A blockchain based architecture for asset management in coalition operations

Dinesh Verma*[a], Nirmit Desai[a], Alun Preece[b], Ian Taylor[b]
[a]IBM TJ Watson Research Center, 1110 Kitchawan Road, Yorktown Heights, NY 10598, USA
[b]Cardiff University, Cardiff, Wales, CF10 3AT, UK

ABSTRACT

To support dynamic communities of interests in coalition operations, new architectures for efficient sharing of ISR assets are needed. The use of blockchain technology in wired business environments, such as digital currency systems, offers an interesting solution by creating a way to maintain a distributed shared ledger without requiring a single trusted authority. In this paper, we discuss how a blockchain-based system can be modified to provide a solution for dynamic asset sharing amongst coalition members, enabling the creation of a logically centralized asset management system by a seamless policy-compliant federation of different coalition systems. We discuss the use of blockchain for three different types of assets in a coalition context, showing how blockchain can offer a suitable solution for sharing assets in those environments. We also discuss the limitations in the current implementations of blockchain which need to be overcome for the technology to become more effective in a decentralized tactical edge environment.

Keywords: coalition systems, blockchain, asset sharing systems, asset management, software defined coalitions, SDC, virtual federation, tear sheets.

1. INTRODUCTION

During coalition operations, where dynamic communities of interests (CoIs) [1] need to be formed rapidly to deal with the events unfolding on the ground, efficient sharing of assets among coalition partners can make a significant impact on the effectiveness of coalition operations. The increasing number and diversity of asset types - including physical assets such as sensing systems and virtual assets such as databases – exacerbates the problem of dynamically assigning assets to coalition missions. Moreover, national policies and security considerations may hinder complete linkage among various coalition members. In order to enable rapid dynamic sharing of assets, while maintaining compliance with national policies, new architectures that enable asset information to be shared between participating coalition members are needed.

In wired environments, various systems built on blockchain are emerging (e.g. [2]) and this technology has many attractive features for coalition operations. It provided a mechanism for parties that do not trust each other to validate transactions without requiring a central trusted authority. This approach can offer a number of benefits for coalition operations. In this paper, we explore some of the use-cases and situations in coalition operations where blockchain technology can be used to advantage. At the same time, since blockchain arose in wired environments, there are issues related to performance and efficiency in tactical edge environments, and we need to rethink blockchain implementations in ways that may be more appropriate for bandwidth constrained environments.

We begin this paper with a brief overview of blockchain technology, followed by a discussion of three different use-cases in coalition operations where blockchain technology can be used to advantage. We then discuss blockchain based solution for each of those use cases. Finally, we discuss some of the challenges that may need to be overcome for coalition operations for blockchain based systems to become more suitable for those environments.

2. BLOCKCHAIN OVERVIEW

A blockchain is a distributed database that maintains a set or information bundles called blocks. The blocks are linked together in an ordered list, with each block containing a pointer to the previous block in the list. A time-stamp maintained in each block would typically be used as the mechanism for ordering the blocks. All but one of the blocks maintained in the distributed database are immutable, i.e. they cannot be modified.

A blockchain can be used to implement a distributed ledger to maintain a record of different transactions happening in a distributed system. The previously signed blocks in the ledger contain an immutable record of various transactions that have been authenticated by the ledger. The current set of transactions make up a new block, and within a finite timeframe, would be collectively signed into the new block that would then become immutable.

In order to provide a secure but efficient recording, the Merkle tree data structure [3] is used to store the contents of a block. In a Merkle tree, each node (other than the leaf nodes) has a label which is a cryptographic hash of the labels of its children nodes, and the label of the lead node is the contents of that node. The Merkle tree has the advantage that the contents of a large set of transactions can be validated very quickly. Typically, each block in the chain would contain a Merkle tree of transactions and a hash of the contents of the previous block. This allows the blockchain data structure to rapidly validate the existence of a previous transaction in any of the blocks and a client can verify that a transaction was included by obtaining the Merkle root from a block header and a list of the intermediate hashes from a full peer.

