Use of blockchains for secure binding of metadata in military applications of IoT

Konrad Wrona*†, Michał Jarosz†
*NATO Communications and Information Agency
The Hague, The Netherlands
†Military University of Technology
Warsaw, Poland

Abstract—**Blockchains offer an interesting solution to some of the security challenges encountered in military applications. They are particularly attractive in the scenarios, where it is difficult or even impossible to establish a common root of trust, e.g., in the context of civil-military collaboration, where military organizations need to build trusted information exchange infrastructure with various types of civilian governmental and non-governmental organizations, local communities, commercial companies and private persons. In our work, we discuss how blockchain can be used to store metadata describing information collected from the IoT devices owned by the federation members as well as crowdsourced from sensors belonging to private users. We present a high level architecture for a metadata binding solution using blockchains compliant with STANAG 4774 and 4778. We also describe a concrete technical solution based on Hyperledger Fabric and some of the open issues.**

I. Introduction

Blockchains, thanks to combination of its properties such as immutability of stored data and decentralization, offer an interesting solution to some of the security challenges encountered in military applications [1]. They are particularly attractive in the scenarios, where it is difficult or even impossible to establish a common root of trust [2]. This is often the case in federated environments, where participants usually belong to disjoint organizational domains, with limited mutual trust. In modern military operations, such federated scenarios are increasingly common and, moreover, often entail need for dynamic federation of coalition members, with a very limited, if any, history of earlier collaboration. Establishment of such dynamic federations can be required in support of defence operations in active conflict areas, but even more often during humanitarian assistance and disaster relief (HADR) operations in natural catastrophe areas and during counter-terrorism operations. The dynamic character and unpredictability of partnerships is especially relevant in the context of civil-military collaboration (CIMIC), where military organizations need to build trusted information exchange infrastructure with various types of civilian governmental and non-governmental organizations, local communities, commercial companies and private persons.

One of important new CIMIC use cases is establishment of improved situational awareness in smart environments, such as smart cities. These smart environments are often equipped with a large number of sensors and actuators, see e.g. [3], which can provide a valuable additional source of situational awareness as well as contribute to generating the desired physical effects in the area of operation.

In our work, we discuss how blockchain can be used to store metadata describing information collected from the IoT devices. The devices can include both the devices owned by the federation members as well as crowdsourced sensors, belonging to private users. We start with presenting a high level architecture for a metadata binding solution using blockchains and compliant with STANAG 4774 [4] and STANAG 4778 [5]. In the next section, we present a concrete technical solution for binding metadata to information received from IoT devices using blockchain. We describe a proof of concept implementation of this solution and discuss possible directions for future work.

II. Use of metadata in IoT applications

Although IoT devices and smart environments offer an attractive and promising source of additional information for enriching and refining situational awareness during military and HADR operations, there are several important challenges related to use of third party devices. The most basic challenge is related to achieving a technical interoperability, and in particular discovery of the information that can be harvested from, e.g., smart city infrastructure or crowdsourced from privately owned sensors and IoT devices. However, even if the existing information sources can be discovered, there are still important challenges related to understanding the meaning, relevance and trustworthiness of the obtained information. Availability of appropriate metadata could facilitate this process of reasoning and knowledge generation based on the information obtained from the external sources. The metadata could be used to describe both the content properties, e.g., as defined in Dublin Core or NATO Core Metadata Specifications [6], as well as context information required to assess trustworthiness and reliability of the collected data, e,g, its provenance, originators/owners identity, time of generation and device accuracy. The availability of both semantic and context metadata is critical in order to support an efficient and effective fusion of data - especially, when generating a combined operational picture, involving both internal and external information sources.

III. Security requirements

As discussed in the previous section, metadata can play an important role in enabling a more detailed operational picture and thus improving situational awareness in scenarios requiring civil-military cooperation. However, opening military systems to inclusion of external data sources opens them also to new possible vector attacks. Some of these attacks are classical security vulnerabilities, such as injection of malformed data, e.g. SQL-injection, and can be dealt with by using well-known defence techniques and secure programming practices, e.g. content validation. Nevertheless, some other attacks may focus specifically on the information layer - e.g. could aim at confusing data analytic algorithms by misrepresenting meaning of the information or deliberately or unknowingly sending incorrect values of the sensor readings. This could potentially fool the data fusion mechanisms and lead to poisoning of the overall data set used for situational awareness and generation of the the combined operational picture. In fact, this type of attack is very similar in nature to fake news attacks, which have been recently successfully performed on the global scale [7]. In order to prevent these types of attacks and ensure a trustworthy fusion of data, it is crucial to not only correctly understand what is the data about, but also to know the exact provenance of the information. These critical metadata needs to be accompanied by an appropriate guarantee of its integrity - as well as integrity of its binding to the original data object.

