2022 International Seminar on Computer Science and Engineering Technology (SCSET) | 978-1-6654-7876-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/SCSET55041.2022.00063

Trust Management Architecture for a Space Dispersed Computing Network

Houtian Wang*,
Qian Xuesen Space Technology Laboratory
Beijing, China
wanghoutian@qxslab.cn

Qing Yang
Shenzhen University
Shenzhen, China
yang.qing@szu.edu.cn

Qingxia Chen
Qian Xuesen Space Technology Laboratory
Beijing, China
chenqingxia@qxslab.cn

***Abstract*—To facilitate the next generation of scientific missions, spacecraft must possess advanced processing capabilities to support computation-intensive and time-sensitive tasks. The core capability of the next generation space information network is computing. However, network security can become compromised in the pursuit of network exposure and openness. Trustability is an important goal of future space information networks. In this paper, trust management architecture for a space dispersed computing network (SDCN) is proposed for the first time. The layered structure is designed from the perspectives of trusted authentication, network transmission, distributed consensus, collaborative computing, and verification audit. The principle-based aspects of each layer are then discussed to improve the security and trustability of the SDCN. In addition, the effective combination of trusted architecture with typical space task scenarios is also discussed.**

Keywords-Trust management, dispersed computing, space information network

I. INTRODUCTION

At present, the computing requirements of various applications in a space information network usually rely on ground stations, resulting in respond delays to users' application requirements. For example, the transmission of a remote sensing image to a ground station for processing can only begin when the remote sensing satellite and ground station are visible. This method greatly reduces the real-time performance of image detection and target recognition. For this reason, on-obit data generation and processing has been investigated by both industry and academia [1]. Moreover, with the rise of giant low-orbit satellite constellations such as Starlink, satellite miniaturization and mass manufacturing have become a trend. Maintaining momentum toward the creation of large constellations of nanosatellites requires a reimagining of space systems as distributed, edge-sensing and edge-computing systems. For typical tasks such as remote sensing, image processing and analysis, multiple satellites are clustered to form a collaborative computing network to process these tasks, so as to improve the response speed. This is a very important indicator for modern military missions. The use of Orbital Edge Computing (OEC) architecture to address the limitations of bend-pipe architecture has been proposed [2]. OEC supports edge computing at each camera-equipped nanosatellite, thus allowing sensed data to be processed locally when downlinking is not possible. Meanwhile, dispersed computing emerges as a promising solution by leveraging the geographically dispersed computational resources in the terrestrial network [3]. With the increasing frequency of space computing-intensive tasks, dispersed computing will be a development trend in the field of space information networks in the future.

However, due to the exposure characteristics of nodes and links in space information networks, they face multiple security threats [4]. Through the detection, monitoring, and analysis of the signal, prior information support is extracted and a variety of attack means such as data theft, deception attack, malicious program attack, replay attack, and data tampering are adopted to realize information theft, destruction, invasion, and control of satellite systems. Space information networks must cope with hostile environments, node subversion, stringent performance constraints, high tempo operations that lead to rapid changes in network topology, and service requirements. As a result, the methodology of building a trusted environment to facilitate dispersed computing in space information network is of critical importance.

Trust is a complex concept, it has different definitions in different fields such as psychology, sociology, economics, communication, and networking science [5]. In the field of communication and networking science, Eschenaur et al. defined trust as a set of relations among entities that participate in a protocol. These relations are based on the evidence generated by the previous interactions of entities within a protocol. In general, if the interactions have been faithful to the protocol, then trust will accumulate between these entities [6]. Meanwhile, the definition of "trust management" is given in [7]. It is clarified that "trust management is to maintain the trust relationship between network entities, so that the network is in a trusted state."

The above definitions of "trust" and "trust management" mainly focus on the network entity itself. With the gradual integration of network and computing, computing will become an important network resource. In this paper, we put forth novel decentralized trust

978-1-6654-7876-2/22/$31.00 ©2022 IEEE
DOI 10.1109/SCSET55041.2022.00063

management architecture for a space dispersed computing network (SDCN) based on the above definition. We address the challenge of trust management in space dispersed computing from three aspects including networking, storage, and computing. The main contribution of this paper is summarized as follows:

- In this paper, the description of the SDCN is given and trust management architecture for the SDCN is proposed for the first time, with the purpose of improving the security and trustworthiness of the network.
- The core components of each layer in the architecture are given and the schemes or research suggestions of each layer are described.
- Using the typical scenarios of intelligence recognition and the processing of remote sensing images as well as spectrum cooperation autonomy as examples, this paper discusses how to effectively combine the trust management framework with the aforementioned scenarios to build a trusted environment.