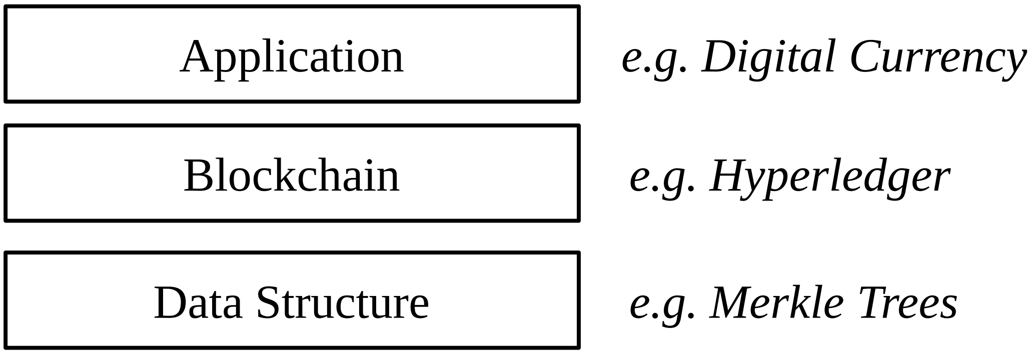

Figure 1. Different layers in a blockchain solution

Conceptually, the blockchain based system can be modeled as a structure consisting of three layers shown in Figure 1, a data structure layer defining how blocks would be structured (based usually on Merkle trees), a distributed ledger structure defining how the chain of blocks in maintained and an application layer. The application would determine when and how to create a new block, the definition of a transaction, and which transactions ought to be included in the new block to become an immutable previous block.

The most common application of blockchain is to enable digital currencies such as bitcoin[4], in which transactions are a record of entities that represent the spending of digital coins with a sender, receiver and account. An initial block is signed with a key, and signing of the new block requires inverting a cryptographic function for the key of the latest immutable block. When an entity discovers such a key (the process of mining in bitcoin), it gets the right to create the latest block in the chain, and it will include the set of transactions that the majority of people are confirming as being valid into the record of the signed block.

The major advantage of the blockchain approach is that different entities can maintain the distributed ledger without requiring any single trusted central location. The major disadvantage of the blockchain approach is its relatively inefficiency (A hash of transactions need to be sent to every entity) and the latency in recording (when a new block is marked immutable), and information leakage (the hash and the volume of transactions is visible to all members in the network).

The ability to maintain a distributed ledger without any central trust authority is a special feature of the blockchain distributed ledger. That feature can be very useful in a coalition context. The distributed ledger would have many participating entities, each belonging to one of the coalition members. Each entity can create an identity for itself and get it signed using a public key issued by the coalition partner that own the entity, i.e. a U.S. entity can sign its identity using a U.S. issued public key, while a UK entity can sign its identity using a UK issued public key. Assuming that the coalition members are sharing their public keys with each other, any entity belonging to the coalition can join the distributed ledger system.

A transaction is information exchanged among different entities belonging to the coalition, and can involve sharing of physical assets or information assets. A cryptographic hash of the transactions is shared with all entities participating in the distributed ledger. Each block in the blockchain consists of a group of transactions. At selected instances of time, when all entities in the network have reached a consensus that a new set of transactions should be signed and marked immutable, a new block is created and marked as such by one of the entities. The set of transactions which the majority of participants agree are valid based on the hashes they have seen previously are the ones that are included in the new block. The newly created block is distributed to all of the entities, and entities move on to the task of collecting transactions for a new block.

3. THREE COALITION SCENARIOS

In a coalition setting, there are many types of assets that may need to be shared among different partners. These assets that may need to be shared include physical assets as well as information assets. In this section, we look at three possible coalition scenarios and contexts where the use of a technology like blockchain can be beneficial. Those three scenarios are (i) software defined coalitions (ii) distributed asset assignment and (iii) information sharing on multiple disparate networks using tear sheets.

