Secure Blockchains from Byzantine Attack using Lamport-Shostak-Pease Algorithm in CR Networks

---

Abstract— **In the past couple of years, the research on the Byzantine attack and its defense strategies has gained the worldwide increasing attention. In this paper, we present a secure blockchain protocol to escape from the Byzantine attack in the cognitive radio networks. This protocol is implemented using the Lamport-Shostak-Pease algorithm. A reliable distributed computing system must be able to handle the faulty components to deliver the error less performance. These faulty components send the conflicting information to the other parts of the system. This problem is expressed as similar to the Byzantine Generals problem. To achieve a reliable system it is necessary to identify and overlook such faulty components. Also, it requires all fault-free processors to agree on a common value, even with some faulty components. This is called as the Byzantine Agreement. Here we have addressed this Byzantine General problem to provide a reliable and secure distributed system using the the Lamport-Shostak-Pease algorithm. It solves the Byzantine Generals problem for $n \geq 3m+1$ processors in the presence of m faulty processors**

Index Terms—**Blockchain Technology, Byzantine attack, Security, Lamport-Shostak-Pease Algorithm, Secure Blockchains and Cognitive Radio.**

I. INTRODUCTION

Unanimity in an unreliable distributed system is still far from being well understood. The major task is to circumvent errors without losing unanimity. This can be achieved if all the reliable members of the system agree upon the content of the messages in the system, especially those messages corresponding to the faulty parts of the system, even where the faulty parts cannot be uniquely identified. The assumption is that a faulty processor can do whatever it likes. Thus, a faulty processor can behave very strangely: It can alter the information relayed through itself; it can block such information from being relayed; it can incorrectly reroute the information, and in the worst case, it can send conflicting information to different members of the system. Some of the processors may consider a faulty processor to be a reliable one and a reliable processor to be faulty. Obviously, there is a limit to the number of faulty processors a system can tolerate. The problem of achieving unanimity is further complicated by the following compound question: Under what conditions does unanimity remain valid and what is the system's unanimity threshold'?

In the general case one does not know which processors are faulty. Moreover, in most cases one will never be able to know. To understand the reason for this, picture yourself as a processor in a distributed system that receives a message from a processor 2. Assume that you want to make sure that z is reliable. So, you inquire what are the messages ths rest of the system received. You find out that the message you have received differs from all the rest. Is z a faulty processor? The answer is not necessarily positive. The possibility that the only reliable processors in the system are you and z always exists.

In the distributed system design, the reliability is an important parameter to be considered. This parameter tells the ability of the system to work efficiently even in case of failures in the different parts of the system. There are different types of failures in the components. One of them is the crash failures in which the failed component has no response. And in some other failures, they send out the conflicting messages. These failures create a situation that is similar to the Byzantine Generals Problem.

Imagine, for example, that the source process is the only faulty process. It tells half the processes that the value of their order is 0, and the other half that their value is 1. After receiving the order from the source process, the remaining processes have to agree on a value that they will all decide on. The processes could quickly poll one another to see what value they received from the source process. In this scenario, imagine the decision algorithm of a process that receives an initial message of 0 from the source process, but sees that one of the other processes says that the correct value is 1. Given the conflict, the process knows that either the source process is faulty, having given different values to two different peers, or the peer is faulty, and is lying about the value it received from the source process. It's fine to reach the conclusion that someone is lying, but making a final decision on who is the traitor seems to be an insurmountable problem. And in fact, it can be proven that it is impossible to decide in some cases. The classic example used to show this is when there are only three processes: One source process and two peer processes.

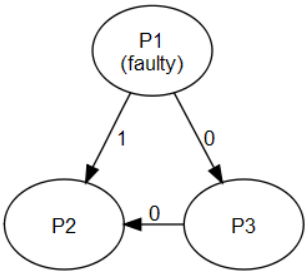


Fig 1.1: Byzantine Generals Problem: Source process is faulty

In the configurations in Figures 1.1 and 1.2, the peer processes attempt to reach consensus by sending each other their proposed value after receiving it from the source process. In Figure 1.1, the source process (P_1) is faulty, sending two different values to the peers. In Figure 1.2, P_3 is faulty, sending an incorrect value to the peer.