The rest of this paper is structured as follows. We give the description of the SDCN in section II, and then, the trust management in the SDCN from the perspective of networking, storage, and computing is discussed in section III. The trust management architecture for the SDCN is mentioned in section IV. Taking typical scenarios as examples, section V discusses how to effectively combine the trust management framework, and the concluding remarks are drawn in section VI.

II. THE DESCRIPTION OF THE SDCN

Fig. 1 illustrates a description of the SDCN. The SDCN is composed of two segments: the space segment and the ground segment. The space segment is mainly composed of satellites. The ground segment has various types of network elements located on land, such as the remote command center, local command center, ground vehicles, and mobile terminals. Meanwhile, computing is regarded as the resource parallel to the network connection. In the SDCN, both the satellite and the local command center have computing capability. They are called network computing nodes (NCN) in this paper. For example, when a remote sensing satellite finds an unknown target, it will search for the adjacent available NCNs, and the NCNs will form a temporary computing network to identify the target, and transmit the recognition results to the remote command center or the local command center, thus forming a fast response chain from sensing to computing and computing to decision-making. This is the main feature of the SDCN.

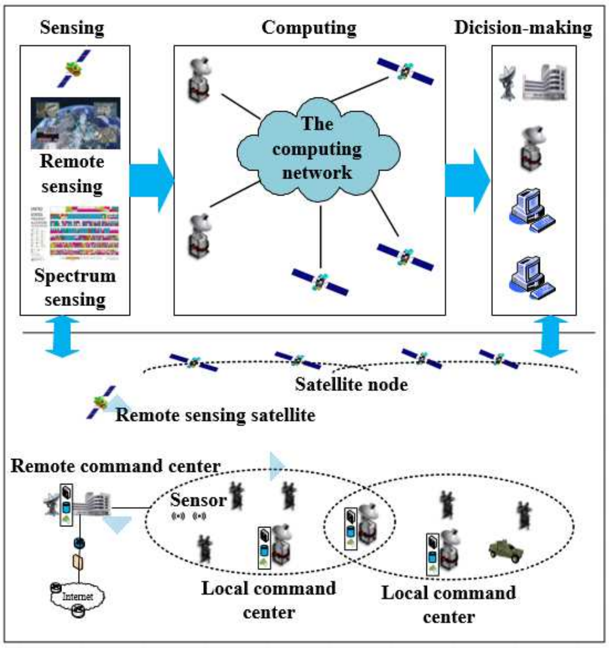

Figure 1. The description of the SDCN.

III. DISCUSSION OF TRUST MANAGEMENT IN THE SDCN

The SDCN keeps data close to users by outsourcing the data to the edge of the network. The integration of networking, storage and computing is an important feature of the SDCN. In this section, we will discuss the trust management problem from the perspective of the aforementioned three aspects.

A. Networking

Due to the dynamically changing nature of network topology, there is no predetermined trust relationship between nodes in the SDCN. Therefore, it is difficult to ensure that all relevant entities are trusted. Before accessing network services, each network node should be authenticated to ensure its authenticity and credibility. Without a sufficient security guarantee, it is easy for external attackers to attack the resources of services and infrastructure and evade punishment [8]. Moreover, the computing resources and nodes in the SDCN are decentralized, it is necessary to study the identity authentication mechanisms that are combined with the new characteristics of the SDCN.

Although identity authentication can eliminate the disguised or forged computing nodes and devices in the application, it is still difficult to ensure that all joining entities are completely trusted. If a node is disturbed or tampered with, it will become illegal, and the network will suffer a series of security threats. For example, the illegal node may block the normal use or management of communication facilities by causing excessive resource consumption. This can generate a significant amount of traffic and ultimately disrupt the functionality of the entire network [5]. Therefore, the effective evaluation of the trustworthiness of nodes in the SDCN is an important problem.

B. Storage

For the SDCN, the user will lose the final ownership of the generated data when the data is maintained on the dispersed computing nodes. As a result, the privacy of the user data is at risk [8]. For example, the dispersed computing nodes are located at the edge of the network near the data source. Compared with the cloud computing data center in the core network, the dispersed computing nodes can collect more valuable and sensitive information of users, including location information. Dispersed computing nodes may become a platform for malicious nodes to collect user privacy data. Also, an attacker can modify user data on the computing node to destroy evidence. Because the computing node has the right to process or modify the data, it is difficult to determine whether the computing node can be trusted with the data. For the SDCN, performing trusted computing while simultaneously ensuring the privacy of input data is a very important aspect of trust management.

