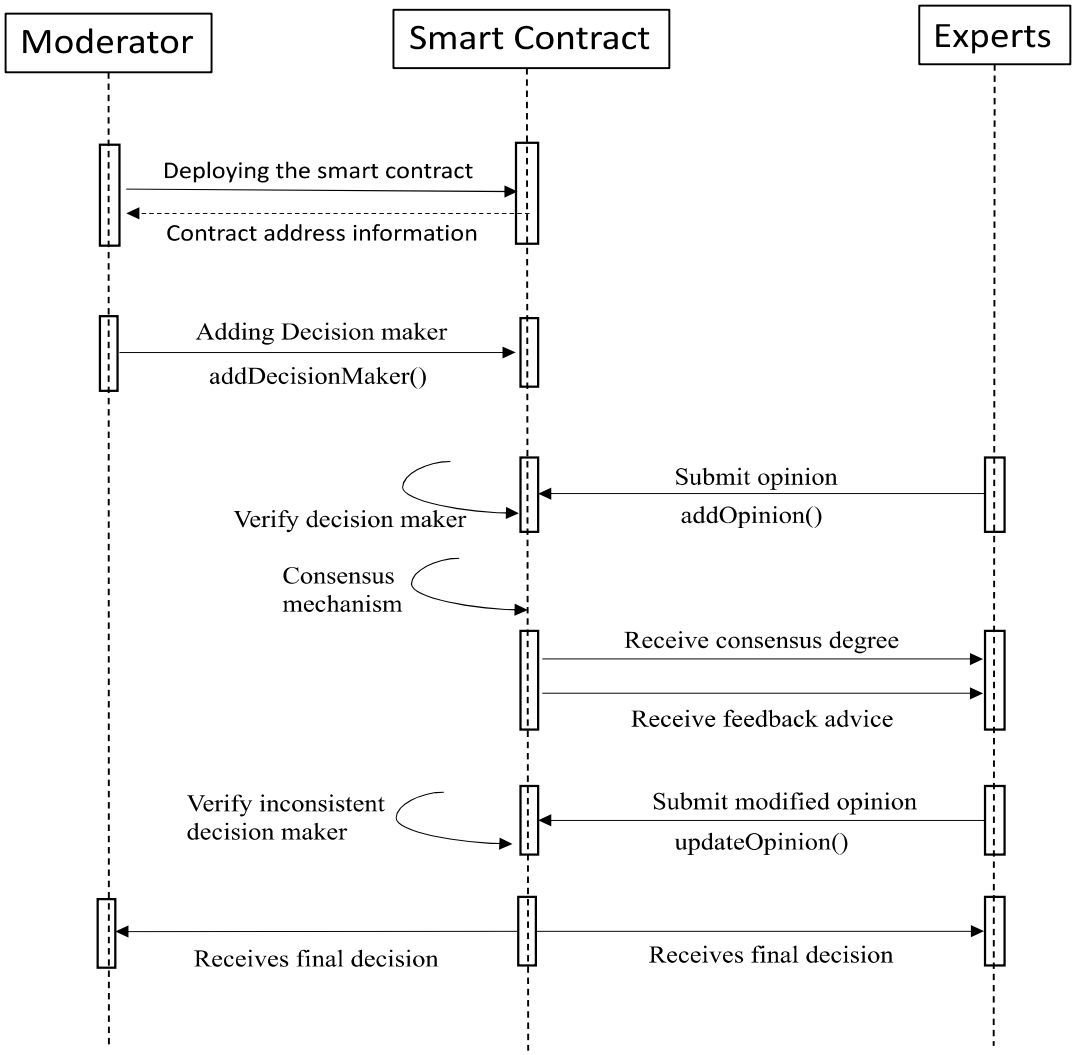

Fig. 6.2: The Proposed System: Sequence Diagram of Group Decision Making

Upon receiving the preferences of all the listed DMs, the moderator initiates the consensus reaching process invoking function *measureConsenus()*. It will first measure the initial consensus measure of all DMs and will identify the DMs with consensus level below the predefined consensus threshold. The identified DM will be advised to adjust their opinions using the given feedback by the smart contract. Then the DM submit their modified opinion invoking function *updateOpinion()*. Now again, the consensus measure will be determined, and the obtained final decision will be informed to every participant.

6.6 Implementation

This section highlights the key implementation aspects related to the smart contract for the proposed GDM model. Fig. 6.3 shows what the contract needs to do at each process in detail. Our smart contract was developed using solidity programming language and implemented and tested using Remix IDE (remix.ethereum.org), which provides rich features for testing and debugging smart contracts before deploying them. The implemented smart contract includes the four main components: A) Deployment of the smart contract: B) Adding DMs and their opinions, C) Evaluation of consensus and feedback generation, and D) Modification of opinion of inconsistent decision-makers.

***Deployment of the smart contract**:* The moderator deploys the group decision making contract on the blockchain when he/she wants to conduct the decision-making process. The function *deploy()* takes the input as the number of DMs (m), the number of alternatives (n) and the consensus threshold value (cl)**.** The initialization of the contract state is done by initializing the number of DMs and the number of alternatives by invoking a constructor.

Adding DMs and their opinions: After the contract is deployed, the moderator needs to add the DMs to keep track of all participants in the decision-making process. Only after they get added to the list, decision-makers can provide their opinions using *addOpinion()* restricted by the modifier function for the decision-makers only. The opinion given by the DM is in the fuzzy preference relation described in Definition 1 of Chapter 2. The function first checks if the decision-maker is listed in the DMs list, then receives and stores the listed DMs' opinion.

Evaluation of consensus and feedback generation: At this phase, once the opinions of all the DMs are received, the moderator interacts with the decision-making contract to figure out the consensus degree among the DMs by invoking *measureConsensus()*

function and identifies the one with lower degree of consensus. The consensus measure is defined using Eq. (6.2) which determines the degree of consensus of a DM. For the one with a consensus degree lower than the consensus threshold is provided with feedback advice (defined in Eq. (6.6)).