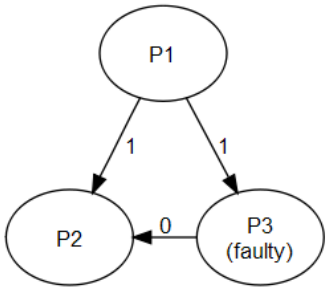


Fig 1.1: Byzantine Generals Problem: Peer process P3 is faulty

You can see the difficulty P_2 faces in this situation. Regardless of which configuration it is in, the incoming data is the same. P_2 has no way to distinguish between the two configurations, and no way to know which of the two other processes to trust.

This situation doesn't necessarily get better just by throwing more nonfaulty processes at the problem. A naïve algorithm (as in Figures 1 and 2) would have each process tell every other process what it received from P_1. A process would decide the correct decision by simple majority.

It's relatively easy to show that, regardless of how many processes are in the system. A subversive source process with one collaborator can cause half the processes to choose to attack, and half the processes to retreat, leading to maximum confusion.

One of the possible solutions is to send oral messages (which are forgeable). In this solution to cope-up with m-traitors, it must be needed with a minimum of (3m+1) generals. Following assumptions are to be considered to use the oral messages.

Assumption 1: Every message is delivered correctly.

Assumption 2: The receiver can identify the sender upon reception of every message.

Assumption 3: The receiver can detect if there's no message.

The first two assumptions assure that there's no interference from a traitor. Assumption 3 will foil a traitor who is silent. In this method, also there's a possibility that the traitor can lie which makes the solution to be very difficult. To overcome this we have another solution which is to use un-forgeable(signed) messages. In this method in addition to the above three assumptions, we have two more assumptions:

Assumption 4: Signature is not forgeable.

Assumption 5: Everyone can verify the signature whether it is authentic or not.

In this approach, the commander sends a signed message to all the lieutenants. Then the lieutenant adds his signature and sends it to the other lieutenants, who add their signatures and send it to others, and so on. This solution is possible for any number of generals and possible faulters.

Now a days, the blockchain technology is most widely used for enabling and securing spectrum sharing in cognitive radio networks(CRNs). The spectrum sharing mechanisms have the more importance due to the needs related to increasing spectrum usage. This protocol is also widely used in virtual currency transactions. In such applications, it is necessary to provide the secure and reliable spectrum sharing mechanism. Here, blockchain is considered as the decentralized database technology in which the owner of the data maintains control.

As a general-purpose database technology, it can be applied to virtually any business context. But it cannot be a cost effective solution in all types of businesses. Following features are to be considered as the requirements before applying it to the any business application--decentralization, transparency, immutability, availability and security.

II. LITERATURE REVIEW

In the paper[1] by Jun Du, Xiang et al., the problem of cooperative spectrum sensing (CSS) in censoring-enabled CRNs is explained in detail. An optimization problem has been formulated to improve the performance of cooperative spectrum sensing in the censoring-enabled the Cognitive Radio Networks and developed an expectation maximization based algorithm to solve it, where the presences of primary user and the reliabilities of each secondary user can be jointly estimated. The proposed robust CSS scheme performs better than the previous reputation-based approaches.

In the paper[2], Byzantine Agreement model for Intrusion Detection, Prevention & Counter-measure Systems(IPS) has been proposed by C. Fernando. The current IPSs have the drawbacks like inability to survive failures and malicious attacks. A Secure Architecture and Fault-Resilient Engine (SAFE) is developed to solve the Byzantine General's Problem. Byzantine Agreement Protocols will be used to achieve consensus about which nodes have been compromised or failed, with a series of synchronized, secure rounds of message exchanges. Once a consensus has been reached, the offending nodes can be isolated and countermeasure actions can be initiated by the system.

The Byzantine attack in cooperative spectrum sensing (CSS), also known as the spectrum sensing data falsification (SSDF) attack in the literature, is one of the key adversaries to the success of cognitive radio networks (CRNs)[3]. In this

paper, a comprehensive survey and tutorial on the recent advances in the Byzantine attack and defense for CSS has been provided. Also, it has been proposed that a taxonomy of the existing Byzantine attack behaviours and elaborate on the corresponding attack parameters, which determine where, who, how, and when to launch attacks. Also, the author has analyzed the spear-and-shield relation between Byzantine attack and defense from an interactive game-theoretical perspective.

