International Journal of Advanced Technology and Engineering Exploration, Vol 8(85)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.874588

Research Article

A novel algorithm for secure and reliable coordinator election in distributed networks

Shital Subhashchandra Supase[1*] and Rajesh Baliram Ingle[2]
Research scholar, Department of Computer Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, Maharashtra, India[1]
Professor, Department of Computer Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, Maharashtra, India[2]

Received: 11-August-2021; Revised: 20-December-2021; Accepted: 22-December-2021

Abstract
In distributed networks, electing an entity to act as a coordinator is a crucial agreement problem. This coordinator election process is vulnerable to security attacks because of the unreliable communication channel used for message passing in the election algorithms. For example, electing a coordinator is a challenge in many blockchain consensus algorithms. There is a need to ensure the safety and liveness of coordinator election algorithms. In this work, the security challenges in the election algorithm are identified and a novel method for coordinator election is designed. A Secure and Reliable Coordinator Election Algorithm (SRCEA) is proposed for reliable and secure coordinator election in distributed networks. The performance of SRCEA is analyzed and compared with the existing secure coordinator election algorithms. SRCEA elects a coordinator node securely with O(n) communication cost and storage space cost where n is the number of member nodes in the system. The communication cost of SRCEA is comparatively 97% less. The computation cost of SRCEA is also considerably less as compared to the existing algorithms.

Keywords
Secure coordinator election, Distributed networks, Election method, Safety.

1.Introduction
Electing a reliable coordinator is a crucial task in peer-to-peer distributed network. The peer-to-peer distributed network comprises a number of peer entities. These member entities elect the coordinator entity by consensus [1–3]. Different consensus algorithms exist for electing an entity as a coordinator and those are generally termed as coordinator election algorithms. The majority of the existing election algorithms focus on improving the performance measures in particular communication cost [4–6]. As the entities in the system are networked and use message passing technique for communication and execution of distributed election algorithm, electing a reliable entity as a coordinator is a challenge. The unreliable communication channel used in message passing makes the election algorithm vulnerable to security attacks [7, 8] The unauthorized access may result in a security and privacy attack which includes the impersonation and denial of voting attack [9, 10]. Unauthorized modifications result in the modification of the election messages and attack on the integrity of the election.

In a blockchain-based application, the behavior of any protocol comprises basic steps where a client issues a transaction, a coordinator election protocol determines a coordinator to marshal the transaction and commit on an ordering proposed by the coordinator [11–13]. In such critical blockchain-based applications, it is important to implement coordinator election protocol safely. The objective of this work is to select competent candidates for the election process and safely elect a reliable candidate as a coordinator. To achieve this, the authors have proposed a novel algorithm for coordinator election. Consider a distributed network with n nodes where all entities have a unique identity assigned to them as P_1, P_2, P_3... P_n. Each node has a set of attributes associated with it. These nodal attributes for all member nodes are stored in a group view G_v at every node. A Secure and Reliable Coordinator Election Algorithm (SRCEA) is proposed out of which the

*Author for correspondence

first phase selects the candidate nodes based on nodal attributes and the second phase carries out the actual voting and election of a coordinator. Say, at time t_1 node P_4 is elected as a coordinator of the system. Assume that at t_1+d, node P_4 fails and its failure is detected by node P_1.

The failure of the coordinator triggers the election algorithm to execute. The nodal attributes associated with member nodes like node's distance from the center of the network, joining time-stamp of the node and earlier failure count of the node are considered. System uses the min-heap tree for storing the node identities according to their nodal attribute values. Min-heap tree root holds the minimum value of these attributes. The desired value of each nodal attribute is a minimum value that is retrieved and stored in P_{idmin}, $P_{ijtsmin,}$ and P_{ifcmin} for distance, joining timestamp, and failure count nodal attributes respectively. The retrieval time for the root of the min-heap tree is constant *O(1)* hence selection of candidate nodes takes constant time [14]. In the first phase of the election process, a set of candidate nodes C is generated. The criterion used for selecting the candidates using nodal attributes is uniformly known to all nodes. The prioritized vote message is sent by the member nodes to candidates. As the entropy of election messages for in election algorithm is low, the different security attacks such as cipher text-only are likely for

1. Accessing and analyzing the votes cast by the nodes
2. Modify and resend the vote messages
3. Deleting the vote messages

Whereas, impersonating a member node for voting is also possible if voter nodes are not authenticated. In this work, an algorithm SRCEA is proposed to elect a reliable entity as a coordinator. Our contribution lies in identifying the security challenges in the coordinator election algorithm and proposing a new solution approach. SRCEA uses a security mechanism that ensures the security of election vote messages even though the entropy of the messages is low and does not allow to establish a relationship among the encrypted messages. SRCEA performance is analyzed and compared with the Secure Extrema Finding Algorithm (SEFA) and Secure Preference-based Leader Election Algorithm (SPLEA) designed by Vasudevan et al. [15].

The literature review is presented in section 2. The proposed coordinator election method and algorithm are discussed in section 3. Section 4 presents the results and comparison of SRCEA performance measures (communication cost, computation cost, and storage cost) with SEFA and SPLEA. The outcome of this work and its limitations are presented in section 5. The conclusion of the work and future scope is given in section 6.

2.Literature review

Voting methods used for electing a coordinator or leader in distributed systems can be categorized as extrema-finding and preference-based methods. In the extrema finding election voting method, a coordinator node is chosen based on the extreme value of node identity. A node with either minimum or maximum node id is elected as a coordinator whereas, in the preference-based election voting method, the nodes are given preferences. A node receiving a maximum number of highest preferences is elected as a coordinator [3, 16]. Dan et al. have proposed three different schemes for a Single Secret Leader Election (SSLE) and designed protocols based on obfuscation, Fully Homomorphic Encryption (FHE), and Decision Diffie-Hellman (DDH) [1]. The protocol keeps the identity of the leader secret until it is made public by the chosen leader itself. SSLE is easy to implement but it is not taking into consideration the credibility of nodes during the election process.