3.1 Software Defined Coalitions

In coalition operations, mission needs often require the formation of dynamic communities of interests (CoI) which are short-lived teams formed involving one or more coalition members. In order to support the dynamic CoIs, an IT infrastructure which can support these groups is needed. Since IT assets from different coalitions may not always be compatible with each other, an architecture for interoperability that borrows from the concepts of software defined networking [5] has been proposed for enabling dynamic CoI and referred to as a software defined coalition [6].

The structure of a Software Defined Coalition as articulated in [6] is shown in Figure 2, where the system assumes that the assets of CoI can come from either U.S. or UK. U.S. assets talk to a U.S. controller and UK assets talk to a UK controller.

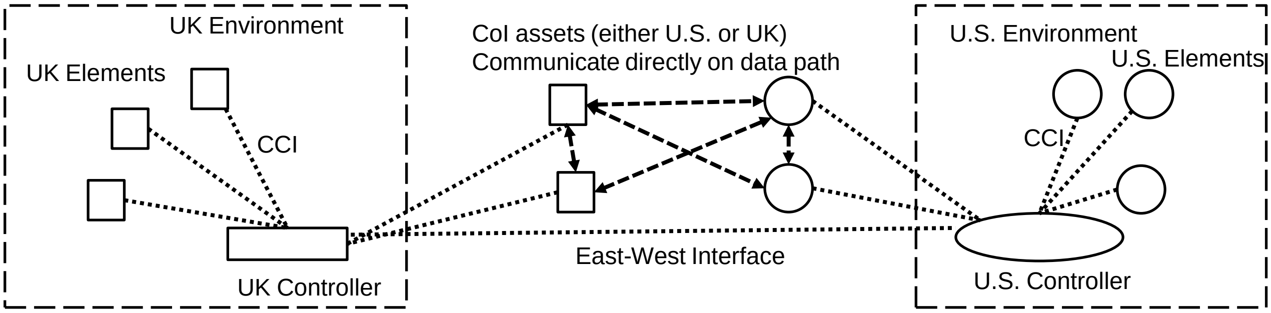

Figure 2. Software Defined Coalition Scenario

The controller provides control information, e.g. routing data, security information, policy information etc. to their corresponding country elements, thereby allowing them to communicate with each other with the right security policies. The communications along the data path happening between the CoI assets are shown using the thick dashed lines in the system. However, each asset only obtains control information from its own controller. This paradigm, which is based on the same principles as that of SDN, allows coalition assets to cooperate within the guidelines and control mechanisms set forth by their organizations. Depending on the level of trust between the two partners, the data communication may involve forwarding packets, sharing storage, allowing offloading of computing capabilities, or sharing of sensing capabilities between devices.

Since there are multiple controllers involved in a dynamic CoI, the different control operations happening among the controllers need to be coordinated. Specifically, if a control configuration for a CoI asset belonging to the U.S. comes from a UK machine, because the specific mission is being operated by the UK commander, the transactions requesting the change in command and configuration need to be recorded, so that they can be used for any post-mission analysis. Obtaining such distributed recording would be an appropriate case for using blockchain technologies since they obviate the need for another authority which all coalition partners would need to trust in order to record the transactions in this environment.

3.2 Distributed Asset Assignment

One of the key challenges in a coalition operation is to determine which assets are available in order to conduct a specific mission. In order to assign assets to the mission, an assignment approach needs to be used. An example of a typical assignment system for the coalition operations is the Sensor Assignment to Missions (SAM) approach [7], which uses knowledge representation and reasoning technologies (ontologies and rules) to assign assets to mission tasks.

When a request for a mission is received, the asset management tool would look at its set of available assets to determine which assets can be best allocated to the mission. In order to perform its task, the asset management tool needs to have an inventory of available and allocated assets, the missions to which the assets are assigned, and the use of its local policies and semantic techniques to assign the assets to the missions.