C. Computing

In the SDCN, the computing nodes have certain computing abilities and can process and analyse data. The network has distributed computing resources to undertake cloud computing offloading. Specifically, the end device submits the task to the local edge computing node to obtain calculation results with low latency. However, due to the security threat inherent in the SDCN, the computing node can be attacked. Not only can the processing data on the computing node be exposed to the attacker, but also the computing results on the edge computing node can be controlled. Errors generated by one computing node will spread to other nodes and lead to incorrect final results. Therefore, all intermediate and final results should be verified to ensure the accuracy of the results and to track the misbehaving computing nodes that output the wrong results.

IV. A TRUST MANAGEMENT ARCHITECTURE FOR THE SDCN

Fig. 2 shows the trust management architecture for the SDCN. As the core element of building a trusted environment, the primary goals of the trust management architecture in this paper are as follows:

➢ Trusted authentication: the identity and legitimacy of the NCN can be verified without a centralized authority.

➢ Distributed consensus: combined with the characteristics of the SDCN, the consistency of the network state can be realized in the presence of Byzantine nodes.

➢ Privacy computing: space dispersed computing can be implemented under the condition of protecting the privacy of input data.

➢ Trusted computing: a guarantee that the computing process on the NCN is unmodified and the result is correct.

As shown in Fig. 2, the trust management architecture consists of five layers from bottom to top: the trusted authentication layer, the network infrastructure layer, the distributed consensus layer, the collaborative computing layer, and the verification audit layer.

A. Trusted authentication layer

The goal of the trusted authentication layer is to solve the problem of relying on a trusted third party in the centralized network structure, and to realize the consistency and temper proofing of authentication data in the distributed environment. Its core is designed to solve two problems, the node's identity authentication in the process of network access and the node's legitimacy verification in the process of communication. In the trusted network architecture proposed in this paper, the two problems were considered under a unified framework.

The trusted authentication layer consists of two core modules: the node identity authentication module (NIAM) and the node reputation value management module (NRVMM). When a new node joins the collaborative computing network, the NIAM is used to confirm the legitimacy of the node's identity. The NRVMM is mainly used to manage the legitimacy of nodes in the process of communication, and to isolate illegal nodes in time. There is a list to record the trust value of each node in the collaborative computing network. The reputation values in the list are maintained by the NCNs in the collaborative computing network with the help of the distributed consensus mechanism.

Because the satellite is in constant motion, the computing nodes in the collaborative computing network will change with time. When a new NCN joins the network, the verification of the legitimacy of the NCN's identity consists of confirming whether it belongs to the same alliance with the NCNs in the collaborative computing network. Suppose that the NCNs in the alliance have a shared key k and it cannot be read and rewritten externally. At the same time, the NCN with the highest trust value will be elected as the primary validator in the collaborative computing network. We use p to represent the new node and q to represent the primary validator. The primary validator q sends a random number r to p, and q encrypts r with the shared key k. If p's identity is legal, p will encrypt r with the same key k and will send the encrypted result to q. In this way, the encrypted results of the two nodes are consistent. If the identity of p is illegal, p cannot complete the encryption operation and the encryption results of the two nodes are inconsistent. Meanwhile, the illegal nodes detection mechanism will run all the time, marking the trust value of neighbour nodes in real-time, and it will write the results into the list in the NRVMM. Each node in the network will refer to the trust value to determine whether to continue to cooperate with neighbour node.