Secure leader election algorithms for wireless ad-hoc networks are proposed by Vasudevan et al. [15]. SEFA and SPLEA algorithms proposed by Vasudevan et al. [15]. are round-based algorithms that use Public Key Infrastructure (PKI) for the confidentiality of the election vote messages and the Message Digest (MD5) hashing technique for the integrity of the election messages. SEFA and SPLEA are kind of benchmark algorithms for secure election in distributed networks. The communication cost of these algorithms is higher as compared to other algorithms as they use PKI which needs to exchange a greater number of messages for exchanging the public keys of nodes needed in the encryption process. The computation cost of these algorithms is also greater because of modular and exponential operations used in the key generation, encryption, and decryption process of PKI. In a one-hop network that corresponds to the best case in SEFA and SPLEA, every node must perform *O(n)* verifications and *O(1)* signatures. The verification cost is as a cost of competing with every other node and when every node is *L* hops away from every other, the node has

to perform *O(n)* verifications and *O(L)* digital signature operations.

EffatParvar et al. [17] proposed a novel election approach with an improvement in the Bully and Ring election algorithm extrema finding a method for the election process. The authors have proposed an algorithm that uses a max-heap tree for electing the coordinator. Total memory used by the proposed algorithm is *4n* and the election is completed in *O(logn)* messages whereas the Bully and Ring algorithm takes $O(n^2)$ messages to elect a coordinator. No any nodal attribute is considered during election process in algorithm proposed by Mohammad et al. that may lead to election of an unreliable node as a coordinator.

The election Sidik et al. [18] have proposed the Practical Agile Leader Election (PALE) algorithm which terminates in bounded time. PALE operates with desynchronized clocks and jittering nodes. The PALE is designed to work in the partially asynchronous system and uses an extrema-finding method for the election that elects a coordinator node in *O(n)* messages.

Coordinator election algorithms in distributed systems are implemented using message passing. The messages used in node communication are sent through the insecure communication channel which makes these messages vulnerable to security attacks. Denial of voting and impersonation attacks are the common security attacks on the process of coordinator election in distributed systems [7]. Hence the security issues in the coordinator election algorithm need to be addressed. It is important to ensure the confidentiality and integrity of the election vote messages.

Bellare et al. [19] have discussed a conventional authenticated-encryption mode based on a symmetric-key encryption mechanism. Authenticated Encryption (AE) and Authenticated-Encryption with Associated-Data (AEAD) are proposed for a block-cipher mode of operation. AEAD's important characteristic is that it is online and it ensures the confidentiality and integrity of the messages. AEAD approach uses symmetric encryption and hashing which makes group communication secure and confidential whereas hashing algorithms used ensure the integrity of the election vote messages.

Al-Mamun et al. [20] have analyzed the security by Advanced Encryption Standard (AES) algorithm. The authors concluded that the AES is the most secured algorithm for message passing in distributed systems as symmetric encryption algorithms need not to exchange the additional keys than the only secret key used for encryption and decryption both.

Jackson [21] has designed a model for an election that addresses the security challenges of confidentiality of the election messages and election process is completed in *O(n)* messages. Election algorithm behavior in various states is discussed by Stephen. The encryption algorithms are used for ensuring the confidentiality of the election vote messages. The issue of the election vote message's integrity is not addressed by Stephen.

The dynamic leader selection algorithm by Madisetti and Panda [22] selects a set of future leaders which are then alerted before the failure of the current leadership and handed over the leadership. This algorithm selects a leader or a coordinator instead of electing it through the election process which is suitable in micro-services-based applications. The security issue in the coordinator election algorithm is not addressed by Madisetti and Panda [22].

Chaparala et al. [23] have designed a three-phase algorithm that identifies and filters the nodes for election in the first phase. Secondly, the filtered nodes are validated for the determination of prime nodes using group communication. Finally, the Prime node is identified and accepted as a coordinator in the acceptance phase. This approach is designed for improving the efficiency of the system via following performance metrics such as time and communication complexity.

The authors have proposed algorithm to elect a coordinator in *O(n)* messages as compared to the exiting algorithms. Fault-tolerance, data aggregation, mobility, quality of service, security, and privacy challenges in leader election are identified by Rahman [24]. The limitations in the current leader election algorithms in IoT, and possible techniques to overcome them are also discussed by Mohsin.

Pitfalls in Ring and Bully algorithms are discussed by Subramanian et al. [25]. The authors have discussed an approach for combining these two algorithms using hypergraphs to overcome the drawbacks identified. Safety and liveness issue is not addressed in work by Subramanian et al. [25].

Mohammed et al. [26] have proposed a mechanism design-based secure leader election model. The authors have addressed an important issue about the credibility and reliability of the leader node by proposing a solution approach where the author considers nodes with the most remaining resources to be elected as the leaders. For addressing the issue of selfish nodes, authors have presented a solution that is based on mechanism design theory where the nodes receive incentives in the form of reputations for honestly participating in the election process. Authors have addressed the issue of reliable leader election but not focused on safety and liveness of election process.

Barki et al. [27] have revised the SSLE protocol. As discussed earlier a group of participants randomly choose exactly one leader in SSLE and the identity of the leader is kept secret. The authors have focused on the security model of the SSLE proposed by Dan et al. [1] and more specifically on the liveness property of the security model.

Byrenheid et al. [28] have proposed attack-resistant leader election in social overlay networks. The authors have addressed the issue of node impersonation during the election process. Authors have proposed the three-majority voting method that uses asymmetric encryption for ensuring the confidentiality of the messages. But election algorithms using asymmetric encryption can be costly as they need a greater number of messages to be exchanged and more time to perform modular exponentiation operations.

It is observed from the existing coordinator election algorithms that the majority of the algorithms are designed to improve the performance parameters of the algorithm such as communication cost and storage cost. The advantage of the majority of the election algorithms is that they use extrema finding method for an election which is simple to implement. Whereas, the major disadvantage of the algorithms discussed is that the nodal attributes are not taken into consideration during the election process. Electing any node as a coordinator without considering its attribute may result in the election of an inefficient coordinator. The issue of considering credible nodes as candidates during the election process and an important issue of security is rarely addressed. The focus of our work is on electing a credible and reliable coordinator securely by considering the nodal attributes.

There is a need for an algorithm to securely elect a reliable coordinator in critical applications like transaction management in blockchain [11–13].

3.Methods

Coordinator election voting methods are classified as extrema finding methods and preference-based methods. Nodes in distributed systems are identified using a unique Identity (ID). As discussed in section 2, the extrema finding algorithm elects a coordinator node based on the extreme ID value. A node with either minimum or maximum ID is elected as a coordinator [3].