In the paper[4], Collaborative (or distributed) spectrum sensing has been shown to have various advantages in terms of spectrum utilization and robustness in cognitive radio networks (CRNs). The data fusion scheme is a key component of collaborative spectrum sensing. We have recently analyzed the problem of Byzantine attacks in CRNs, where malicious users send false sensing data to the fusion center (FC) leading to an increased probability of spectrum sensing error. Also the author has proposed a novel and easy to implement technique to counter Byzantine attacks in CRNs. In this approach, the FC identifies the attackers and removes them from the data fusion process. The analysis indicates that the proposed scheme is robust against Byzantine attacks and can successfully remove the Byzantines in a short time-span.

In the paper[5], it has been proposed that a novel defense reference whichjointly exploits the cognitive process of spectrum sensing and spectrum access in a closed-loop manner, to provide the defense scheme a solid basis without requiring any prior knowledge. Moreover, this paper analyzes the proposed reference's favourable reliability and high robustness over the state-of-the-art references, from two perspectives of spectrum sensing performance and the capability of identifying malicious sensors, respectively. Next, we design an optimal cooperative spectrum sensing scheme based on the proposed defense reference. Remarkably, from an information theoretic perspective, it is observed that based on the proposed reference, the information value of falsified reports is also exploited to further improve the global sensing performance. Furthermore, numerical simulations verify the proposed scheme's favourable performance, even in critical cases when malicious sensors are in majority.

Spectrum sensing plays an important role in improving the spectrum utilization in cognitive radio. However, the existence of Byzantine attackers impairs the overall performance of Cognitive Radio Networks (CRNs) severely, no matter in detection accuracy, throughput or energy efficiency and so on. Therefore, it is vital to identify attackers and decrease their abominable influences. In the paper[6] by Juan Sheng et al., a novel attacker-identification algorithm with dynamic attack probability is proposed, which can enhance the overall detection probability markedly and effectively reduce the false alarm probability. Meanwhile, a weight coefficient is explored to energy detection (ED) to promote the robustness of the system. Simulation results shows promising performance of proposed algorithm compared to previous works.

Cognitive radio (CR) is a revolutionary paradigm to solve the spectrum scarcity problem in wireless networks. In cognitive radio networks (CRNs), cooperative spectrum sensing is regarded as a promising approach method to significantly improve the performance of spectrum sensing, but it can be threatened by Byzantine attack. The existing defense references have focused on how to mitigate the negative effect of Byzantine attack, but with some strong assumptions, such as the attackers are in minority and/or a trusted node exists for data fusion. This observation motivates us to comprehensively analyze strategies of Byzantine attack and the fusion center (FC) in the absence of these restrictions. To be specific, it has been considered a generic Byzantine attack model by analyzing sophisticated malicious behaviours, which goes beyond the existing models for its generalization[7]. Under this generalized attack model, we derive the condition which makes the FC blind from malicious perspective. On this basis, the optimal attack strategy to maximize Bayes risk is analyzed, respectively, in the case of the unknown and known fusion rule. Further, we extend our analysis to the scenario where the FC has the knowledge of the attack strategy by an estimation algorithm and adopts the optimal fusion rule. Thus, it is also given that the closed form expression, in terms of the optimal attack strategy under different scenarios, sequentially.

In the paper[8], two new counterattacks are proposed to combat Byzantine attacks comprising coalition head and cognitive radio as attackers which target to reduce the number of available channels for sensing in distributed multi-channel cooperative spectrum sensing. In the proposed counterattack for coalition head attack, by using statistical properties of the exchanged SNRs in coalitions, the probability of attack is derived and a new selection formula for coalition head is proposed. In the second proposed counterattack for multi-channel Byzantine attack, the probability of available channel detection (the counterpart of false alarm probability) in the presence of Byzantine attackers is first formulated. Since in practice the coalition heads have no information of the presence of Byzantine attacks, predicting such information is a very challenging issue when the attack takes place. Then, the probability of each cognitive radio changing its local decisions and become Byzantine attacker, called probability of attack, is derived. By applying such probability in the probability of available channel detection, an iterative algorithm is proposed. Based on the actions of the cognitive radios, after each round of the algorithm, the probability of attack is updated. If Byzantine attackers continue attacking the system, their contribution to their associated coalitions decreases and they will be blocked out of coalitions. Simulation results show that the proposed counterattacks can remarkably eliminate Byzantine attackers in the cognitive radio network.