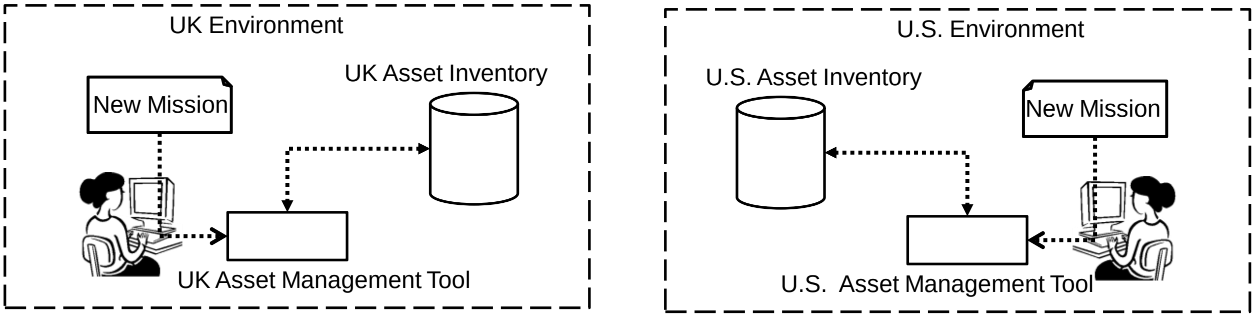

Figure 3. Distributed Asset Assignment Scenario

An asset assignment system like SAM would be used by each of the coalition partners to determine the manner in which to best assign their assets to the mission. However, when coalition operations need to be conducted, the prevailing approach will be for each coalition to consider assignments independently for their own needs. This leads to a challenge because the totality of the assets is not known to any individual partner, and cannot be used to share assets optimally across the missions of the entire coalition. That setup is shown in Figure 3.

It would be ideal if the information about all the assets available to each coalition partner could be maintained in a central inventory, which would allow the assets to be used in the best manner possible for the missions that are to be conducted. However, in a coalition context, where the trust between different partners is not absolute, it is not easy to get agreement on who should be owning and operating this global asset inventory. Furthermore, maintaining a central resource is challenging in such an environment because of the mobile and transient nature of the environment.

In this context, a blockchain based solution could be effective in that it could enable the creation of a virtual centralized repository, consisting of multiple decentralized cooperating peers, without requiring a physical central inventory.

3.3 Tear Sheets

In coalition environments, as well as in many other scenarios, multiple networks operate with different levels of security and each network operates independently with an air gap between them, i.e. no device can be concurrently connected to both of the networks at the same time. At the same time, information sometimes needs to be passed between the different secure networks. The concept of tear sheets is used to enable the sending of information between the networks of different security levels.

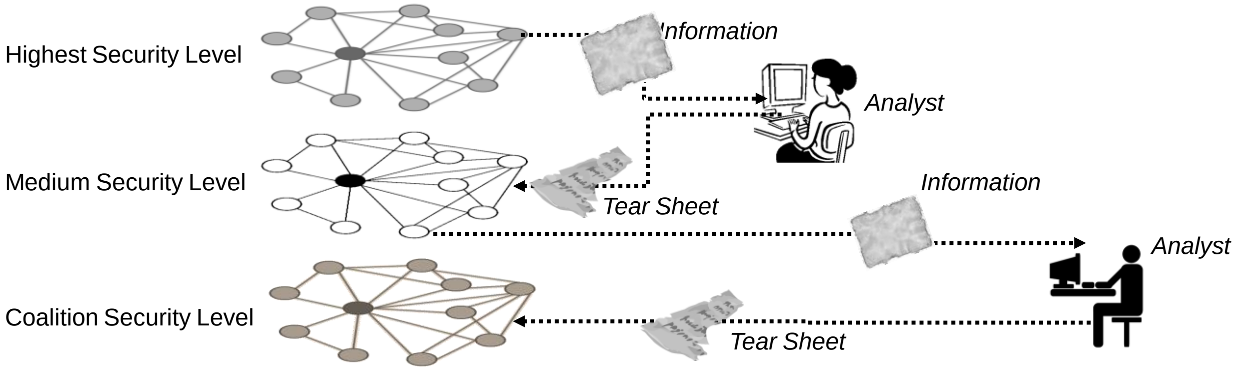

Figure 4. Tear Sheets among Networks with Multiple Level of Security

Using a tear sheet, a user with access to a network with a higher security level can extract a piece of information that needs to be provided to a network with a lower security level. The piece of information is extracted from the more secure network will remove any information that is considered having a high level of security, and usually any details about the origin or the validity of the information will be removed. The sanitized information is called a tear sheet from the previous practice of tearing out part of a page from a book, and handing it over. The tear sheet would then be sent to the less secure network.