Modification of opinion of inconsistent DMs: After receiving the customized feedback advice from the smart contract, the decision-maker invokes the function *updateOpinion()* to submit the modified opinions to reach consensus. The opinion of the DM is modified as per Eq. (6.6).

After completing the above phases, the contract usually outputs a final decision result. The moderator then again executes the *measureConsensus()* to obtain the degree of consensus among DMs. Based on the proposed GDM model, the consensus is reached at once, provided that all the DMs show cooperative behavior. The DMs may also reflect non-cooperative behavior, thereby manipulating or delaying the consensus process. However, this paper does not focus on the non-cooperative behavior of DMs, a separate research area to be explored.

As already discussed, for every function of a smart contract invoked, Ethereum incurs some cost in terms of gas price. The gas cost is charged for the one who calls the function, and as the number of times the function call increases, the gas cost charged increases. This is where our proposed CRP is suitable in the blockchain context. In the proposed CRP, the feedback mechanism is designed so that the moderator generates feedback suggestions only at once and accordingly, and the DMs modifies opinion at once. Whereas in the traditional CRP [103], [46] [34], reaching consensus at once is not guaranteed, and it may consume multiple rounds to reach consensus. Thus, compared to the proposed CRP, the traditional CRP needs to call the functions more and hence

incurring the greater gas cost charged. Therefore, the proposed model reduces the gas cost charged and thus is suitable for deployment on the blockchain. Here in our proposed GDM model, we consider that DMs are cooperative enough and modify opinions as provided by the moderator. Also, the selection of the consensus threshold is decided according to the practical requirement of the decision problem.

Variable and structure	address Moderator struct DecisionMaker { address ad; uint[] alt; uint cl; uint[] fm; uint wt; } DecisionMaker [] public DM;
deploy:	Receiving (numDecisionMaker, numAlternatives) from **Moderator**
addDecisionMaker:	Receiving(address) from **Moderator** if DM.length< numDecisionMaker: DM.push(address) Else: Revert ();

addOpinion:	receiving (alternatives, fm) from **DecisionMaker**
	for each DM in DecisionMaker:
	if (msg.sender == DM.address):
	DM.push(DecisionMaker (msg.sender,alternatives,0, fm)

updateOpinion:	receiving (updated_alt) from **DecisionMaker**
	for each DM in DecisionMaker:
	if (DM.address == msg.sender and DM.cl<threshold):
	DM.alt = updated_alt

createGV:	receiving None from **measureConsensus** gv = weightedAverage(DM.alt) return gv;
Consensus:	receiving (gv) from **measureConsensus** for each DM in DecisionMaker: DM.cl = avg((abs(gv,DM.alt))); return DM.cl;

measureConsensus:	receiving (address) from **Moderator**
	gv = [], fm = [], cl = [], wt;
	gv = createGV();
	cl = Consensus(gv);

	cm = weightedAverage (DM.cl);
	for each DM in DecisionMaker:
	if (cm<threshold):
	identify DM.cl < threshold
	fp = (1-threshold)/(1 – DM.cl);
	DM.fm = feedback(DM.alt, gv, fp);
	return DM.cl, DM.fm;

feedback:	receiving (DM.alt, gv, fp) from **measureConsensus**
	fm = [];
	fm = fp x gv + fp x DM.alt;
	return fm;

Fig. 6.3: Smart Contract for Group Decision Making

6.7 Experimental Analysis

In this section, a hypothetical emergency plan is first used to demonstrate the proposed consensus-reaching model, and then the proposed CRP model is compared against the traditional CRP model. After that, we present the security analysis of the proposed blockchain-based decentralized GDM and how it attempts to address the security concerns concerning data integrity, transparency, and public verifiability. Lastly, some experimental analysis is provided to justify the proposed consensus model.

6.7.1 Demonstration of the Proposed GDM

A smart contract, as discussed above, has an Ethereum account that has some balance and can send transactions over the network. However, it is not controlled by any of the participants once deployed to the network and run as programmed. The smart contract for group decision making is designed so that it first lists the decision-makers by adding their account address using the function *addDecisionMaker()*. The moderator (the creator of the contract) adds the decision-makers to the list. Suppose that some unconventional emergency situation happened that requires an optimal resource allocation. The disaster management team has to execute a plan for allocating resources and hence requires a group of DMs to take the decision. Suppose $x = \{x_1, x_2, x_3, x_4\}$ are the four emergency

plans recognized in advance for a certain unconventional emergency event like an earthquake or terrorist attack. Let a group of six DMs say $d = \{d_1, d_2, d_3, d_4, d_5, d_6\}$ are invited to provide their opinions regarding the execution of the plan. After that, the decision-makers' account added by the moderator can then interact with the smart contract. The decision-makers interact by submitting their opinions in the form of transactions by executing the function *addOpinion()*. We implemented the proposed contract on the remix IDE, and the opinion in the form of preference relation provided by the decision makers is shown in Table 6.1.

Table 6.1: Preference Provided by Decision Makers

	p_{12}	p_{13}	p_{14}	p_{23}	p_{24}	p_{34}
P^1	0.40	0.10	0.21	0.55	0.6	0.77
P^2	0.20	0.12	0.58	0.20	0.10	0.90
P^3	0.54	0.35	0.86	0.95	0.78	0.68
P^4	0.31	0.54	0.30	0.65	0.53	0.62
P^5	0.05	0.80	0.45	0.82	0.44	0.82
P^6	0.34	0.20	0.84	0.50	0.25	0.85

Once the opinion of all the DMs is received, the moderator executes the function *measureConsensus()* to determine the degree of consensus among the decision-makers. As discussed, the consensus measure first calculates the group decision and obtains the individual's consensus degree. We here considered that all decision-makers are of equal importance. The obtained group value using the weighted average operator given in Eq. (6.1) is shown in Table 6.2.