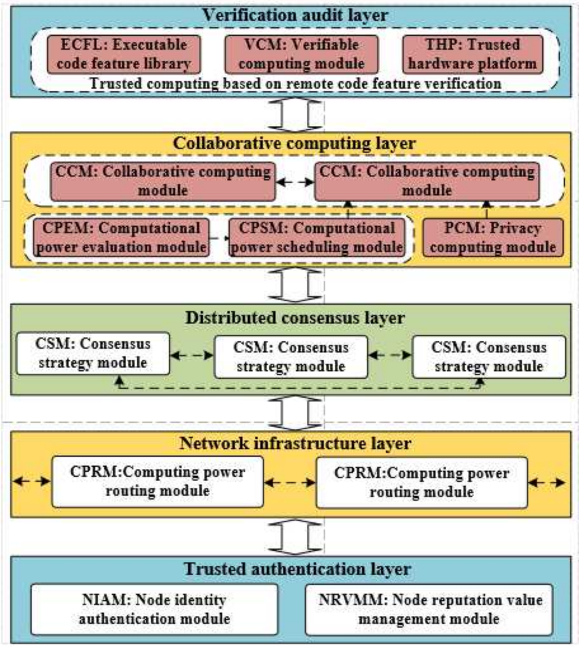
Verification audit layer
ECFL: Executable code feature library
VCM: Verifiable computing module
THP: Trusted hardware platform
Trusted computing based on remote code feature verification
Collaborative computing layer
CCM: Collaborative computing module
CCM: Collaborative computing module
CPEM: Computational power evaluation module
CPSM: Computational power scheduling module
PCM: Privacy computing module
Distributed consensus layer
CSM: Consensus strategy module
CSM: Consensus strategy module
CSM: Consensus strategy module
Network infrastructure layer
CPRM:Computing power routing module
CPRM:Computing power routing module
Trusted authentication layer
NIAM: Node identity authentication module
NRVMM: Node reputation value management module

Figure 2. The space trusted network architecture based on dispersed computing.

B. Network infrastructure layer

The underlying data communication network is a key infrastructure of the SDCN. However, we confront several challenges that are unique to the SDCN. First, the data link is unreliable and transient because it depends on the distance and antenna angle of the satellites. Second, satellites move quickly in their unique orbits, therefore the topology of the SDCN changes dramatically in real-time. Third, the computing capacity, storage capacity, and the ongoing task execution of the nodes should be considered in the process of information transmission.

As is shown in Fig. 2, the network infrastructure layer contains the CPRM, which can perceive parameters such as the network status, computing capacity, and storage capacity. The computing delay, network transmission delay, and trust value are regarded as the service performance indicators of routing. To provide network infrastructure support for the SDCN, the module can route and address information according to the above indicators.

C. Distributed consensus layer

The distributed consensus layer is responsible for synchronizing the state of all the NCNs. In the space information network, the NCN may be down due to hardware failure, or it can be hacked to behave maliciously (Byzantine fault). To achieve consensus under these scenarios, distributed consensus algorithms are needed for the SDCN. The proof of Work (PoW) algorithm is the most popular consensus algorithm used by popular blockchain projects such as Bitcoin and Ethereum. However, running the PoW algorithm consumes a large amount of computational resources such as CPU cycles and Input-Output (IO) resources. Unfortunately, the computational resources of satellites are limited in the SDCN, and thus PoW algorithm is infeasible in our case. Unlike the PoW algorithm, the practical Byzantine Fault Tolerance (PBFT) algorithm is a communication-based consensus algorithm that uses less computing power [9]. In the PBFT consensus algorithm, a group of nodes are chosen as the validators to participate in the consensus process; other nodes update their states by sending transactions to the validators. Among the validators, one is selected as the primary validator who coordinates the consensus process. Although the original PBFT algorithm consumes little computing power, it requires large amounts of message communication to reach consensus. The communication complexity of the original PBFT algorithm is $O(n^2)$, where n is the total number of validators. The original PBFT algorithm occupies a lot of bandwidth resources and is not practical in our scenario.

HotStuff, which is a leader-based Byzantine fault-tolerant replication protocol and an improved version of PBFT, is proposed in [9]. Fig. 3 shows the implementation of HotStuff. As shown in Fig. 3, the validators send endorsement messages with their own signatures to the primary validator. When the primary validator collects more than 2/3 of the endorsement messages, it aggregates the signatures into a single signature. Then, the primary validator broadcasts the message to other validators. This method can reduce the communication complexity to O(n) without degrading its security. Moreover, other efficient consensus algorithms have been proposed in recent years, which provide a reference for the design of consensus strategy in the SDCN [10], [11]. We emphasize that the design of the strategy in the SDCN should minimize the consumption of computing power and bandwidth.