The challenge in the less secure network is that there is no way to trace the origin of the tear sheet which has been completely anonymized, and even the information about the network that originated the tear sheet is lost. As a result, the validity or authenticity of the tear sheets in the network with lower security is frequently lost.

In an ideal environment, a central repository that indexes all tear sheets and associates them with the network they originated from, along with appropriate details that can be seen at an appropriate security level would help in tracing and tracking tear sheets across different network security levels. However, security considerations prevent the creation of such a central repository. However, the use of blockchain technology may enable the creation of such a virtual repository without compromising the constraints of maintaining air gaps between different networks.

In the next few sections, we will discuss how a blockchain based solution can help in addressing the issues encountered in each of the three contexts.

4. BLOCKCHAIN FOR SOFTWARE DEFINED COALITIONS

Blockchain technology can be used as a mechanism to record and track transactions happening between the different controllers participating in a software defined coalition. In order to perform post-mission analysis, a record of the different interactions happening between the different controllers will be useful. However, the storing of records in a central repository for all the operations happening in a coalition setting is not possible. Using a blockchain based approach, one can create a distributed system which can keep track of different transactions happening within the coalition.

One approach to create such a system would be to use one of the existing distributed ledger projects based on blockchain such as the Hyperledger Fabric [8]. The distributed ledger protocols allow for different participating elements (the SDN Controllers in this specific case) to register to the ledger, and record transactions. The transactions are considered final when a consensus is reached among participating members, with the mechanism for consensus being a pluggable module that can be modified. New blocks can be created on every transaction on which consensus is reached, or at some periodic time interval. Hyperledger Fabric also contains a full REST API, which makes this level of control more seamless for third party implementations.

Software defined coalitions are a mechanism to support dynamic community of interests, so a way to achieve consensus mechanism for a transaction would be to have a consensus management system which includes only the participating members of a CoI. While the same Hyperledger can be used to record all the transactions of all the concurrent activities, the transactions (e.g. control information that is exchanged among SDN controllers) can be assumed to reach consensus only when the participants in the CoI agree upon the transaction exchange. The consensus algorithm can be further simplified by only requiring that the controller belonging to the leader of the CoI, and the other partners involved in the control plane interactions, have reached an agreement on a transaction happening. Such consensus can be recorded by having both controllers sign a hash of the transaction and distributing it to all of the controllers. The coalition partners can then decide on taking all such transactions at periodic intervals and making that into a new immutable block.