Table 6.2: Group Preference of Decision Makers

	p_{12}	p_{13}	p_{14}	p_{23}	p_{24}	p_{34}
P^c	0.31	0.35	0.54	0.61	0.45	0.78

It then evaluates the individual's degree of consensus and generates feedback using function *feedback()* for those with lower consensus degree. Let us suppose that the predefined consensus threshold for the given decision-making problem is 0.9. Eq. (6.2) is used to generate the consensus degree of the DMs. Comparing the obtained individual

consensus degree $ICD(d_k)$, all 6 DMs have a lower consensus degree than the predefined consensus threshold. Therefore, the feedback advice is generated using Eqs. (6.5) and (6.6) for the decision-makers with lower consensus degree, shown in Table 6.3.

Table 6.3. Consensus Measure and Feedback Advice

	$ICD(d_k)$	Advise Adjustment Direction					
		p_{12}	p_{13}	p_{14}	p_{23}	p_{24}	p_{34}
d_1	0.8507	[0.31, 0.36]	[0.18, 0.35]	[0.32, 0.54]	[0.57, 0.61]	[0.45, 0.55]	[0.77, 0.78]
d_2	0.8001	[0.25, 0.31]	[0.23, 0.35]	[0.54, 0.56]	[0.40, 0.61]	[0.27, 0.45]	[0.78, 0.84]
d_3	0.7688	[0.31, 0.41]	[0.34,0.35]	[0.54, 0.67]	[0.61, 0.76]	[0.45, 0.59]	[0.73, 0.78]
d_4	0.8715	[0.30, 0.31]	[0.35, 0.49]	[0.35, 0.54]	[0.61, 0.64]	[0.45, 0.51]	[0.65, 0.78]
d_5	0.8024	[0.18, 0.31]	[0.35, 0.58]	[0.49, 0.54]	[0.61, 0.71]	[0.44, 0.45]	[0.78, 0.82]
d_6	0.8649	[0.31, 0.33]	[0.24, 0.35]	[0.54, 0.76]	[0.53, 0.61]	[0.30, 0.45]	[0.78, 0.83]

The inconsistent decision-makers after receiving the feedback advice, would use it to modify their initial opinions according to Eq. (6.6). Decision-makers then invoke *updateOpinion()* to update their opinions after receiving the feedback. The updated opinions of the decision-makers provided are given in Table 6.4.

Table 6.4: Updated Preference of the Decision Makers

	p_{12}	p_{13}	p_{14}	p_{23}	p_{24}	p_{34}
$\bar{P}^1$	0.37	0.18	0.32	0.57	0.55	0.77
$\bar{P}^2$	0.25	0.23	0.56	0.41	0.27	0.84
$\bar{P}^3$	0.31	0.35	0.60	0.51	0.49	0.77
$\bar{P}^4$	0.31	0.42	0.38	0.61	0.46	0.72
$\bar{P}^5$	0.26	0.57	0.51	0.40	0.44	0.80
$\bar{P}^6$	0.33	0.24	0.57	0.49	0.40	0.82

Table 6.5: Updated Group Preference of Decision Makers

	$\bar{p}_{12}$	$\bar{p}_{13}$	$\bar{p}_{14}$	$\bar{p}_{23}$	$\bar{p}_{24}$	$\bar{p}_{34}$
$\bar{P}^c$	0.31	0.34	0.49	0.50	0.44	0.79

After receiving the updated opinions of decision-makers, the moderator contract invokes function *measureConsensus()* to obtain the consensus degree of the decision-makers which is $CD(\bar{P}^1, \bar{P}^c) = 0.9084$, $CD(\bar{P}^2, \bar{P}^c) = 0.9030$, $CD(\bar{P}^3, \bar{P}^c) = 0.9376$, $CD(\bar{P}^4, \bar{P}^c) = 0.9492$, $CD(\bar{P}^5, \bar{P}^c) = 0.9499$ and $CD(\bar{P}^6, \bar{P}^c) = 0.9679$, where the obtained consensus degree of each decision-makers is improved over their

initial preferences. Then the group consensus degree obtained is 0.9360 which is greater than the consensus threshold 0.9 and hence the final solution is $\overline{P}^c$.

It is to understand that with each transaction, a cost in terms of gas has to be paid by the user initiating the transaction. Thus, decision-makers will always try to modify their opinions according to the feedback provided by the moderator; otherwise, they may be asked to modify their preferences again. Hence, to avoid bearing the transaction's cost, decision-makers will try to modify their opinions accordingly.

6.7.2 Analysing the Proposed GDM model

In order to analyze our proposed GDM model implemented in the smart contract, we replace the Eq. (6.5) with Eq. (6.7), to compare the proposed CRP with the existing consensus reaching model [46] to show the novelty of the work. The adjustment process generated in the feedback mechanism in traditional CRP is given in Eq. (6.7).