In the preference-based election voting method, preferences are assigned to the nodes based on their nodal attributes. The highest preference node is elected as a coordinator. In existing algorithms, nodal attributes of the nodes are not taken into consideration [3, 29, 30, 31].

3.1Novel method for candidate selection

In SRCEA, candidate selection is done based on the nodal attributes maintained by the member nodes as a group view G_v in the system. Algorithm 1. Candidate_Selection shown below is used for forming the candidate set C. Abbreviations and Notations used in Candidate_Selection, and SRCEA are listed in *Appendix I*. The nodal attributes identified and maintained in the system designed for implementing SRCEA are joining timestamp, a distance of the node from the center of the network, and failure count [32]. The desirable value of all the attributes used for selecting the node as a candidate is the minimum value. Looking at this requirement, the min-heap data structures are used for storing the nodal attributes.

Algorithm 1. Candidate_Selection (P_d, P_{jts}, P_{fc})
1. C = Null
2. MyP_i = Node running Candidate_Selection
3. P_{idmin} = root (P_d)
4. $P_{ijtsmin}$ = root (P_{jts})
5. P_{ifcmin} = root (P_{fc})
6. C = {P_{idmin}, $P_{ijtsmin}$, P_{ifcmin}}

End

The nodal attributes joining timestamp, distance of the node from the center of network and failure count are stored in set P_{jts}, $P_{d,}$ and P_{fc} respectively. This set of attributes values are updated on both the membership event of join and leave on G_v. At any instance of time, the node with minimum attribute value can be retrieved by accessing the root of the

respective min-heap tree. A set of candidate nodes C is formed using a set of member nodes as shown in Equation 1. The number of candidate nodes selected is based on the number of nodal attributes being processed.

$$C = \exists\ i, j, k\ (P_i, P_j, P_k) \mid (P_i, P_j, P_k \varepsilon P)\ \&\ (P_i = P_{dmin}\ \&\ P_j = P_{jtsmin}\ \&\ P_k = P_{fcmin}) \quad (1)$$

Where, $i, j, k \varepsilon \{1...n\}$, P_i, P_j, P_k are node identities and P_{dmin}, P_{jtsmin}, P_{fcmin} are nodes identities with minimum distance, minimum joining timestamp, and minimum failure count respectively. The set of eligible candidate nodes C is formed in constant time *O(1).* The set is formed as $C = \{P_{idmin}, P_{ijtsmin}, P_{ifcmin}\}$ having one, two or three nodes, based on the attribute tree roots. In every election algorithm run, the candidate selection step takes constant time. There are three possible cases for the size of set C. If **$|C| = 1$** i.e., a single node is having a minimum value of all the attributes' then the second phase of the voting process is not carried out. The node selected in set C declares itself a coordinator by sending message I Am Coordinator (IAC). On receiving the IAC message, all member nodes can verify the coordinator node by using the group view G_v and send message M_v (verify message) to the newly selected coordinator node. For **$|C| > 1$,** the preference-based election voting process is carried out. Preference votes cast by each node can assign a priority to each of the nodal attributes. The elected coordinator node has the best minimum value of at least one nodal attribute.

3.2A secure and reliable coordinator election algorithm (SRCEA)

In a distributed network, the election process is carried out using message passing protocol and it is easy to modify a vote or flip a vote by modification attack in absence of security mechanisms. It is important to ensure the safety of the election process and hence the election messages which are exchanged. The confidentiality and integrity of the election messages need to be ensured. The loss occurring as a consequence of these attacks can be avoided if the confidentiality and the integrity of the election process are ensured. A security solution is needed for the safe termination of the election algorithm. As discussed in section 2, very few of the existing algorithms address the security challenges in election algorithms.

SRCEA is proposed for safely electing a reliable candidate as a coordinator in a peer-to-peer distributed system. While designing SRCEA, we have made some assumption like every node has a unique identifier assigned to it as P_i and the nodes and communication links are reliable (the nodes and communication links would not fail during the election process). SRCEA is designed as a preference-based algorithm since ID numbers alone do not signify any priority of the coordinator elected. Hence using nodes' ID alone may not result in the election of a durable or reliable coordinator. SRCEA assumes that all the member nodes of the system have agreed upon a secret key to be used for symmetric encryption. It ensures the confidentiality and integrity of the election vote messages. *Figure 1* and 2 shows the overview of SRCEA and the security mechanism of SRCEA respectively. As mentioned in section 1, SRCEA is a two-phase distributed coordinator election algorithm. During the first phase, candidate nodes are selected using the nodal attributes – a distance of the node from the center of the network, joining timestamp of the node, and failure count. The format of the election vote message is decided as per the number of nodes in set C. The formats of messages used in SRCEA are listed in section 3.21. The voting can be performed with prioritized votes to assign preferences to the candidate nodes. Weighted votes of one candidate can be compared with other candidate nodes' weighted votes and the elected coordinator node is verifiable by member nodes. As the elected candidate has stability it prevents the frequent re-election procedures thus saving cost and time involved therein.

Figure 1 gives the overview of the SRCEA. As shown in *Figure 1*, messages are sent and received by message service. The security mechanism of SRCEA ensures the confidentiality and integrity of every message received and sent. The security mechanism block is explained in detail in *Figure 2* which is responsible for encrypting messages and calculating the hash of the encrypted message. The integrity of the election vote message is verified by recalculating the hash of the received message and comparing it with the received hash code. Membership service receives the JOIN and LEAVE messages from member nodes. Whenever a node joins the system it sends a JOIN message and the LEAVE message is sent whenever the node leaves the system. Group view G_v is updated on every membership event. The candidate selection process is carried out using the current group view G_v. Min-heap of all the attributes is referred to form the set C. On selecting members for set C from set P, the format of the vote message is decided by preparing message service as per the

formats discussed in section 3.2.1. If |C| = 1, i.e., the node P_i present in C is having a minimum value for all the attributes. In this case, there is no need for the voting phase to be carried out and the node say P_i declares itself as a coordinator and asks to verify by sending a message M_{iac}. It sends the current timestamp and its id also, to avoid the replay attack. In case if the number of candidates is more than one, the voting process is carried out which is followed by the verification of the coordinator. The received vote messages are decrypted and verified for integrity by a security mechanism. The preference vote messages are used in voting. The weighted sum of votes is calculated as shown in Equation 2.

$$Cwvci = \sum_{j=1}^{3} j \times VCj \quad (2)$$

Where, Cwvci is the weighted vote value for node P_i, Vcj is total votes received with preference j where 1<= j <=3.