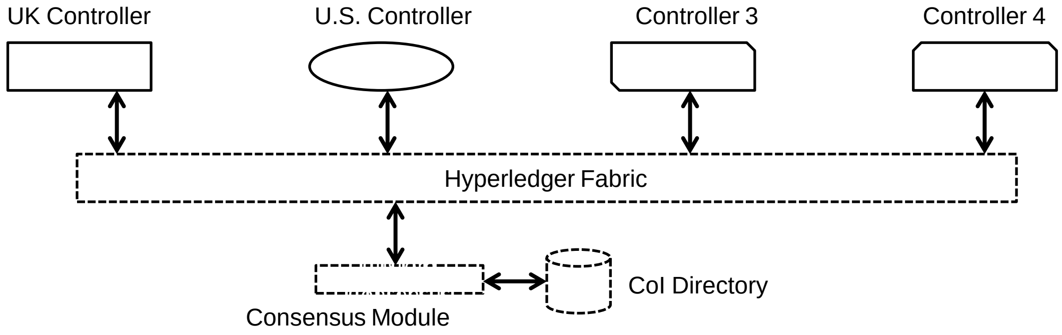

Figure 5. Blockchain based solution for Software Defined Coalition Scenario

The blockchain based solution as it looks for this scenario is described in Figure 5. Each of the controllers in the coalition talk to Hyperledger Fabric, which is shown in dashed line, representing the fact that it is not a physical entity but a collaborative software running at each of the physical controllers. The consensus module is customized for this scenario, with each consensus looking at the local CoI Directory. Each of the controllers can maintain a directory of the CoI their country is participating in, thereby getting consensus related messages only for transactions in the CoI that they participate in. A coalition partner who is not in a CoI would not need to get the messages for that CoI, whereby coalition CoIs may happen with different partners in parallel and maintain a level of information security among coalition members.

5. BLOCKCHAIN FOR DISTRIBUTED ASSET ASSIGNMENT

In the case of the distributed physical asset assignment, the goal of using blockchain is to create a virtual inventory which spans all of the assets available across the various coalition members. While each member of the coalition would maintain its own inventory of the different assets, the assets may need to be shared across partners for a given set of missions. The main task in the federation of the asset inventory is to manage the assignment of assets to missions, decide whether an asset ought to be reassigned from a mission, and to track the health of the asset over its life cycle.

The virtual directory model is shown in Figure 6. The inventory management systems talk to the hyperledge fabric.

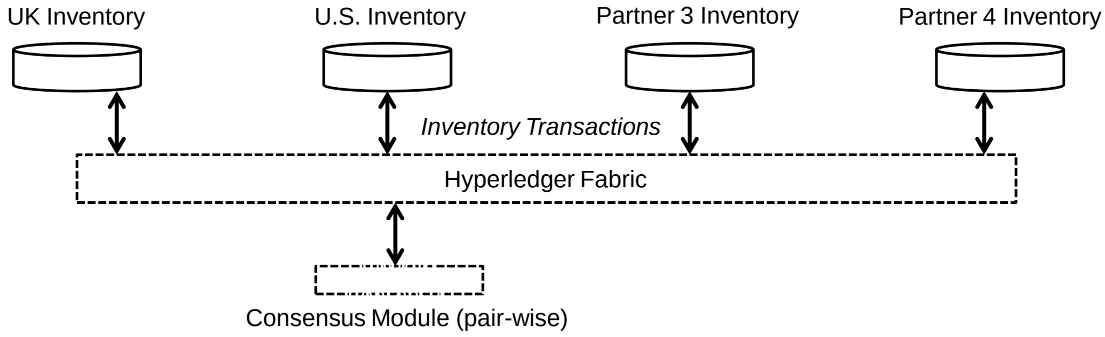

Figure 6. Blockchain based solution for Distributed Asset Assignment Scenario

As in the case of software defined coalitions, a consensus module that records transactions pair-wise is needed since inventory transactions would happen among two pairs of parties. A transaction is considered final if it is signed by both of the two parties involved in the transaction. The blocks can be represented using a Merkle tree, created for a passage of time and contain the hash of all transactions that happen within the given interval of time.

6. BLOCKCHAIN FOR TEAR SHEETS

Tear Sheets provide an interesting use-case of blockchain because it would need to operate in an environment which is not completely connected. In order to improve the operation and effectiveness of tear sheets, we envision augmenting the task of a human being by means of a robot which connects to only one network at a time. The robotic system assists a human being who prepares the tear sheets by taking them and transmitting them over the network with lower security classification. In the future, we can also envision the robot assisting the human further by validating the tear sheets for compliance with policies negotiated for coalition members [9], or using intelligent systems like IBM Watson [10] to ease the task or creating tear sheets. In this paper, we take the position that the robot has a far less significant role, that of ensuring that the tear sheets can be tracked and maintained without violating the constraints of the multi-level security.