$$\begin{cases} fm_{ij}^k = \delta . p_{ij}^c + (1-\delta) . p_{ij}^k, & i \geq j \\ fm_{ji}^k = 1 - fm_{ij}^k, & i < j \end{cases} \tag{6.7}$$

where $\delta \in [0,1]$ is a feedback parameter to control the degree of advice given to inconsistent DMs. The value of δ controls the emphasis on group opinion and individual opinion. When $\delta > 0.5$, the model will lead DM towards the collective opinion whereas when $\delta < 0.5$, the model will lead DMs towards their own opinion. Once the fm_{ij}^k is generated for inconsistent DM d_k, Eq. (6.5) is used to update the opinion. In the traditional CRP, feedback parameter will be same for all the DMs and throughout the decision-making process, whereas in our work the feedback parameter δ_k for each inconsistent DM d_k depends on threshold cl and degree of deviation $D(d_k)$. Since threshold cl is the same for all the participants, we conclude that feedback parameter δ_k for DM d_k depends on $D(d_k)$ for a given threshold. Thus, without altering the essence of the CRP, and by

replacing the Eq. (6.5) with Eq. (6.7), we compare the proposed CRP with the traditionally existing CRP. First, we compare the proposed CRP with the existing CRP based on the opinions provided by the 6 DMs in the section 6.7.1, and obtained results are shown in Table 6.6. In this experiment, we evaluated the CRP on two different values of feedback parameter, i.e., δ = 0.4 and δ = 0.65, in order to show the validity of the comparison. From Table 6 following observations can be drawn: The number of iterations required to reach consensus using the proposed CRP is less than the existing CRP defined using Eq. (6.7), which is as expected. For the value of the final consensus, δ = 0.65 is greater than the value at $\delta = 0.4$. This confirms the saying that model will lean more toward the group for $\delta > 0.5$ and more towards DMs' opinion for $\delta < 0.5$. Lastly, in all three cases, the ranks of the alternatives are the same. Hence, the validity of the proposed method.

Table 6.6. Evaluation of Proposed CRP for Different Values of δ_k

At $\lambda = 0.9$	Feedback Parameter		Initial Consensus	Final Consensus	No. of Iterations	No. of inconsistent DMs		Ranking result
						Iteration 1	Iteration 2	
Proposed CRP	Depends on $D(d_k)$	Different for different DMs	0.8264	0.9360	1	6	-	$x_3 > x_2 > x_4 > x_1$
CRP using Eq. (6.7)	Voluntarily selected	$\delta = 0.4$	0.8264	0.9232	2	6	3	$x_3 > x_2 > x_4 > x_1$
		$\delta = 0.65$	0.8264	0.9289	2	6	3	$x_3 > x_2 > x_4 > x_1$

In the second experiment, we compared the proposed CRP with the existing CRP discussed using Eq. (6.7) for the number of iterations required to reach a consensus. For the given setting, the experiment is performed for different values of threshold ranging from [0.8,1]. The graph that compares the number of rounds required to reach consensus in the proposed model and the traditional one discussed in Eq. (6.7) is shown in Fig.6.4. It can be observed that the number of iterations to reach consensus for the existing CRP increases with increase in the threshold value. In contrast, for the proposed CRP, the

number of iterations goes only up to 1 for any threshold value. This ensures the validity of the proposed method.

Following observations can be drawn from two experiments and the obtained results.

Observation 1: In most CRPs [103], [46], the adjustment advice produced by the feedback mechanism is used as reference advice for the DMs' to adjust their preferences. But these CRPs do not consider the predefined consensus threshold that has to be ultimately achieved. For the case solved in section 6.7, compared to these CRPs, the proposed model provides the adjustment suggestions based on the individual's degree of deviation from the group preference matrix and the consensus threshold. The proposed GDM model is analyzed against the traditional GDM model. In order to analyze the proposed GDM model, based on the opinions provided by the 6 DMs in section 6.7, the results in Table 6.6 show that the proposed CRP converges to the consensus faster than the CRP that selects the feedback parameter voluntarily using Eq. (6.7), where the feedback parameter is the same for all inconsistent experts.

Observation 2: For feedback process described in Eq. (6.7), when δ is greater than 0.4, DMs will converge more to the group decision compared to their own opinion. For the value of δ smaller than 0.4, Eq. (6.7) will show a higher impact of the DMs' initial opinion over its modified opinion. Different values of δ will show different results, and accordingly, the number of iterations may increase or decrease to reach consensus in traditional CRP.

Observation 3: It is not desirable for DMs to repeatedly accept feedback advice and change preferences accordingly in real-world decision-making situations. The reason is that decisions are required to be taken quickly because of time constraints, like in emergency group decision-making [106]. In such cases, the proposed model respecting

the DMs' initial opinion generates feedback recommendations such that consensual decision is obtained quickly, which is more reasonable than the works in [26], [27]. Also, the parameter consensus threshold *cl* should be determined properly as this determines the maximum allowed degree of disagreement among the DMs. The graph that compares the number of rounds required to reach consensus in the proposed model and for the traditional one discussed in Eq. (6.7) is shown in Fig. 6.4.

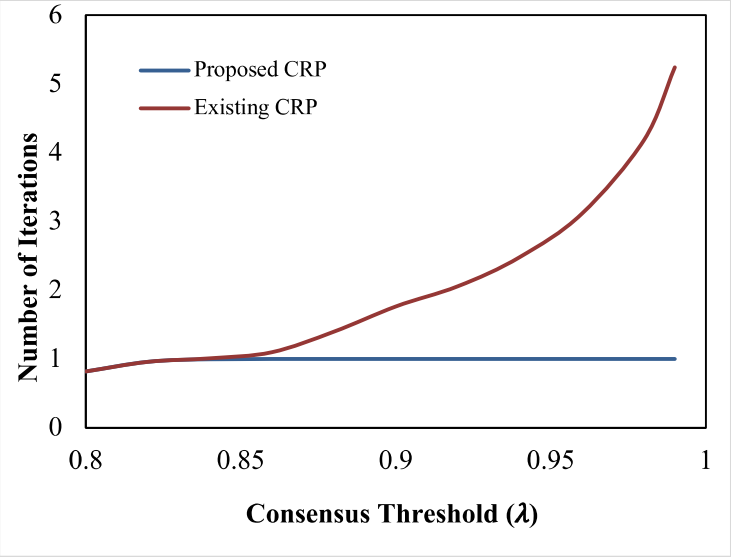

Fig. 6.4: Average Number of Iterations Required to reach Consensus

6.7.3 Security Analysis

This section discusses the possible threats and attacks on the overall group decision structure and how the proposed GDM defends against these threats. The moderator deploys the smart contract as per the prescribed GDM model. We assumed that the decision-makers show cooperative behavior. In contrast, the moderator can perform malicious acts like tempering the data or being biased towards any decision-maker or alternative. Note that the moderator cannot perform any malicious activity with the smart contract. Hence, the situation is analyzed when the moderator is malicious. We prove that our proposed contract achieves the security goals in the following.