In the paper[9], it was aimed at the problem of Byzantine Attack in secure network communication, the network security assumptions which may contain Byzantine attacks are analyzed and described firstly. Then a secure random network coding model is proposed for resisting the Byzantine attacks where the CBC (Cipher Block Chaining) technology is combined with random network coding, we prove the correctness and security also, finally realize safety codes.

III. LAMPORT-SHOSTAK-PEASE ALGORITHM

A commanding general must send an order to his n-1 lieutenant generals such that:
IC 1: All loyal lieutenants obey the same order.
IC 2: If the commanding general is loyal, then every loyal lieutenant obeys the order he sends.
Here, IC 1 and IC 2 are known as the interactive consistency conditions. In this paper, it has been proposed that the solution for the Byzantine Generals problem using the Oral Messages algorithm(Unsigned messages)[10].

Unsigned Messages algorithm(UM)

This algorithm is designed for one commander with (n-1) lieutenants based model. It is the recursive algorithm with m-number of traitors/faulters where m is the non-negative integer.
Require default value $\mathbf{v_{def}}$ if traitorous commander does not send a message
Define function **majority(v_1,...,v_{n-1}) = v** if a majority of the values $\mathbf{v_i = v}$.

Algorithm UM(n, 0):

Step1: The commander sends **v** to all the lieutenants.
Step2: All lieutenants use the value **v** received from the commander or $\mathbf{v_{def}}$ if nothing is received.

Algorithm UM(n, m), m>0:

Step1: The commander sends value **v** to all the lieutenants.
Step2: For each lieutenant$_i$, the commander sends value v_i to the lieutenant$_i$. In case of no message from the commander the value '$\mathbf{v_{def}}$' is considered. Now lieutenant$_i$ acts as the new commander to perform UM(m-1) with (n-2) remaining lieutenants.
Step3: For each i & each j≠i,
let $\mathbf{v_j}$ = value lieutenant$_i$ received from lieutenant$_j$ in step (2) or $\mathbf{v_{def}}$ if no value received. Now each lieutenant computes the majority values from (v_1,....., $v_{(n-1)}$) to take the decision.

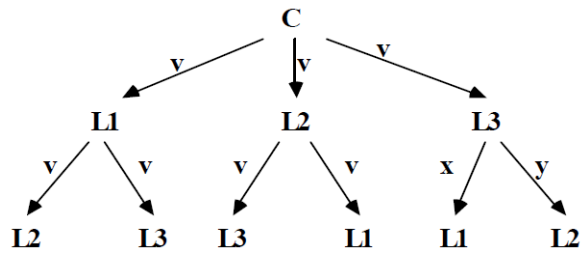
Complexity calculations of the algorithm:

In the first iteration, applying **UM(n, m)** causes the issuing of **(n - 1)** messages. In the second iteration, the unsigned message function **UM(n-1, m-1)** causes the issuing of **(n – 2)** messages etc.
Stage1: **(n - 1)** messages
Stage2: **(n - 1)*(n - 2)** messages
.....
Stage(m+1): **(n - 1)(n - 2). . . .(n - (m + 1))** messages
Total number of messages are $\mathbf{O(n^{m+1})}$.

IV. RESULTS

There exists no solution with fewer than **3m+1** generals can cope with **m** traitors. Also the solution becomes impossible with < (*m*+1) message exchanges. Consider n=4 with 1-traitor the following results can be obtained with UM(4,1).