IV. Metadata syntax and binding mechanisms

The NATO Standardization Agreement (STANAG) 4774 describes the XML format and syntax for metadata labels. Currently, it focuses on confidentiality metadata labels that describe the collection of confidentiality elements and attributes that indicate the sensitivity of the information. These attributes include Policy Identifier, Classification, Security Category as well as Date and Time of Creation of the metadata label. It is also possible to include in the label information about the originator and provenance of the data – these can be used for further assessment of information trustworthiness and optimization of the data fusion process.

NATO STANAG 4778 describes possible ways of associating a metadata label to a data object. In particular, it introduces three different types of *bindings*, which we briefly introduce below. Each of the binding types consists of following elements: *metadata*, a *data object* and a *binding*. A single binding can be associated with several metadata and data objects (or references to these objects). Each binding can also include a *cryptographic artefact* that protects the integrity of the above elements. Examples of possible cryptographic artefacts are: a digital signature, a cryptographic digest or a message authentication code.

In the case of an *encapsulating binding*, a data object and a metadata label are located in the same binding element, for example in an XML document. Two forms of an encapsulating binding, without and with cryptographic protection of integrity and origin are depicted in Figure 1.

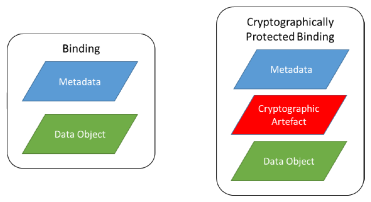

Fig. 1. Encapsulating binding

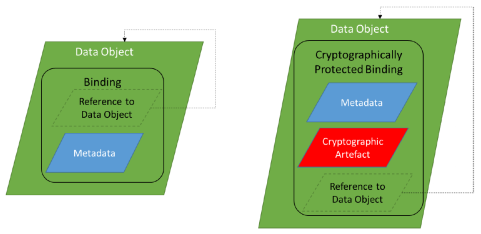

Fig. 2. Embedded binding

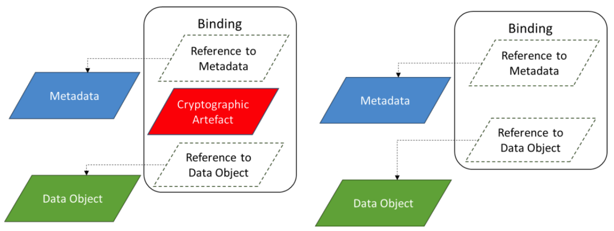

Fig. 3. Detached binding

In the case of an *embedded binding*, the binding element containing the metadata label is included in the data object. Two forms of an embedded binding, without and with cryptographic protection of integrity and origin are depicted in Figure 2.

Finally, in a *detached binding* some, or all, elements (i.e. metadata and data object) are stored separately to the binding. The binding includes only reference to this objects, and possibly a cryptographic artefact or a reference to such an artefact. This case is depicted in Figure 3.

It is important to stress that although the current edition of STANAG 4774 focuses predominantly on sensitivity markings and corresponding confidentiality labels, the techniques described in STANAG 4774 and 4778 can be used to attach any standardized metadata, including general metadata, such as described in the NATO Core Metadata Specification [6].

V. Blockchain-based IoT applications

A. Generic architecture

A generic architecture for a blockchain-based IoT application, which is also used for our blockchain-based labelling solution, is presented in Figure 4. The architecture consist of five main types of components. IoT Devices are various types of sensing devices, ranging from highly constrained low-end sensors to relatively powerful components of smart city infrastructure. We assume that most of these devices are not able to directly interact with the blockchain network -

either due to performance limitations or lack of appropriate configuration. These sensing devices communicate using a wireless interface, such as NB-IoT, LoRa, Zigbee, Bluetooth or WiFi, with an edge node, which re-formats data and annotates it with metadata in a way compliant with STANAG 4774 and suitable for inclusion in a blockchain transaction. The IoT Edge Nodes are fully functional end-users of a blockchain application, and similar to Other Federated Nodes, can send requests to read data stored in a blockchain, send a transaction to be included in a blockchain, or invoke smart contracts. The IoT Edge Node can communicate with a blockchain system via various types of connections, i.e. wired or wireless, and various protocols, e.g., HTTPS or other protocols based on TCP/UDP. It is assumed that this communication channel provides a much higher bandwidth and reliability than the communication channel used between the IoT devices and the edge node.