Decision Maker Authentication is provided using a blockchain-based GDM structure in which the decision-makers interact with the smart contract and get authenticated before participating in the GDM process. To date, the authentication methods of any GDM structure (be on some social media platform or dedicated decision-making platform) are centralized. This approach reflects drawbacks like a single-point failure, hacking, high cost and privacy invasion [117]. With the blockchain-based GDM protocol, the DMs get registered first by the moderator and only provide their opinions. There is no chance that non-registered DM can provide their opinions. This scheme facilitates the management of DM while providing security.

Transparency is achieved by deploying the group decision making smart contract on a public blockchain. It provides full transparency to all the participants. We know that smart contracts are the executable code hosted on the blockchain to store information, thus improving transparency. Also, the blockchain is designed to prevent any participant from modifying the data, transactions or contract. It results in increased trust and thus eliminates the sense of biased behaviour seen in established group decision making systems. Blockchain provides immutable storage of data and transactions, resulting in ensuring data provenance. Moreover, this ultimately enforces data integrity and data transparency too.

Integrity and Non-repudiation are the two essential security requirements needed to overcome the challenges of the existing system. These security measures are required to verify the identity of the sender of information and avoid modifying data. Non-repudiation is the assurance that the DMs cannot deny their opinion. The smart contract can guarantee it in our work. All DMs submit their opinion preference by publishing the blockchain transactions, which ensures that their opinion cannot be modified and denied by the features of blockchain. Therefore, once a decision-maker

submits the opinion about its preference for alternatives, it cannot deny and modify the opinion. By achieving these security requirements system can be protected against the attacks like Man-in-The-Middle attacks and replay attacks. Also, our system is secured against Denial-of-Service attacks (DoS) as the list of DMs, and their opinions are stored on the public ledger, which is decentralized and distributed and not subject to failure or hacking. The public Ethereum blockchain is highly resistant to DDoS attacks as it is distributed globally under the concept of decentralization.

Public verifiability A ensures that the participants can verify the opinions, feedback or the final decision of the process. In the proposed protocol, all the participants' data is transmitted in a public channel accessible by every participant. Therefore, the final decision produced can be verified using such public data. Hence, it would not be possible for the malicious participant to claim the false or incorrect result without being identified.

6.7.4 Gas Cost analysis

As already discussed, executing any transaction on the Ethereum network requires a discrete amount of gas, such as for the execution of a smart contract or the execution of a function changing the state of the smart contract. Thus, each line of code written in solidity demands a certain amount of gas quantity to be executed. Thus, specifying the sufficient amount of gas to be needed is important for the proper execution of the contract on time.

“Ethereum Gas” is the fundamental unit measuring the computational effort required to execute any transaction. Thus, the amount of gas needed is specified under the consideration of gas price and gas limit. The gas limit defines the total gallons of gas kept in the smart contract tank. This amount should be sufficient enough so that all operations mentioned in the construct get executed. Contrary to that, gas price is

associated with the gas consumption of smart contracts. Since, executing any transaction on the Ethereum network requires a fee. Gas denotes the fee required to conduct a transaction on the Ethereum environment. The Ethereum currency, ether (ETH) is used to pay the gas fee. Gas prices are denoted in gwei where 1 gwei = $10^{-9}ETH$. The reason why gas fees are required to be paid is to keep the Ethereum network secure (ethereum.org). Preventing the malicious entities from spamming the network, fees are associated with every computation performed on the network. The gas fee is required to be paid to keep the Ethereum network secure. To prevent the malicious entities from spamming the network, fees are associated with every computation performed on the network. "Gas" is the fundamental unit of computation. Since two types of entities, i.e., moderator and decision-makers, are involved in group decision making. We analyze the gas cost paid by both types of entities. Here we evaluate the gas cost performance of the proposed blockchain-based GDM for its experimental results. We implemented experiments on Remix solidity IDE.

Transaction Gas Cost of Moderator: The smart contract designed for the GDM allows moderator to invoke the function *addDecisionMaker()* and *measureConsensus()* only, discussed in Fig. 6.3. But the cost of invoking these functions may depend on the number of alternatives, the number of decision makers and the value of the threshold. Considering all these parameters one by one, we perform the experiment to observe the gas cost of the moderator. The gas cost of the moderator is observed for the proposed GDM model and compared against the traditional GDM discussed in Eq. (6.7).

Transaction Gas Cost of Decision Makers: While interacting with the blockchain-based GDM, a DM has to submit the opinion and update the opinion as per the feedback provided if found inconsistent. Thus the gas fee it gets charged will be from the two functions: *addOpinion()* and *updateOpinion()*. The function *updateOpinion()* is

conditional. The cost of these functions depends on the number of alternatives and the threshold value. The average gas cost of decision-makers is observed for the proposed GDM model and compared against the traditional GDM discussed in Eq. (6.7).

Experiment 1: Gas cost of the example

In the example discussed in section 6.6, the average gas cost associated with each function is observed and given in Table 6.7. As already discussed, the DM d_k has to deal with only *addOpinion()* and *updateOpinion()*. From Table 6.7 it can be observed that the cost of invoking function *addOpinion()* has to be incurred by all the DMs. While the cost of invoking *updateOpinion()* is conditional, which is paid by that DM whose consensus degree is below threshold. On the other hand, the moderator invokes function *addDecisionMaker()* for registering the DMs and the *measureConsensus()* function for measuring the degree of consensus and generating feedback to those with lower consensus degree.