Case I: L3 is a traitor

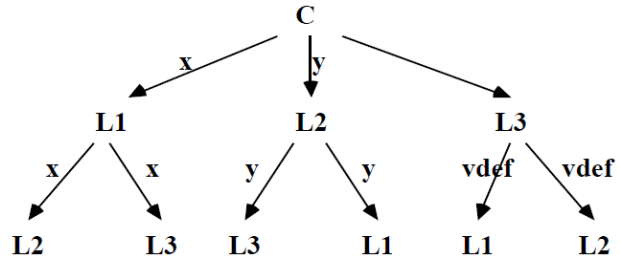


Stage1 results:
L1: v1 = v
L2: v2 = v
L3: v3 = v

Stage2 results:
L1: v1 = v, v2 = v, v3 = x
L2: v1 = v, v2 = v, v3 = y
L3: v1 = v, v2 = v, v3 = v

Majority calculation results:
L1: Mayority(v,v,x)=v
L2: Mayority(v,v,y)=v
L3: Mayority(v,v,v)=v

Case II: C is a traitor



Stage1 results:
L1: v1 = x
L2: v2 = y
L3: v3 = v_{def}

Stage2 results:
L1: v1 = x, v2 = y, v3 = v_{def}
L2: v1 = x, v2 = y, v3 = v_{def}
L3: v1 = x, v2 = y, v3 = v_{def}

The three loyal lieutenants receive the same value majority(x,y,v_{def}).

V. CONCLUSION & FUTURE SCOPE

In this paper, it has been described about the Byzantine Generals problem. Also we have defined the conditions for the solution of this problem. The solution has been presented using Lamport, Shostak and Pease algorithm. The algorithm has been designed and simulated for the solution of Byzantine Generals Problem. The implemented design leads to the

following conclusions. More recent algorithms have lower complexity, but in principle none can better the *m*+1 rounds taken by this algorithm. Every solution to BG requires the number of faulty processes to be less than a third of the total. Also, we have presented the secure method to escape from the Byzantine attack.

REFERENCES

[1] J. Du, *et al*., "A byzantine attack defender for censoring-enabled cognitive radio networks," *International Conference on Wireless Communications and Signal Processing*, Nanjing, 2015, pp. 01 - 06.

[2] Fernando C, "Using Byzantine Agreement in the Design Of IPS Systems," *2007 IEEE International Performance, Computing, and Communications Conference*, New Orleans, LA, 2007, pp. 528-537.

[3] Linyuan Zhang, Guoru Ding, et al., "Byzantine Attack and Defense in Cognitive Radio Networks: A Survey," in *IEEE Communications Surveys and Tutorials*, volume. 017, no. 03, p. p. 01342 - 01363, third-quarter 2015.

[4] Ankit S. Rawat, Priyanka Anand, Chen and Varshney, " Countering byzantine attacks in cognitive radio networks, " *IEEE International Conference on Acoustics, Speech & Signal Processing*, Dallas, 2010, p.p. 03098 - 03101.

[5] L. Zhang, et al., "Defending Against Byzantine Attack in Cooperative Spectrum Sensing: Defense Reference and Performance Analysis," in *IEEE Access*, vol. 04, p.p. 04011 - 04024, 2016.

[6] Fulai Liu, et al., "Dynamic attack probability based Spectrum Sensing against Byzantine attack in Cognitive Radio," *2nd IEEE International Conference on Computer & Communications*, Chengdu, 2016, p. p. 01494 - 01498.

[7] Jun Wu et al., "Generalized Byzantine Attack and Defense in Cooperative Spectrum Sensing for Cognitive Radio Networks," in *IEEE Access*, vol. 06, p. p. 53272-53286, 2018.

[8] B. Kasiri, J.Cai and Alfa, "Secure cooperative multi-channel spectrum sensing in cognitive radio networks, " *MILCOM Military Communications Conference*, Baltimore, 2011, p. p. 0272 - 0276

[9] F. Tao, et al., "Security Random Network Coding Model against Byzantine Attack Based on CBC," *4th International Conference on Intelligent Computation Technology & Automation*, Shenzhen, 2011, p.p. 01178 - 01181

[10] S. Mapunya and Velempini, "The Design of Byzantine Attack Mitigation Scheme in Cognitive Radio Ad-hoc Networks," *International Conference on Intelligent & Innovative Computing Applications*, Plain, 2018, p. p. 01 - 04.