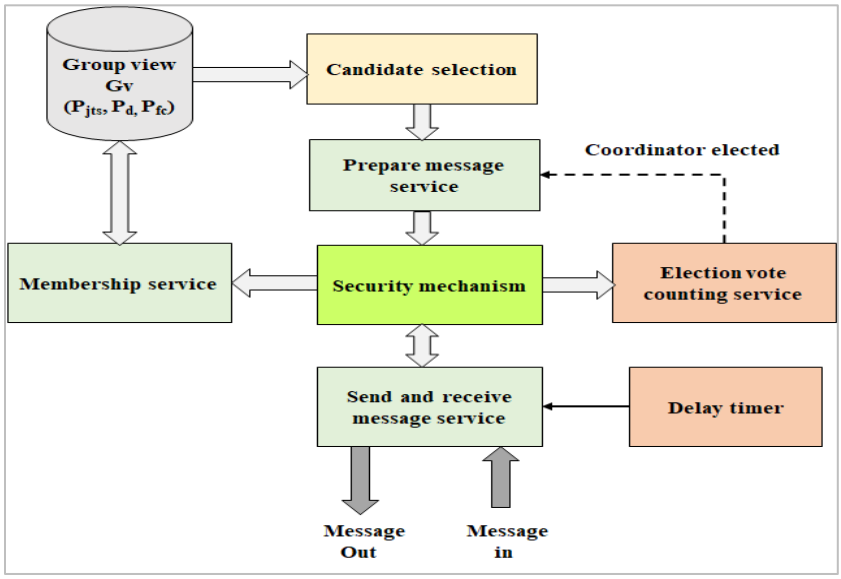

Figure 1 Overview of secure and reliable coordinator election algorithm (SRCEA)

Election vote counting service finds the candidate with maximum votes received and communicates the same to the prepare message service. In case if more than one candidate receives the maximum and the same number of votes then-candidate with the smallest ID among them is elected as coordinator. Once the coordinator is elected, it is verified by other candidate nodes and/or member nodes. Any of the member nodes can verify the eligibility of the coordinator and respond with VERIFIED message M_v to member nodes in the current G_v. All the messages including verification message are also hashed and encrypted in the similar manner as M_{iac}. After verification of coordinator node, it starts sending I Am Alive (IAA) message periodically. The timer is used to insert a fixed delay in message transmission.

As shown in *Figure 2*, the security mechanism of SRCEA ensures the confidentiality and integrity of the election messages. Symmetric key encryption technique along with the keyed hash function is used for encryption and hashing of messages. All the messages exchanged during the election process are encrypted using secret key K_{scr} and then hashed using Hash-based Message Authentication Code using Secure Hash algorithm (HMACSHA-256) [10] which generates a fixed size message digest. The encrypted and hashed message are concatenated with the current timestamp. The initialization vector E_{IV} used in the encryption and decryption process is updated for each election round. E_{IV} is calculated as shown in Equation 3.

$$E_{IV} = K_{scr} \oplus (n \times \text{height}(d_{minHeap})) \quad (3)$$

Where $d_{minheap}$ is the min-heap tree for distance attribute and n is the number of nodes in the system. Updating E_{IV} for each election round secures communication and election vote messages against ciphertext-only attack as the parameters used in calculating E_{IV} are known to member nodes only. Encryption followed by hashing makes it possible to authenticate the node. Election vote message M_{vote} is prepared as per formats specified in section 3.2.1 and encrypted using a secret key. The encryption process

is carried out with updated initialization vector E_{IV}. Encrypted message M is hashed with a secret key to generate HM which is appended to M along with the current timestamp. On receiving the election message, a hash of the received message is calculated again and the integrity of the message is ensured.

3.2.1SRCEA message formats

SRCEA is a distributed algorithm and executed using message passing protocol. The formats of the messages exchanged in SRCEA during election process are discussed below. The message M_{vote} is sent during voting process. It is prioritised vote message, where either two or three candidates are voted with priorities. Message M_{iac} is sent by the coordinator node once it is safely elected.

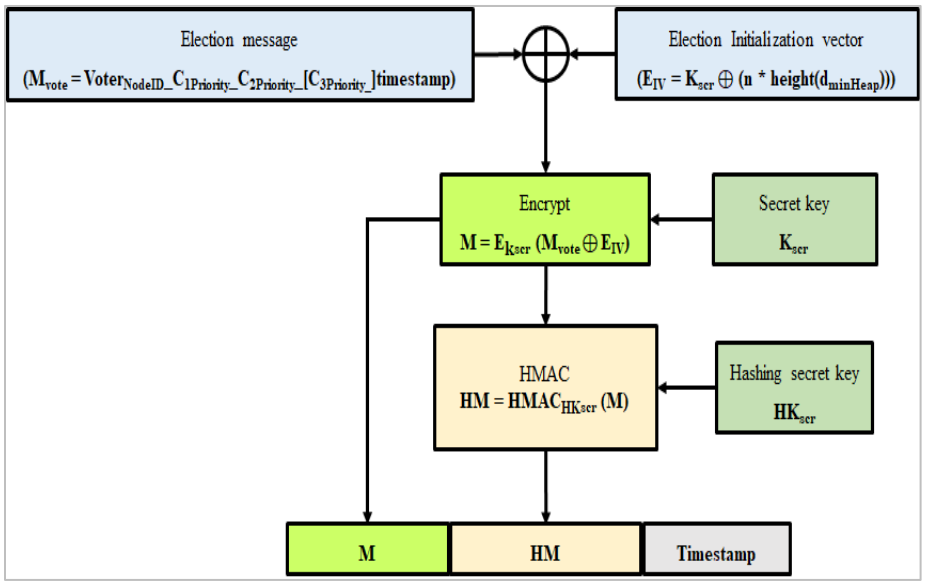

Figure 2 Security mechanism in secure and reliable coordinator election algorithm (SRCEA)

Vote message (M_{vote})
$\mathbf{M_{vote}} = P_{sid} \,||\, P_{i1} \,||\, P_{i2} \,||\, [P_{i3}] \,||$ timestamp

Set C i.e., the number of candidate nodes are selected based on number of the nodal attributes. M_{vote} is sent by member nodes in cases where $|C| > 1$. P_{sid} is the sender node ID and P_{ij} where $1 \leq j \geq 2$ or 3 based on the candidate set size. P_{i1} is the candidate node with first preference, P_{i2} is the candidate node with second preference and P_{i3} is the candidate node with third preference (in case if $|C| = 3$). This message is sent by member nodes to candidate nodes of set C.