In the example discussed in section 6.7, the average gas cost associated with each function is observed and given in Table 6.7. As already discussed, the DM d_k has to only deal with the functions *addOpinion()* and *updateOpinion()*. The function *addOpinion()* will be called initially by all the DMs in order to submit their opinion. Therefore the cost of invoking *addOpinion()* function will be incurred by all the DMs. Invocation of function *updateOpinion()* is conditional, whose cost is paid by the DM with lower consensus. In the example, all the 6 DMs are inconsistent and so all of them need to update their opinion. On the other hand, the moderator invokes function *addDecisionMaker()* for registering the DMs and the *measureConsensus()* function for measuring the degree of consensus and generating feedback to those with lower consensus degree. After the DMs update their opinion as per the feedback provided to them, the function *measureConsensus()* is

invoked again to obtain the final decision. Thus, the function *measureConsensus()* is called two times in the example.

Table 6.7. Gas Cost for Invoking Functions

	addDecisionMaker()	*addOpinion()*	*measureConsensus()*	*updateOpinion()*
Moderator	345834	–	1509934	–
d_1	–	245344	–	84386
d_2	–	245332	–	84386
d_3	–	245332	–	84386
d_4	–	245332	–	84386
d_5	–	245332	–	84386
d_6	–	245332	–	84386

Gas cost analysis for the comparison discussed in Table 6.6 is shown in Table 6.8. It is clearly observable in Table 6.8 that the total gas cost incurred by the moderator and the total average gas cost incurred by the DMs is greater for the traditional model than the proposed model. This is because the proposed model incurs *measureConsensus()* at once only whereas in the traditional model the function *measureConsensus()* can be called for more than one time and hence the reason for greater gas cost. Hence the proposed model is suitable in the blockchain structure with respect to gas cost.

Table 6.8. Comparison of Total Gas Cost of DM and Moderator

		Total cost of Moderator	Total average cost of Decision Maker
Proposed CRP		1855768	329730
Existing CRP	$\delta = 0.4$	2239731	357858
	$\delta = 0.65$	2239731	357858

Experiment 2: Impact of number of alternatives on gas cost of the moderator

In this experiment, number of DMs is 6 and consensus threshold is 0.90. The opinions of DMs are generated randomly. We vary the number of alternatives, and observe the gas cost of the moderator, shown in Fig. 6.5.

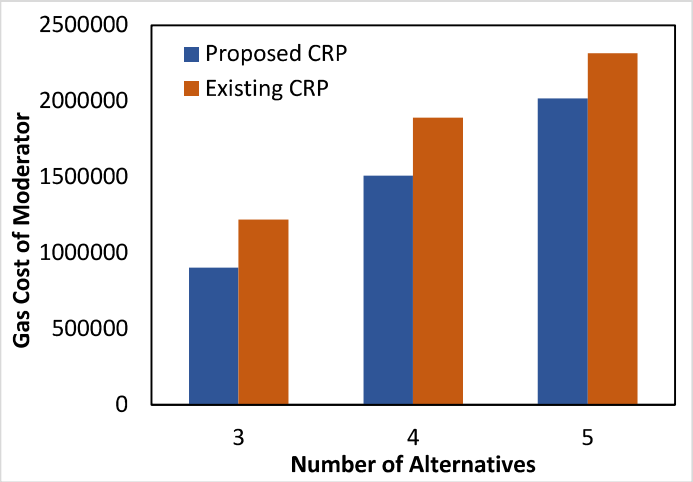

Fig. 6.5: Impact of Number of Alternatives on Gas Cost of Moderator

One can observe that gas cost increases as the number of alternatives increases, because the number of alternatives determines computational cost of moderator in *measureConsensus()*. Also, the gas cost of the moderator in case of the proposed CRP is found to be less than that of the traditional CRP (where $\delta = 0.40$). The reason behind this result is as follows. For a given number of decision-makers, the cost of adding the decision-maker will remain constant. Whereas the cost of invoking the function *measureConsensus()* is different for both the CRP models. The reason is that the proposed CRP model produces the consensual result in one feedback round only, while the existing CRP model has to produce feedback until consensus is reached. So, in the existing CRP model, the *measureConsensus()* will be called several times. Thus, the transaction cost of the moderator will always be higher for the traditional GDM model than the proposed one.

Experiment 3: Impact of number of decision makers on gas cost of the moderator

In this experiment, the number of alternatives is 4 and the consensus threshold is 0.90. For the different number of DMs, the obtained average gas cost of the moderator is given in Fig. 6.6.