One of the core elements of the blockchain system is an access control module. Access control module limits access to the blockchain API. It also participates in the process of authentication of users and devices. In the case of permissionless blockchains, the access control module might enforce a null policy, i.e. it might admit all users and requests, and might solely focus on validation and sanitization of input data. The access control module is managed by a node administrator. The blockchain network provides a distributed ledger, i.e. a distributed data repository and a consensus mechanism that ensures consistency of the ledger. In Figure 4 for sake of readability, we presented only connection between the access control module and the sample nodes, whereas in the actual network each node is connected to the access control module.

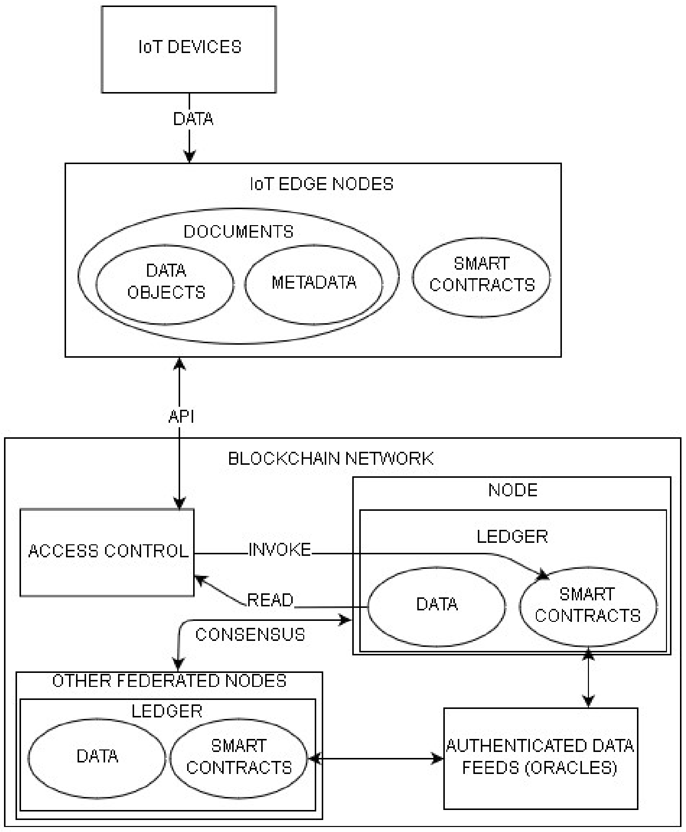

Fig. 4. Architecture of blockchain system

B. Hyperledger Fabric

1) General information: Hyperledger Fabric [8], [9] is an open-source project hosted by the Linux Foundation within the Hyperledger collaboration. The main focus of the Hyperledger Fabric is to facilitate creation of blockchain-based applications.

One of the reasons, why we chose Hyperledger Fabric for our proof-of-concept implementation, was its ability to enforce flexible security policies. The Hyperledger Fabric it is a permissioned and a private distributed ledger, which means that only authorized users have access to data and selected nodes can participate in consensus process. Moreover, the Hyperledger Fabric supports separated channels. A channel is a fully separated ledger, within which it is possible to create private data collections accessible only to the users of a specific channel. In order to enforce an adequate access control and accountability, each participant, e.g. peer, orderer, client application, administrator, is identified by a membership service provider (MSP) [10]. Default implementation relies on X.509 certificates of identity and on traditional Public Key Infrastructure (PKI), however it is possible to implement also other authorization mechanisms, e.g. providing identity protection. Every organization has at least one MSP.

Data is stored in the blockchain is in a key-value format. The Hyperledger Fabric is modular solution [11]. In particular, it is possible to change the database engine (the default one is LevelDB) and the employed consensus mechanism. This second option is important in the IoT applications, where the use of proof-of-work based consensus mechanisms might not be feasible.

Other important feature of Hyperledger Fabric is so-called chaincode. It is a code deployed on the each node. It is triggered when the user or device call a smart contract. Smart contract checks whether the conditions stored in it are met for the arguments given during the call. If not, no changes will be made. If the conditions recorded in the smart contract are met, then the result of the chaincode execution is checked with the results obtained by the other nodes in order to reach a consensus.