IAC message (M_{iac})
$\mathbf{M_{iac}} = P_{cid} \,||$ IAC $||$ VERIFY $||$ timestamp

Once the election process is completed safely, a coordinator node is elected say P_{cid}. M_{iac} message is sent by P_{cid} to indicate that a coordinator node is elected and need to get verified by the member nodes.

IAA message (M_{iaa})
$\mathbf{M_{iaa}} = P_{cid} \,||$ IAA $||$ timestamp
Once elected, coordinator node P_{cid} start sending M_{iaa} to member nodes indicating the coordinator node is alive. The message M_{iaa} is sent periodically may be after every δ *milliseconds* to ensure that coordinator is not failed, crashed or left the system.

VERIFIED message (M_v)
$\mathbf{M_v} = P_{sid} \,||$ VERIFIED $||$ timestamp

On receiving message $M_{iac,}$ the coordinator node needs to get verified. The verification of coordinator is performed by other candidate nodes than coordinator. The candidate nodes refer to the weighted preference for verifying the current coordinator vote count. Once verified, candidate node sends message M_v to member nodes specifying coordinator is elected.

The proposed algorithm SRCEA is shown in algorithms 2. Set C is passed as an argument to the algorithm. MyP_i is the ID of the node running SRCEA. Algorithm elects and returns at most one node as a coordinator. Count of votes received for different priorities are stored in V_{ci} variable. $C_{wvc1,}$ C_{wvc2} and C_{wvc3} are used to store weighted vote count of candidates. Maximum of three candidate's votes is stored in MaxVote. M_{wv} holds the majority vote count

for current group member count n. SRCEA is executed by member nodes on identifying the failure of coordinator in the system.

4.Results

The performance measures for SRCEA and SPLEA are identified and analysed. The important parameters analysed are communication cost (message complexity - *comm_cost)*, computation cost (*comp_cost)* and storage cost (space complexity - *space_cost)*.

Algorithm 2. SRCEA(C)
Result: At most one node is elected as a coordinator
1.$MyP_i = P_{id}$ of the node executing SRCEA
2.V_{cj} = Vote Count for Priority j
3.$C_{wvc1} = 0$
4.$C_{wvc2} = 0$
5.$C_{wvc3} = 0$
6.MaxVote = 0
7.$M_{wv} = (n \times 3)/2$
8.**if** | C | = 1 **then**
9. **if** $MyP_i \epsilon$ C **then**
10. Create message M_{iac}
11. $M = E_{kscr} (M_{vote} \oplus E_{IV})$
12. $HM = HMAC_{HKscr} (M)$
13. Send message [M || HM || Timestamp]
14. Wait for δ ms to receive M_v
15. Decrypt M_v and check integrity by calculating hash
16. **if** Message is received safely **then**
17. **exit** (MyP_i)
18. **else**
19. **exit** (P_1)
20. **else**
21. Wait to receive M_{iac} message from coordinator for δ ms
22. Decrypt M_{iac} and check integrity by calculating hash
23. **endif**
24.**else**
25. Create M_{vote}
26. $M = E_{Kscr} (M_{vote} \oplus E_{IV})$
27. $HM = HMAC_{HKscr} (M)$
28. Send message [M || HM || Timestamp]
29. Wait for δ ms
30. **if** $MyP_i \epsilon$ C **then**
31. Decrypt received vote messages and check integrity
32. **for** i = 1 to 3 **do**
33. $Cwvci = \sum_{j=1}^{3} j * VCj$
34. MaxVote = Maximum (C_{wvc1}, C_{wvc2}, C_{wvc3})
35. **If** MaxVote >= M_{wv} **then**
36. Coordinator node P_c create, encrypt and sends M_{iac}
37. Wait to receive M_v message for δ ms
38. Decrypt M_{iac} and check integrity by calculating hash
39. **else**
40. **exit** ("Re-election")
41. **else**
42. Wait for δ ms to receive M_{iac} message
43. Verify coordinator node votes and create, encrypt and send M_v message
End

4.1Simulation environment

The distributed system application is designed for implementing the SRCEA. The computer system with i7 processors, 8 GB RAM, 1 TB HDD is used for experiments. *Comm_cost* and *comp_cost* is evaluated, averaged and logged for multiple executions. Comparison of communication cost with the SEFA and SPLEA algorithms is plotted.

4.2Result analysis

Communication cost and computation cost of SRCEA is analysed and presented. The communication cost or message complexity is calculated by analysing the number of messages exchanged for SRCEA and SPLEA to completion. *Figure 3* shows the comparison of communication cost of SRCEA, SEFA and SPLEA. The computation cost (time required for executing the algorithm) in milliseconds is analysed. The *Table 1* depicts the computation cost for different stages in SRCEA and total computation time.

It is observed that SRCEA elects coordinator with less communication cost as compared to SEFA and SPLEA. SEFA and SPLEA are round-based algorithms. Communication cost of the algorithm increases with the number of rounds. Hence if algorithm takes n rounds to execute where n is the number of member nodes then the communication cost is $O(n^2)$. Another factor that affects communication cost in SPLEA and SEFA is the number of messages exchanged for PKI. There is communication overhead of exchanging the public keys if asymmetric encryption mechanism is implemented. Whereas, SRCEA is not round-based algorithm and exchanges the number of messages which are proportional to the number member nodes in the system. Second important factor which is reducing the number of messages exchanged is use of symmetric key encryption mechanism which does not

need to any additional key(s) once member nodes are agreed over a secret key for secure communication.

4.2.1Communication cost

Figure 3 shows the graph for comparing the number of messages exchanged in the system to complete the election process (communication cost i.e., *comm_cost*) of SRCEA with SPLEA. The *comm_cost* for SEFA and SPLEA is same in best and worst case hence SPLEA algorithm *comm_cost* is plotted. The *comm_cost* for SRCEA is *O(n)* which is precisely *2*n* messages in best case and *5×n* messages in worst case. Whereas, the *comm_cost* for SPLEA is *O(n)* in best case and *O(L×n)* in worst case where n is number of nodes in hierarchy and *L* is number of election rounds. Typically, best case is resulted when all nodes are one hop apart and worst case occurs when every node is *L* hops away from each other. The graph in *Figure 3* shows the communication cost for the different sizes of the system i.e., number of nodes ranging from 50 to 500. The numbers of messages exchanged in best and worst cases are plotted using a logarithmic scale.