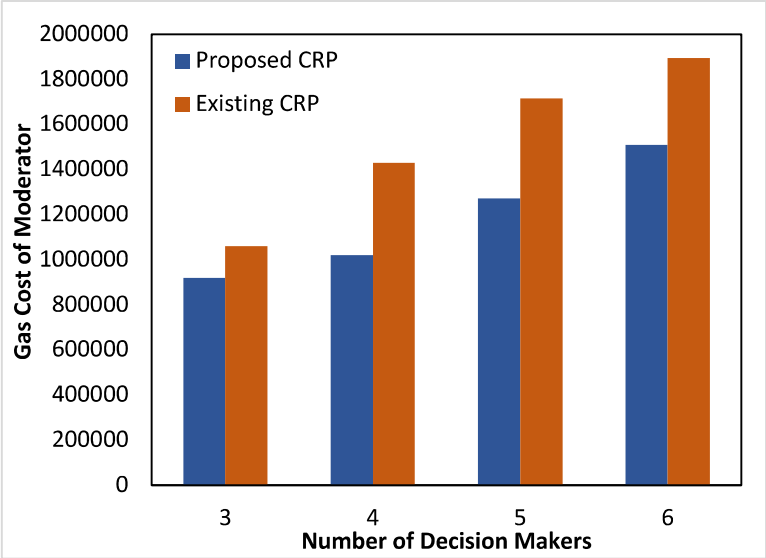

Fig. 6.6: Impact of Number of Decision Makers on Gas Cost of Moderator

It can be observed that the gas cost of the moderator increases as the number of DMs increases, and despite that the gas cost of the moderator is higher in case of the traditional GDM (where $\delta = 0.40$) than the proposed GDM. The reason behind this result is as follows. Several DMs having different opinions leads to a complex decision-making process, which adds to the computational cost of *measureConsensus()*. Compared with the traditional GDM, the consensus is reached in such situations gradually in several steps with multiple invocations of *measureConsensus()* function until consensus is reached. Thus, the gas cost of the moderator is higher for traditional GDM than the proposed GDM. The cost associated with the moderator for invoking function *addDecisionMaker()* will only depend on the number of decision makers added which will remain same in both the cases.

Experiment 4: Impact of consensus threshold on gas cost of the moderator

In this experiment, number of DMs is 6 and the number of alternatives is 4. The opinions of DMs are generated randomly. The value of consensus threshold λ is varied between $[0.85, 1]$. For the different values of threshold, the obtained average gas cost of the moderator is given in Fig. 6.7.

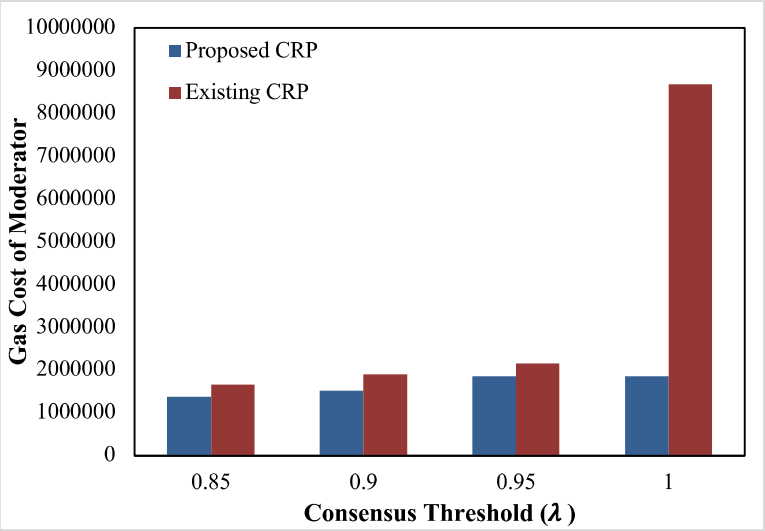

Fig. 6.7: Impact of Consensus Threshold λ on Gas Cost of Moderator

It can be observed that the gas cost of the moderator increases with an increase in the consensus threshold value, and the moderator has to pay the higher gas cost in the case of the traditional GDM (where $\delta = 0.40$) compared to the proposed GDM. This is due to the fact that the number of inconsistent DMs increases as the threshold value increases. In the proposed method, the consensus threshold is used to determine the feedback advice to the DMs with lower consensus degree by invoking the *measureConsensus()* function. This function invocation at a higher threshold value adds computational cost to the moderator as it inherently calls *feedback()* function for more of the DMs. In the case of traditional GDM, the number of rounds required to reach consensus increases with the increase in the consensus threshold. This burdens moderator invoking *measureConsensus()* multiple times until consensus is reached unlike the proposed method, hence the reason for higher gas cost.

Experiment 5: Impact of number of alternatives on average gas cost of the DMs

For observing the impact of the number of alternatives on the average transaction gas cost of the DMs, an experiment is performed where the number of DMs is 6 and the consensus threshold is 0.90. While performing the experiment, we find that the cost associated with

the functions *addOpinion()* and *updateOpinion()* increases on increasing the number of alternatives (since the preference matrix size increases), shown in Fig. 6.8.

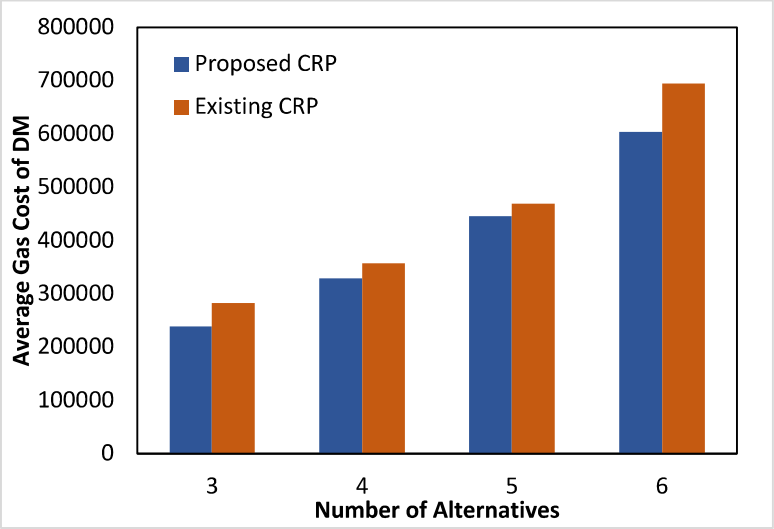

Fig. 6.8: Impact of Number of Alternatives on Gas Cost of DM

As discussed above, the proposed GDM model provides feedback in such a way that inconsistent DM reaches consensus at once after accepting the feedback advice. Whereas the traditional GDM model discussed in chapter 2 may produce feedback advice many times until consensus is reached. For this reason, the DM has to invoke *updateOpinion()* multiple times and hence the transaction fees imposed. We can say that *updateOpinion()* is the only differentiator that increases the cost of DM in traditional GDM. From the Fig. 6.8, we can clearly see that the transaction gas cost of the proposed GDM model is always smaller than the traditional GDM model (where $\delta = 0.40$), as discussed.