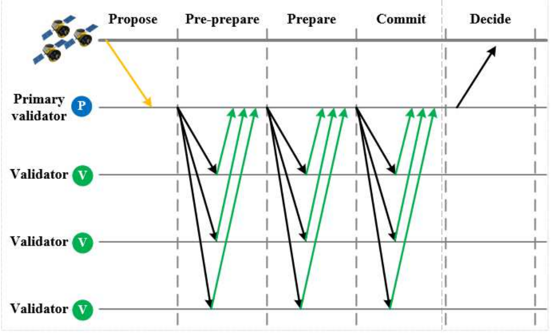

Figure 3. The detailed implementation of the improved PBFT consensus algorithm.

D. Collaborative computing layer

In the collaborative computing layer, the function of the CPEM is to evaluate the available computing resources of each computing unit in combination with the hardware types (GPU, FPGA, etc.), so as to provide an alternative set for the CPSM. Considering networking, storage, computing, and other indicators, the CPSM is responsible for making the best scheduling strategy for a given task. Considering the computing capacity and network state, the CCM is used to interconnect multiple NCNs for collaborative computing. Additionally, the CCM is the core module of the collaborative computing layer.

Specially, the function of the PCM is to obtain the correct calculation results under the condition of protecting the privacy of input data. There exist several typical privacy computing methods. The idea of fully homomorphic encryption [12] is to calculate the ciphertext data to get a ciphertext result and to decrypt the ciphertext result to get the plaintext calculation result. But the limitations are as follows:

- For the task execution algorithm, the complexity of realizing full homomorphic encryption is high;
- High computing resource consumption;
- Secure computing from the perspective of encryption and decryption

Under the premise of a lack of trusted third party, the correct result can be obtained by jointly calculating a function without disclosing the input value, which is the main function of secure multiparty computation (MPC) [13]. For the MPC, one can enforce honest results given that 2/3 of the players are honest. This is a very active field of research. But the limitations are as follows:

- The honesty of participants cannot be guaranteed;
- There are many task scenarios in the SDCN. Different computing nodes can calculate different subtasks or jointly compute a task. Deeply matching secure multiparty computing with the SDCN requires further research.

In general, although the current privacy computing methods still have technical bottlenecks, the methods mentioned above can provide solutions for the subsequent implementation of space privacy computing.

E. Verification audit layer

For the SDCN, due to the threat of Byzantine errors, it is impossible to ensure that all nodes involved in the calculation use the correct algorithm to complete the calculation task, resulting in final calculation results that cannot ensure correctness. The role of the verification audit layer is to verify the correctness of the calculation results. In this paper, a method to verify the calculation results is proposed. We assume that the task code to perform the calculation is pre-set. The implementation framework of the verification audit layer is shown in Fig. 4.

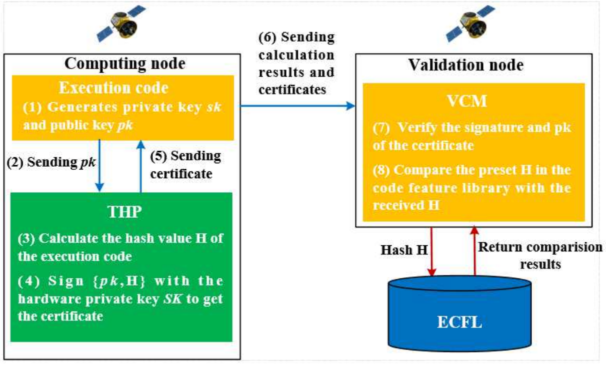
Computing node
Execution code
(1) Generates private key *sk* and public key *pk*
(2) Sending *pk*
(5) Sending certificate
THP
(3) Calculate the hash value H of the execution code
(4) Sign {*pk*,H} with the hardware private key *SK* to get the certificate
(6) Sending calculation results and certificates
Validation node
VCM
(7) Verify the signature and pk of the certificate
(8) Compare the preset H in the code feature library with the received H
Hash H
Return comparision results
ECFL

Figure 4. The detailed implementation of the improved PBFT consensus algorithm.

The verifiable computing model consists of computing nodes and verification nodes. Firstly, a cryptographic module that generates a random private key and the corresponding public key is embedded in the execution code, and the corresponding private key *sk* and public key *pk* are generated for the node before the computing task starts. Then, the public key *pk* is sent to THP of the computing node. The THP is an independent security module in the CPU chip. It has a private key *SK* that is written into the CPU when it is delivered. The private key is only used for the digital signature and cannot be read and rewritten externally. The THP is responsible for analyzing the task code, obtaining the signature code, signing the analysis result with THP's private key, and returning the signed certificate to the task code. Then, the computing node starts to execute the computing task. When the computing task is completed, the computing node sends the calculation result and certificate to the verification node. The verification node first verifies the signature of the certificate and the public key of the task. The purpose of this step is to ensure that the execution code is indeed audited by the THP. Thereafter, the hash value H of the execution code is extracted. Assuming that the execution code is known in advance, the characteristic value of the execution code is recorded in the library. The verification node compares this value with H. If the two values are equal, the execution code has not been tampered with and the calculation result is correct.