2) Consensus mechanism: The process of voting and reaching agreement between nodes is called consensus. The consensus is carried out automatically when the block reaches the maximum size, or after the predefine time has elapsed. Hyperledger Fabric provides two types of consensus mechanism. The SOLO consensus is used only in the development environment. The second consensus mechanism is based on Apache Kafka and it is used in a production environment. In Kafka consensus, the ordering services nodes (OSN) send transaction to Kafka cluster. Kafka start a stopwatch after receiving the first transaction and starts grouping successive transactions into the block. The block is cut either when the maximum number of transactions are reached or when the timer expires, whichever comes first. The block is sent to all OSNs and then sent to all nodes. Thus, all nodes end up with the same block recording the same transactions in the same order. In [12] authors showed that Hyperledger Fabric can handle up to 2700 transactions per second (TPS).

3) Transaction Flow: The flow of each transaction in Hyperledger Fabric consists of three phases:

1) *Proposal*: In this phase, application generate transaction proposal and send to required set of peers for endorsement. Each of this peers generates a response using chaincode (simulates the transaction execution), sign it and returns it to application. In this phase, the ledger is not modified.
2) *Ordering*: Applications sends transactions to ordering service (orderer). This peer packages transactions into blocks.
3) *Validation*: In this phase the orderer send block to all peers. Each peer process block independently. Peers verify that the transaction has been signed by the required organizations (according to endorsement policy). If this condition is met, the peer will attempt to append transaction to the ledger. This is done by checking if the current state of ledger is consistent with the state of the ledger when the proposed update is generated. After checking all transactions, the peers update the ledger. Failed transactions are not applied to the ledger, but they are retained (in blocks) for audit purposes[1]. At the end, each peer generates an event.

In Figure 5 we have presented a simplified architecture of a labelling solution using Hyperledger Fabric. Access control is carried out on two levels. The first level is provided by the data acquisition *Application*. It checks, on the basis of authentication data stored in the *Channel 2*, whether the user has access to a specific object stored in the *Channel 1*. The second level access control is based on a PKI-based mechanism implemented within HF. It controls administrators access and smart contract management, and allows nodes to participate in the network. Each node maintains two ledgers (one for each channel). Each ledger has two objects: *Data* (or *Documents*) and a chaincode (depicted as *SC*). *Ledger 1* in *Node 1* contains the same data as *Ledger 1* in *Node 2*. The same situation occurs in *Ledger 2*. Network configuration (*NC*) contains information about network configuration and network access policy. In addition, each channel has its own policies (*CC*) determining, among other things, which nodes can join a given channel. The triangular shapes represent organizations that can change policy.

VI. Metadata binding using blockchain mechanism

The use of a blockchain as a binding mechanism is depicted in Fig. 6. Blockchain binding ensures that each *Reading* element is protected against unauthorized modification. *Reading*

[1]Each peer has a PeerLedger. PeerLedger it is a mask, which indicates valid and invalid transactions.