Experiment 6: Impact of the consensus threshold on the average gas cost of the DM

In this experiment, number of DMs' is 6 and the number of alternatives is 4. The value of consensus threshold λ is varied between $[0.85, 1]$. The opinions of DMs are generated randomly. For the different threshold values, the obtained average gas cost of DM is given in Fig. 6.9.

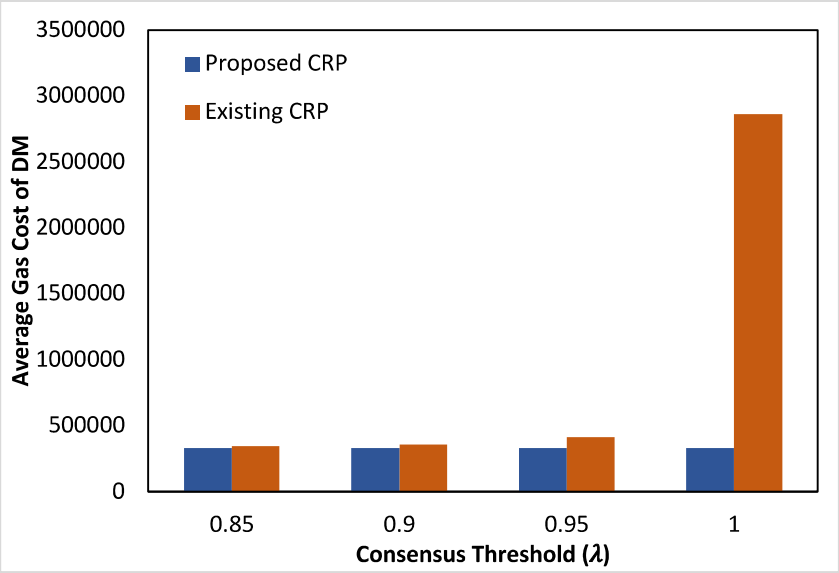

Fig. 6.9: Impact of Threshold Value on Gas Cost of DM

It can be observed that with increase in the consensus threshold value, the average cost of a DM increases. As the threshold value increases, the DMs have to adjust their opinions more towards the group to reach the consensus. Thus the number of inconsistent DMs increases with the threshold value, which adds the cost associated with the function updateOpinion(). Another observation that can be drawn from the figure is that the cost paid by the DM in the proposed CRP is always less than the cost incurred in the existing CRP. This is because the proposed method achieves consensus after one iteration whereas the traditional GDM the number of times feedback required increases with increase in threshold.

6.8 Discussion

One can observe in this work that proposed decentralized GDM using Blockchain resolves many of the issues of existing GDM models. E.g., decentralization overcomes the issue of single-point failure, anonymity prevents DMs from colluding, and immutability ensures data provenance. Since the Blockchain is highly secured using cryptography, it is an efficient approach to addressing the integrity of transactions. The shared blockchain and every transaction broadcast allow each participant to create the

chain independently. This feature of Blockchain allows the GDM process to be transparent, and hence trust is established. But in order to overcome all such deficiencies of the existing decision-making models, the DMs and the moderator has to bear the cost called gas cost in Ethereum Blockchain for every transaction they perform. In order to minimize the gas cost to be paid by the DMs or the moderator, the proposed GDM model is suitable for the blockchain environment. By implementing the proposed model on the public Ethereum blockchain, we find some interesting findings regarding the moderator's gas cost and the DM. It is observed that with an increase in the number of DMs, the number of alternatives, and the value of the consensus threshold, the gas cost of the moderator increases. The DM average gas cost increases with the increase in the number of alternatives and the threshold value. Another interesting finding is the gas cost of the moderator, and the DM is always less for the proposed CRP model than the existing CRP model. Hence it can be said that the proposed model is suitable for the decentralized blockchain environment. However, these findings are based on the public Blockchain for managing and controlling the participants in GDM in order to reach a consensual solution. The public Ethereum blockchain charges the gas cost to the participants of the decision-making process. Thus, by shifting to the private or consortium Blockchain, the gas cost can be replaced with digital assets known as tokens. These tokens can be used to track and monetize the transactions [118]. But the problem with the consortium blockchain is that tampering is possible if the majority of the dominant organization wants [119], which is an important issue for future work.

6.9 Summary

This paper discusses the issues of the existing centralized group decision making (GDM) systems, where blockchain can be leveraged to decentralize the GDM systems, which holds several advantages compared with the centralized GDM systems. In particular,

equipped with blockchain technology, decentralized GDM systems can guarantee the transparency, anonymity and security of the DMs' opinions across the system. To overcome the issues of the centralized decision-making structure and provide DMs with a more secure and trustful platform, a decentralized GDM system is proposed leveraging the Ethereum blockchain platform. This platform facilitates the distributed decision-making and consensus reaching process (CRP) with the use of self-executing smart contracts. In order to make the decision-making suitable for execution on the blockchain, we introduced a CRP that reduces the transaction fees paid on the public Ethereum Blockchain by minimizing the number of feedback rounds required to achieve consensus. We presented the security analysis of our proposed GDM system and discussed the security goals achieved. Moreover, the gas cost analysis is done to show that the proposed decentralized GDM is less costly in terms of Gas than the traditional GDM process. It is important to note here that to validate the proposed decentralized GDM, we implemented the CRP for decision-making on the Ethereum blockchain. However, the same can be implemented on any other blockchain platform supporting smart contracts like Hyperledger. The idea of blockchain for group decision making begins to introduce this novel approach towards making the system secure, transparent and trustful for the participants. Nevertheless, the advanced approaches will likely be a natural consequence of this work. Here we investigate the requirements of blockchain for the group decision-making system.