In this model, the robot maintains a blockchain-based registry of tear sheets that are provided out of each level of the registry. The robot is the only entity accessing the blockchain, so it manages the blocks by means of creating a sequence of keys, computing the hash of each of those keys for a given large number, and then using the sequence in an inverse order for signing the blocks to make them immutable. For each level of the network, the robot maintains information in two repositories (e.g. two separate physical USB flash drives), one containing the control information for that level, and the other maintaining the data. The control repository contains the chain of keys to be used for signing the blocks, and any other security credentials and keys needed to access the data repository. The data repository contains the actual tear-sheets, which are indexed by means of a cryptographic hash computed over their contents.

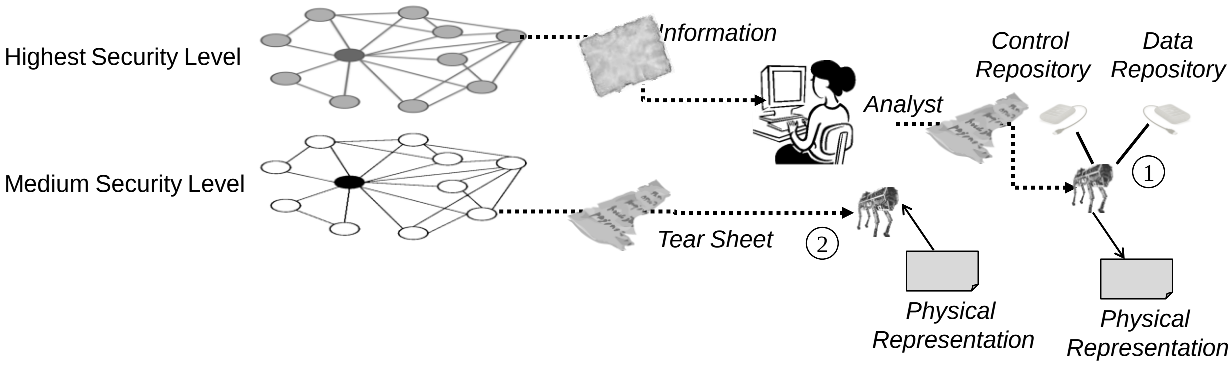

Figure 7. Tear Sheets Validation using a robot across multiple levels of networks

When a tear sheet is created from a network with a higher security level to a network with a lower security level, the robot initially attaches its computer to the two physical repositories at that level of security. It takes the tear sheets created by the human user, and computes a cryptographic content-based hash for each of the tear sheets, adding the contents of the tear sheet and the hash to the data repository for that level. The robot also creates a physical representation of the tear sheets, e.g. printing out an encoded message of the contents on a piece of paper that can be retrieved later. The robot then physically disconnects from the network at the higher security level, and connects to the network at the lower security level. It can then read the physical representation of the tear sheets, and transmit them to the lower security level network, making the repetitive tasks of transmitting the tear sheets manually much easier. Figure 7 illustrates this process.

On the reverse side, when someone needs to find out the origin of a tear sheet, the robot can obtain that inquiry when connected to the network at the lower level. It then converts the hash of the tear sheet being queried for to a physical representation, e.g. a UPC code or a QR code that is printed on a sheet of paper. The robot can then connect physically to the higher level network control and data repositories, and check whether the hash is present in the blockchain. The unique structure of blockchain and Merkle trees makes such searches very efficient. The answer can then be physically printed out on another sheet of paper, so that the robot can access the answer when connected to the lower security network.

While a somewhat unorthodox usage of both blockchain and multi-level security model in networks, this system enables the validation of tear sheets at a lower level of the network while preserving the strict air-gap requirements that need to exist between the networks at different level of security.

7. CHALLENGES AND OPEN ISSUES

While we have given some examples of how blockchain technology can be used to resolve some critical challenges in coalition inter-operation scenarios, several challenges remain in the implementation and attainment of a final vision. Apart from the fact that the application usage is preliminary, and one may run into unexpected challenges during the implementation of the approaches described above, some of the challenges with blockchain implementation are obvious even at this early stage.

In a typical blockchain implementation, the hash of every transaction is shared with all the participating members in the blockchain ecosystem. This leads to two issues, the first is that of the amount of bandwidth that is consumed due to all the message exchanges, and the other is the fact that all members of the coalition can count the number of transactions that are happening. The former may be an issue in environments, such as the tactical edge, where bandwidth is at a premium. The latter can be undesirable in environments where some of the coalition members are less trusted than others.