- SRCEA improves communication cost by 97% as compared to SPLEA.
- Another factor affecting the communication cost is the security mechanism used. SRCEA uses symmetric encryption technique which reduces the exchange of messages considerably as there is no messages exchanged for additional keys (public keys) other than secret key.
- It is observed that the numbers of messages exchanged are more in SPLEA as it is round-based election algorithm as compared to SRCEA. Round based algorithm message count increases considerably with the increased number of rounds.

4.2.2Computation cost

The computation cost i.e., *comp_cost* for the SEFA, SPLEA and SRCEA are identified, evaluated and analysed for encryption and hashing security mechanisms. *Table 1* depicts the computation cost of SEFA, SPLEA and SRCEA in milliseconds. As discussed in section 4, SRCEA is proposed to with symmetric encryption mechanism whereas SEFA and SPLEA are designed with PKI. Hence the asymmetric encryption algorithms are used for analysing the *comp_cost* of SEFA and SPLEA. SEFA-1, SPLEA-1 and SEFA-2, SPLEA-2 notations denote the *comp_cost* evaluated using 1024-bit Rivest–Shamir–Adleman (RSA) algorithm and 384-bit Elliptic Curve Cryptography (ECC) algorithms respectively. Similarly, SRCEA-1, SRCEA-2 and SRCEA-3 notations indicate the *comp_cost* evaluated using AES algorithm with 128-bit, 192-bit and 256-bit key respectively. The hashing technique used for ensuring the message integrity in SPLEA is MD5 whereas HMACSHA-256 technique is used in SRCEA. The *comp_cost* is calculated for different stages of election messages. The *comp_cost* for hashing and encrypting the election messages along with decrypting and checking the integrity of those messages is logged and analysed. *Table 1* shows that the *comp_cost* of implementing the coordinator election algorithm using PKI is more as compared to the symmetric encryption technique.

- SRCEA encryption and decryption time is considerably less as compared to SPLEA and SEFA as it uses symmetric encryption which does not involve modular exponentiation operations which are computation costly.
- SRCEA implemented using symmetric encryption algorithm (for example AES-256) provides high security level as compared to the one using PKI implemented using RSA or ECC algorithms [20].
- SRCEA *comp_cost* for HMACSHA-256 is more as compared to MD5 hashing technique as it uses keyed hash function to calculate the hash of 256 bits.

Typically, According to National Institute of Standards and Technology (NIST) [20], high level of security can be provided using symmetric encryption techniques with smaller key size as compared to PKI with large key sizes. *Table 1* shows that the total *comp_cost* of SRCEA is significantly less as compared to SEFA and SPLEA. Overall computation cost of SRCEA is comparatively less as it does not use any of the modular exponentiation operations to hash or encrypt the messages.

4.2.3Storage cost

The important factor affecting storage cost or space complexity *space_cost* of SRCEA is the number of nodal attributes considered. In proposed algorithm three nodal attributes are considered hence minimum 3×n words space is needed for SRCEA in view of storing the group view G_v. Another factor affecting the space complexity is space utilised by secret key(s) of encryption algorithm and hashing function.

- Space complexity of SEFA and SPLEA is more comparatively as these algorithms uses PKI which need to store public keys of member nodes along node's own secret key (private key) whereas SRCEA uses only two secret keys (encryption secret key and hashing secret key).
- In general, space complexity of SRCEA would be *O(m×n)* where m is the number of nodal attributes considered and n is number of nodes in the system.

- Storage cost of SRCEA depends on number of nodes and the number of attributes it considers whereas in SEFA and SPLEA it depends on the number of nodes in the system.
- Total storage space needed by SRCEA is less as compared to SEFA and SPLEA as there is no need of storing public and private keys in every node.

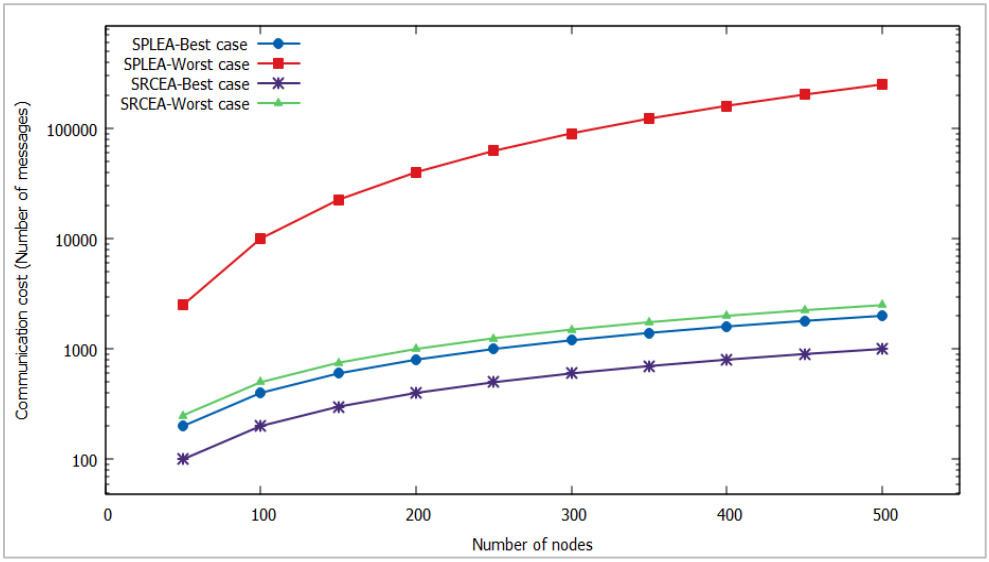

Figure 3 Comparison of communication cost

Table 1 Computation cost (milliseconds)