V. APPLICATION SCENARIO ANALYSIS

Scenario A: On-orbit intelligent recognition and processing of remote sensing images. In this task scenario, multiple satellites are required to achieve adaptive coordination and task allocation without completely relying on ground stations. The intelligent recognition of remote sensing images includes tracking, confirming, and observing the observed targets. The multidimensional observation data of the targets are then fused. Finally, the concise and effective information is transformed to improve the situation analysis ability and support the rapid response. As is shown in Fig. 5, this scenario will go through the process of "complex task decomposition-collaborative computing-result aggregation". This scenario uses a lightweight container computing carrier. The resources on the satellite are relatively limited, and the container deployment needs less resources. However, these resources can flexibly update, schedule, and migrate quickly, and support the rapid verification, deployment and updating of new algorithms.

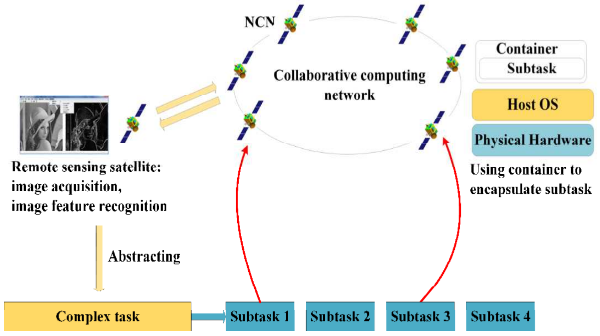
NCN
Collaborative computing network
Container
Subtask
Host OS
Physical Hardware
Using container to encapsulate subtask
Remote sensing satellite: image acquisition, image feature recognition
Abstracting
Complex task
Subtask 1
Subtask 2
Subtask 3
Subtask 4

Figure 5. On-orbit intelligent recognition and processing of remote sensing images.

Due to the dynamic changes of the network topology, there are two key issues surrounding the theme of trust management in this scenario: (1) the trusted authentication of nodes under the dynamic changes of network topology; (2) ensuring the trust of the calculation results when the nodes are disturbed. In the trust management architecture proposed in Section IV, the access control layer and the verification audit layer can provide solutions to the aforementioned two problems.

Scenario B: spectrum cooperative autonomy. Spectrum cooperative autonomy is mainly to make the multiple heterogeneous wireless networks distributed in the complex electromagnetic environment coordinate their waveforms based on network interaction and independent decision-making. Meanwhile, the order of electromagnetic space use is established independently, so as to get rid of the dependence on the central control node, and realize low-interference elastic coexistence in the physical space with limited resources. Fig. 6 shows the spectrum collaboration application scenario for the space information network.

As is shown in Fig. 6, each network is equipped with a spectrum decision node, which is acted on by the satellite. When the networks overlap each other geographically and use the same frequency band, the spectrum decision node performs spectrum decision by training the machine learning algorithm according to the local electromagnetic information, so as to minimize the spectrum conflict when the nodes access the network. The spectrum cooperative computing network is composed of spectrum decision nodes. There are three key issues surrounding the theme of trust management in this scenario: (1) Spectrum decision nodes rationale for performing security calculation under the condition of protecting input data; (2) In the process of spectrum collaboration, reaching consensus in a distributed mode; (3) Ensuring the trust of the calculation results when the nodes are disturbed. In the space trusted network architecture based on dispersed computing, the distributed consensus layer and the verification audit layer can provide solutions to problem (2) and (3). The privacy computing module in the collaborative computing layer can provide a reference for technical direction for problem (1).

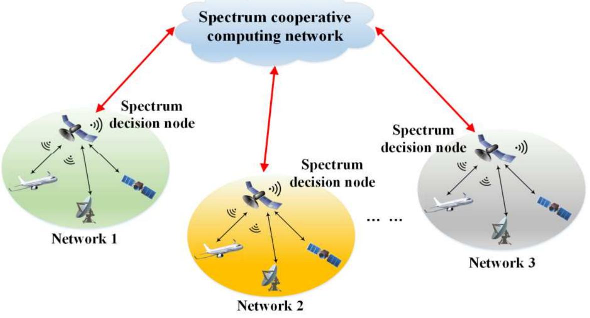

Figure 6. The spectrum collaboration application scenario for the space information network.