It may be possible to have implementations of the distributed ledger that are more efficient than those available in open source which take into account the special needs of the tactical environment. In the case of the software defined coalitions, it is possible to have a scenario where a separate blockchain is set up for each dynamic community of interest, thereby eliminating the need to broadcast hashes to everyone. The approach of defining specialized consensus algorithms provides a partial solution, but it may be possible to have even more efficient approaches, which would need further exploration. By exploiting the fact that each dynamic CoI would have a commander, the presence of the commander can be used to streamline some of the blockchain processes. Similarly, in the asset assignment scenario, the fact that each asset belongs to a single coalition partner may be used to improve the efficiency of the blockchain operation. The tear sheet model, which uses blockchain only as a data structure, can also be further improved upon if data can be transmitted without requiring an air-gap from the higher level of network security to a lower level of network security.

At the transport level, there is also an issue in the fact that most current blockchain implementations rely on the use of TCP. TCP is generally not suitable for a tactical edge networks because the transient connectivity of the peers often leads to numerous transmission failures, which TCP is not designed to deal with. For example, Fabric uses Google RPC (gRPC [11]), which is implemented over HTTP/2 standards, built on TCP. One potential avenue for exploration here is to investigate different gRPC bindings to more efficient protocols e.g. QUIC [12]. QUIC is a new transport which is designed to reduce latency compared to that of TCP for such environments, and has already seen some success in Android apps based on QUIC Chromium integration.

Despite these open issues which need to be explored further, our current analysis leads to the conclusion that the blockchain technologies have many interesting applications in the domain of coalition operations, and can provide elegant solutions to many tricky challenges that arise in a group of partially trusted collaborating entities.

8. ACKNOWLEDGEMENTS

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy-right notation hereon.

REFERENCES

[1] Asmare, E., Dulay, N., Lupu, E., Sloman, M., Calo, S. and Lobo, J., "Secure dynamic community establishment in coalitions," Proc. IEEE MILCOM, 1-7 (2007).

[2] Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C., "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," Proc. IEEE Security and Privacy Symposium, 839-858 (2016)

[3] Merkle, R. C., "A Digital Signature Based on a Conventional Encryption Function," Proc. Conference on the Theory and Application of Cryptographic Techniques, 369-378 (1987).

[4] Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M., "Zerocash: Decentralized anonymous payments from bitcoin," Proc. IEEE Symposium on Security and Privacy, 459-474 (2014).

[5] Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T., "A survey of software-defined networking: Past, present, and future of programmable networks," IEEE Communications Surveys & Tutorials 16(3), 1617-1634 (2014).

[6] Mishra V., Verma D., Williams C. and Marcus K., "Comparing Software Defined Architectures for Coalition Operations", Proc. International Conference on Military Communications and Information Systems, (2017).

[7] Preece, A., Gomez, M., de Mel, G., Vasconcelos, W., Sleeman, D., Colley, S., Pearson, G., Pham, T. and La Porta, T., "Matching sensors to missions using a knowledge-based approach," IEEE Intelligent Systems, 28 (1), 57-63 (2013).

[8] Cachin C., "Architecture of the Hyperledger blockchain fabric," Proc. Workshop on Distributed Cryptocurrencies and Consensus Ledgers, (2016).

[9] Calo, S., Wood, D., Zerfos, P., Vyvyan, D., Dantressangle, P., & Bent, G., "Technologies for federation and interoperation of coalition networks," Proc. 12th International Conference on Information Fusion, 1385-1392 (2009).

[10] High, R., [The era of cognitive systems: An inside look at IBM Watson and how it works]. IBM Corporation Redbooks, (2012).

[11] Google Corporation, "Google Remote Procedural Call Protocol (gRPC)," <http://www.grpc.io>.

[12] Chromium Projects, "QUIC, a multiplexed stream transport over UDP," < https://www.chromium.org/quic>.