Algorithm	Hashing time (ms)	Encryption time (ms)	Decryption time (ms)	Integrity check time (ms)	Total computation time (ms)
SEFA -1	16.7	635.03	198.63	17.04	867.4
SEFA - 2	18	324.97	157.2	19	519.17
SPLEA - 1	16.92	675	214.8	16.08	922.8
SPLEA - 2	19.24	368.96	164	18.07	570.27
SRCEA -1	12.67	13	2.4	18.02	46.09
SRCEA -2	12.2	14.6	3	13	42.8
SRCEA -3	13.02	16.3	4.9	12.04	46.26

5.Discussion

Coordinator election algorithms in the distributed system are vulnerable to security attacks as message passing protocol is used in the election process [7]. Existing coordinator election algorithms are rarely addressing the security vulnerabilities in the election process. When exploited, security vulnerabilities in the election process may cause a denial of voting attack and an impersonation attack. These attacks critically affect the termination property of the election algorithm and as a result the same, the liveness property of the application would not be satisfied. SRCEA is designed for selecting reliable candidates and safely electing a candidate as a coordinator in the distributed system. The algorithm is designed to tolerate the security attacks like denial of voting and impersonation thus ensuring the safety and liveness of the algorithm. The communication cost, computation cost, and storage cost of SRCEA are analyzed and compared with existing secure coordinator election algorithms. SRCEA elects a reliable candidate as a coordinator safely in *O(n)* communication cost and with notable less computation cost.

Vulnerabilities in the coordinator election algorithm include insecure communication, faults, and failures of nodes and communication channels. SRCEA assumes that the nodes and communication links are reliable. Another limitation of SRCEA is the need for additional storage space to store the attributes of the nodes. Hence the space complexity of SRCEA depends on the number of nodes and the attributes it considers for candidates' selection. It is designed to

address the issue of insecure communication during the election process, ensuring confidentiality and integrity of the election messages.

SRCEA can be implemented in distributed system applications with nodes that have the attributes associated with them. Nodal attributes help out in electing the reliable candidate as a coordinator. For example, considering a node's battery life as an attribute while electing a coordinator in the mobile ad-hoc network would elect an efficient candidate as a coordinator. If extrema finding method is used in the mobile ad-hoc network for electing a coordinator then it may elect an inefficient node as a coordinator. Because mobile nodes with the highest ID may have the lowest battery life as compared to the remaining nodes in the network. The majority of the existing coordinator election algorithms are not taking into consideration the nodal attributes before electing a node as coordinator.

This may result in the election of the incompetent node as coordinator and early failure of the same. It is observed that SRCEA securely elects the coordinator with considerably less communication cost and computation cost.

To summarise, symmetric encryption and keyed hashing used in SRCEA make it security attacks resistant and ensure the consistency of the election messages. The initialization vector E_{IV} in the SRCEA security mechanism is updated for every election, which takes place and ensures forward and backward secrecy of the messages exchanged in the system.

Hence the SRCEA security mechanism makes the election process more secure even without a need for sharing a new secret key for every election process. The communication cost of SRCEA is 97% less than SEFA and SPLEA. The important factor which is affecting the communication of SRCEA is the use of symmetric encryption technique for encryption that reduces the number of keys exchanged before actual communication in the system.

6.Conclusion and future work

A novel method for coordinator election in a distributed network is designed. In this method, the eligible candidates are identified and then the election process is carried out using the preference-based voting. The security mechanism is implemented in SRCEA to ensure the confidentiality and integrity of the election messages. The performance measures for the SRCEA are identified, evaluated, and compared with the existing secure coordinator election algorithms. SCREA elects a reliable coordinator securely by taking into consideration the nodal attributes. SRCEA elects a reliable coordinator with *O(n)* communication complexity and space complexity with considerably less computation cost.

In this work, we have assumed that the nodes and the communication links in the system are reliable and do not fail during the election process. Our future work involves addressing the vulnerabilities in election algorithms and designing a fault-tolerant coordinator election algorithm.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Boneh D, Eskandarian S, Hanzlik L, Greco N. Single secret leader election. In proceedings of the 2nd ACM conference on advances in financial technologies 2020 (pp. 12-24).

[2] Brooker M. Leader election in distributed systems. AWS, Leader Election in Distributed Systems; 2019.

[3] Garcia-molina H. Elections in a distributed computing system. IEEE Transactions on Computers. 1982; 31(1):48-59.

[4] Zhang R, Jacquemot B, Bakirci K, Bartholme S, Kaempf K, Freydt B, et al. Leader selection in vehicular ad-hoc networks: a proactive approach. In vehicular technology conference 2020 (pp. 1-5). IEEE.

[5] Yang Z, Xiao J. Alibaba group holding ltd, assignee. Efficient, time-based leader node election in a distributed computing system. United States patent US 10,534,634. 2020.

[6] Casteigts A, Métivier Y, Robson JM, Zemmari A. Deterministic leader election takes $$\theta (D+\log n) $$ Θ (D+ log n) bit rounds. Algorithmica. 2019; 81(5):1901-20.

[7] Supase SS, Ingle RB. Are coordinator election algorithms in distributed systems vulnerable?. In international conference on computing, communication and networking technologies 2020 (pp. 1-5). IEEE.

[8] Lakhani G, Kothari A. Coordinator controller election algorithm to provide failsafe through load balancing in distributed SDN control plane. In international conference on computing science, communication and security 2020 (pp. 234-50). Springer, Singapore.

[9] Jiang F, Cheng Y, Dong C, Yu E. A novel weight-based leader election approach for split brain in distributed system. In conference series: materials science and engineering 2020 (pp. 1-5). IOP Publishing.