VI. CONCLUSIONS AND FUTURE WORK

This work presents a novel decentralized trust management architecture for the SDCN. Unlike conventional trust management that relies on a centralized authority, the proposed architecture is decentralized and suits the environment of the space information network. We discuss the problem of trust management for the SDCN from three aspects: networking, storage, and computing. Furthermore, we design a complete trust management architecture that consists of five layers: the trusted authentication layer, the network infrastructure layer, the distributed consensus layer, the collaborative computing layer, and the verification audit layer. The role of each layer in the hierarchical architecture is described, and the technology selection is discussed. To show the potential application of the trust management architecture, we discuss two typical space missions involving remote sensing image processing and spectrum collaborative autonomy. Both use cases show that the proposed trust management architecture is necessary in the future SDCN.

However, challenges still remain to fully implement the trust management architecture in a practical SDCN. When dose the SDCN perceive the distribution of the underlying resources, how to guide different business requirements to the corresponding computing service nodes, and how to quickly match the task data with the computing services to form an adaptive resource routing strategy, are some issues that need to be resolved, and a set of space dispersed computing protocol needs to be developed to address these problems. Second, blockchain does not rely on a trusted authority to store and update a ledger. The structure and characteristics of the blockchain enable it to be tamper-proof, traceable, highly trustable and highly available. Making full use of the core idea of blockchain for the SDCN is a new research topic for our future study. Finally, with more and more organizations planning to build their own space information networks (e.g., the Starlink project by SpaceX), cross-organization cooperative trust management will be an interesting research area that will be worth further investigation.

ACKNOWLEDGMENTS

This work was financially supported by the National Natural Science Foundation of China (No. 61801478).

REFERENCES

[1] Z. Zhang, W. Zhang, and F.-H. Tseng, Satellite mobile edge computing: Improving qos of high-speed satellite-terrestrial networks using edge computing techniques, IEEE network, 33 (2019) 1 70–76.

[2] B. Denby and B. Lucia, Orbital edge computing: Nanosatellite constellations as a new class of computer system, in Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, (2020) 939–954.

[3] M. R. Schurgot, M. Wang, A. E. Conway, L. G. Greenwald, and P. D. Lebling, A dispersed computing architecture for resource-centric computation and communication, IEEE Communications Magazine, 57 (2019) 7 13–19.

[4] B. Li, Z. Fei, C. Zhou, and Y. Zhang, Physical-layer security in space information networks: A survey, IEEE Internet of Things Journal, 7 (2019) 1 33–52.

[5] J.-H. Cho, A. Swami, and R. Chen, A survey on trust management for mobile ad hoc networks, IEEE Communications Surveys & Tutorials, 13 (2010) 4 562–583.

[6] L. Eschenauer, V. D. Gligor, and J. Baras, On trust establishment in mobile ad-hoc networks, in International workshop on security protocols.Springer, (2002) 47–66.

[7] M. Blaze, J. Feigenbaum, and J. Lacy, Decentralized trust management, in Proceedings 1996 IEEE Symposium on Security and Privacy, (1996) 164–173.

[8] J. Ni, K. Zhang, X. Lin, and X. S. Shen, Securing fog computing for internet of things applications: Challenges and solutions, IEEE Communications Surveys & Tutorials, 20 (2017) 1 601–628.

[9] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, Hotstuff: Bft consensus in the lens of blockchain, arXiv preprint arXiv:1803.05069, (2018).

[10] T. Wang, X. Bai, H. Wang, S. C. Liew, and S. Zhang, Game-theoretical analysis of mining strategy for bitcoin-ng blockchain protocol, arXiv preprint arXiv:1911.00900, (2019).

[11] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, Pobt: A lightweight consensus algorithm for scalable iot business blockchain, IEEE Internet of Things Journal, 7 (2019) 3 2343–2355.

[12] A. Alabdulatif, I. Khalil, and X. Yi, Towards secure big data analytic for cloud-enabled applications with fully homomorphic encryption, Journal of Parallel and Distributed Computing, 137 (2020) 192–204.

[13] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, and Y.-a. Tan, Secure multi-party computation: Theory, practice and applications, Information Sciences, 476 (2019) 357–372.