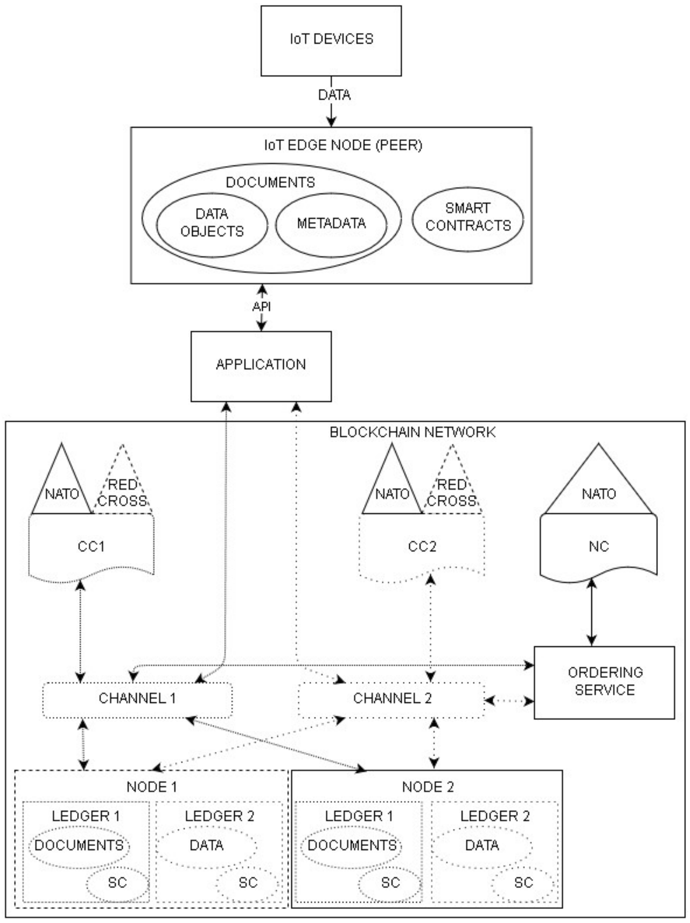

Fig. 5. Trusted metadata annotation with Hyperledger Fabric

object stores, among others, information about location of data (*dataReference*, hash of stored data, data encoding and type of data. Each *Reading* object has link to *Reading Metadata* that stores information about *Reading* such as date and time of its creation as well as location where the data was collected. Our solution allows user to add several links to *Reading* objects in the *Sensor* object. The *Sensor* object stores information about the particular sensor that provides *Readings*. The *Sensor Metadata* object stores attributes about the sensor, which can be used during a trustworthy data fusion process. Each of this object has an unique ID. We have created a sample binding application using the Hyperledger Fabric. Our test scenario has involved a federated information sharing environment with two organizations, each of them with two peers. We have created two channels. The first channel was responsible for metadata binding mechanism. In this channel we have created two chaincodes, first was responsible for handling *Sensor* and *Sensor Metadata* objects, second for handling *Reading* and *Reading Metadata* objects. The *Sensor* chaincode has four functions: 1) add Sensor and Sensor Metadata; 2) query value for a specific key stored in the ledger; 3) add a ReadingID to an existing Sensor and 4) modify Sensor Metadata. The key is a unique string that uniquely identifies the object. Chaincode

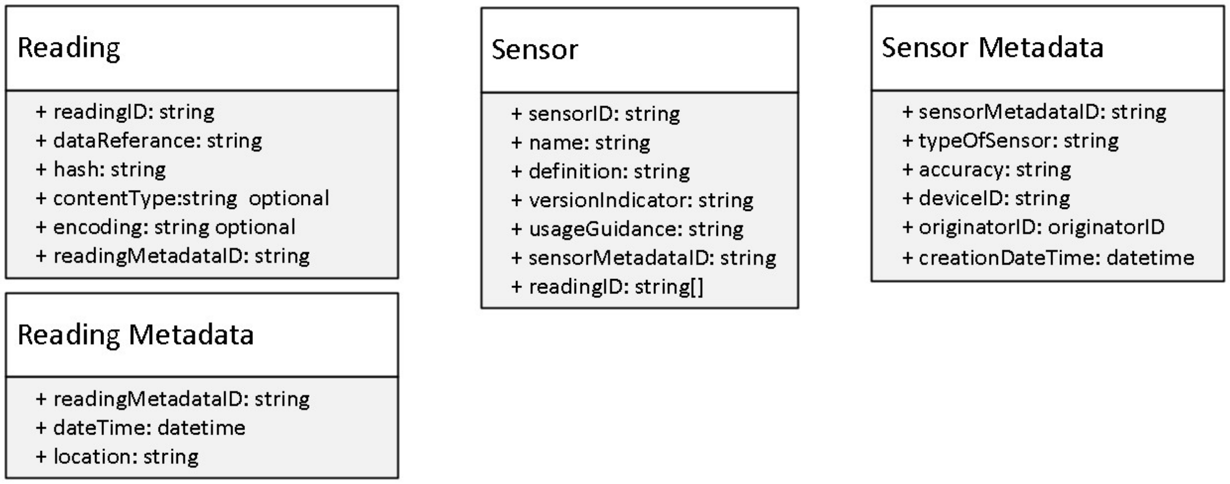

Fig. 6. Structure of blockchain objects

for Reading and Reading Metadata has two functions: 1) add new Reading and Reading Metadata; and 2) get a value for a specific key. Each of these chaincodes has been written in Go.

The second channel is responsible for storing information about users, including their access rights to specific objects stored in the first channel. We have also created an application in Python for Raspberry Pi that reads data from a temperature and a humidity sensors. The data are stored in text files on a shared disk. Because Hyperledger Fabric does not yet support REST API, we have written our own application using Fabric-GO-SDK [13] that receives data from IoT devices/IoT edge nodes and invokes a chaincode to add a data object to a blockchain. It is one of the situations presented in Section V. The process of adding new data to the blockchain consists of two phases. In the first phase, the user add the sensor to the blockchain, while in the second phase the device stores data in text file on the separate server and add a *Reading* object to the blockchain. The data in the blockchain is stored in JSON format.