[10] Stallings W. Cryptography and network security, 4/E. Pearson Education India; 2006.
[11] Guerraoui R, Pavlovic M, Seredinschi DA. Blockchain protocols: the adversary is in the details. In symposium on foundations and applications of blockchain 2018 (pp. 24-30).
[12] Han R, Yu J, Lin H, Chen S, Esteves-veríssimo P. On the security and performance of blockchain sharding. Cryptology ePrint Archive. 2021:1-15.
[13] Kuraganti CK, Robert BP, Gurrala G, Joglekar A, Puthuparambil AB, Sundaresan R, et al. A distributed hierarchy framework for enhancing cyber security of control center applications. arXiv preprint arXiv:2010.04955. 2020.
[14] Horowitz E, Sahni S, Anderson-freed S. Fundamentals of data structures in C. WH Freeman & Co.; 1992.
[15] Vasudevan S, Decleene B, Kurose J, Towsley D. Secure leader election in wireless ad hoc networks. UMass Computer Science Technical Report. 2001:1-50.
[16] Abdullah M, Al-kohali I, Othman M. An adaptive bully algorithm for leader elections in distributed systems. In international conference on parallel computing technologies 2019 (pp. 373-84). Springer, Cham.
[17] Effatparvar M, Yazdani N, Effatparvar M, Dadlani A, Khonsari A. Improved algorithms for leader election in distributed systems. In international conference on computer engineering and technology 2010 (pp. 2-10). IEEE.
[18] Sidik B, Puzis R, Zilberman P, Elovici Y. Pale: time bounded practical agile leader election. IEEE Transactions on Parallel and Distributed Systems. 2019; 31(2):470-85.
[19] Bellare M, Rogaway P, Wagner D. A conventional authenticated-encryption mode. Manuscript, 2003:1-14.
[20] Al-mamun A, Rahman S, Shaon TA, Hossain MA. Security analysis of AES and enhancing its security by modifying S-box with an additional byte. International Journal of Computer Networks & Communications (IJCNC). 2017; 9(2): 69-88.
[21] Jackson SC. Models of leader elections and their applications. PhD Thesis. Missouri University of Science and Technology. Missouri. 2016.
[22] Madisetti VK, Panda S. A dynamic leader election algorithm for decentralized networks. Journal of Transportation Technologies. 2021; 11(3):404-11.
[23] Chaparala P, Atmakuri AR, Rao SS. 3-phase leader election algorithm for distributed systems. In international conference on computing methodologies and communication 2019 (pp. 898-904). IEEE.
[24] Rahman MU. Leader election in the internet of things: challenges and opportunities. arXiv preprint arXiv:1911.00759. 2019.
[25] Subramanian ER, Sri GB, Sayee SAS, Aishwarya V, Balaji N, Umamakeswari A. A novel hypergraph-based leader election algorithm for distributed systems. In innovations in computer science and engineering 2020 (pp. 437-45). Springer, Singapore.
[26] Mohammed N, Otrok H, Wang L, Debbabi M, Bhattacharya P. Mechanism design-based secure leader election model for intrusion detection in MANET. IEEE Transactions on Dependable and Secure Computing. 2009; 8(1):89-103.
[27] Barki A, Gouget A, Toulemonde A. Revisiting security properties in single secret leader election. In international conference on blockchain and cryptocurrency 2021 (pp. 1-3). IEEE.
[28] Byrenheid M, Strufe T, Roos S. Attack resistant leader election in social overlay networks by leveraging local voting. In proceedings of the international conference on distributed computing and networking 2020 (pp. 1-10).
[29] Datta AK, Devismes S, Larmore LL, Villain V. Self-stabilizing weak leader election in anonymous trees using constant memory per edge. Parallel Processing Letters. 2017; 27(2).
[30] Parhami B. Voting algorithms. IEEE Transactions on Reliability. 1994; 43(4):617-29.
[31] Franchi A, Giordano PR. Online leader selection for improved collective tracking and formation maintenance. IEEE Transactions on Control of Network Systems. 2016; 5(1):3-13.
[32] Supase S, Ingle R. Method and system for election of a coordinator node in a distributed network(https://ipindiaservices.gov.in/PatentSearch/PatentSearch/ViewApplicationStatus). Indian Patent Number 360624. 2021 (pp. 1-24).

Shital Subhashchandra Supase is a PhD research scholar in Pune Institute of Computer Technology, Pune under Savitribai Phule Pune University. She has completed ME (Computer Engineering) and BE in Computer Science and Engineering from Pune Institute of Computer Technology in 2011 and Government College of Engineering, Amravati in 2001 respectively. Her research area is Distributed Systems and Security.
Email: sssupase@gmail.com

Dr. Rajesh Baliram Ingle is Professor in Department of Computer Engineering at Pune Institute of Computer Technology. He has received a PhD in Computer Science and engineering from Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, Powai, Mumbai. He is also working as an Adjunct Professor at Department of Computer Engineering, Government College of Engineering Pune. He has also received 2014, IEEE outstanding branch counsellor award from IEEE, USA.
Email: rbingle@pict.edu

Appendix I

S. No.	Notation	Meaning
1	AE	Authenticated Encryption
2	AEAD	Authenticated-Encryption with Associated-Data
3	AES	Advanced Encryption Standard
4	C	Set of candidate nodes i.e., C = {P_{idmin}, $P_{ijtsmin}$, P_{ifcmin}}
5	$d_{min\text{-}heap}$	Min-heap for P_d
6	ECC	Elliptic Curve Cryptography
7	E_{IV}	Initialization vector
8	FHE	Fully Homomorphic Encryption
9	G_v	Current group view (set of member nodes)
10	HMACSHA	Hash-based Message Authentication Code using Secure Hash algorithm
11	IAA	I Am Alive
12	IAC	I Am a Coordinator
13	K_{HKscr}	Secret key for hashing algorithm
14	K_{scr}	Secret key for symmetric encryption algorithm
15	MD5	Message Digest
16	n	Number of nodes
17	NIST	National Institute of Standards and Technology
18	P	Set of member nodes {P_1, P_2, …, P_n}
19	PALE	Practical Agile Leader Election
20	PKI	Public Key Infrastructure
21	P_d	Set of distance attribute {P_{1d}, P_{2d}, P_{3d}, …, P_{nd}} for G_v
22	P_{fc}	Set of failure count {P_{1fc}, P_{2fc}, P_{3fc}, …, P_{nfc}} for G_v
23	P_i	ID of i^{th} node
24	P_{id}	Distance of P_i from the center of network
25	P_{idmin}	ID of minimum distance node from the center of network among G_v
26	P_{ifc}	Failure count of P_i
27	P_{ifcmin}	ID of node failing minimum number of times among G_v
28	P_{ijts}	Joining time stamp of P_i
29	$P_{ijtsmin}$	ID of node with minimum joining time stamp among G_v
30	P_{jts}	Set of joining time stamp attribute {P_{1jts}, P_{2jts}, P_{3jts}, …, P_{njts}} for G_v
31	RSA	Rivest–Shamir–Adleman
32	SEFA	Secure Extrema Finding Algorithm
33	SCREA	Secure and Reliable Coordinator Election Algorithm
34	SPLEA	Secure Preference-based Leader Election Algorithm
35	SSLE	Single Secret Leader Election