Each type of binding introduced in Section IV has advantages and disadvantages in the context of blockchain applications, which are discussed briefly below.

Embedded binding mechanism is least suitable for use with a blockchain, as metadata is stored inside of a data object, e.g. a Word document. Therefore, the whole document needs to be processed in order to check a metadata label.

Encapsulating binding is less suitable for use with blockchain. Since the data object is stored with metadata in the same blockchain, the large amount of data might need to be stored in the blockchain, introducing a large overhead and limitations in a capacity of transactions. The advantage of encapsulating binding when used with blockchain is immutability of both data and metadata, and that only one blockchain is needed to store all objects.

Detached binding mechanism is in our opinion best suited for use with our current blockchain-based solution. In the case of detached binding, a document includes a reference to a data object and metadata. Advantage of this solution is ability to limit the required storage space in the blockchain and thus an improved scalability and performance. Data object can be stored in different places. Please note the data stored beyond blockchain, so a separate mechanism needs to be used to protect data confidentiality, integrity and availability, although integrity violation can be detected thanks to the data cryptographic hash stored in the blockchain. We have chosen this type of binding for our initial implementation, because IoT devices can produce a large amount of data.

VII. Related work

A survey of various proposed distributed ledger technologies, including both blockchains and alternative solutions can be found in [14]. Some selection criteria for choosing a suitable blockchain solution have been investigated in [15]. An example of system that can be used for providing an authenticated data feed for smart contracts has been presented in [16]. A survey of various proposed blockchain applications can be found in [17], however it does not cover any of possible military applications. A blockchain-based data management system has been proposed in [18].

A Hyperledger Fabric implementation relying on Intel SGX platform and providing a secure execution environment for smart contracts has been proposed in [19]. Advantage of using Hyperledger Fabric in IoT are discussed in [20]. A comparison of Hyperledger Fabric, Ethereum and IOTA can be found in [21].

Several different types of architecture styles for blockchain IoT applications have been discussed in [22] - our approach can be classified as a variant of the distributed things architecture according to the taxonomy presented there. Various types of distributed ledger platforms, which can be used in IoT applications, are discussed in [23]. Various applications of blockchain technology in context of smart cities are presented in [24]. More general discussion of applicability of blockchain to IoT, including various use cases can be found in [20], [25]–

[27]. Performance of different consensus algorithms in context of IoT is evaluated in [28]. A more security-focused reviews of the blockchain technology in context of IoT can be found in [29]–[31]. Specific issues related to use of blockchain for identity and access management are discussed in [32]–[34].

VIII. Conclusions and future work

In this article we have presented an architecture for a blockchain-based cryptographic binding of metadata to sensor data collected in context of civil-military cooperation in smart environments. We have introduced Hyperledger Fabric as a possible implementation framework for our solution. Hyperledger Fabric is currently one of the most complete and widely used frameworks for developing blockchain-based applications. STANAG 4774 and 4778 have been recently adopted by NATO in order to ensure a coherent labelling of information. The different types of binding described in STANAG 4778 have some specific advantages and disadvantages in the context of blockchain-based binding of metadata to data objects. We have presented an initial solution for storing sensor data and associated metadata in a Hyperledger Fabric. Our implementation confirms feasibility of our approach. In the future we plan to perform a larger scale performance tests and to integrate our new binding profile with a proof-of-concept implementation of a situational awareness system for HADR and defense operations in intelligent environments, such as smart cities. In particular, in our current implementation IoT devices can be connected to the blockchain only through an application written in Python, installed on an IoT Edge Node or an IoT device. We are working on client implementation written in C and on improving our application for translation of data formats and requests coming from various types of IoT devices.

Acknowledgment

This work was partially performed within the research project SEMACITI sponsore by Polish Ministry of Defense.

References

[1] K. Wrona and M. Jarosz, “Does NATO need a blockchain ?” in *MILCOM Military Communications Conference*, 2018, pp. 667–672.

[2] K. Wüst and A. Gervais, “Do you need a Blockchain ?” in *Crypto Valley Conference on Blockchain Technology*, 2018, pp. 45–54.

[3] D. Bruneo, S. Distefano, M. Giacobbe, A. Longo Minnolo, F. Longo, G. Merlino, D. Mulfari, A. Panarello, G. Patanè, A. Puliafito, C. Puliafito, and N. Tapas, “An IoT service ecosystem for Smart Cities: The #SmartME project,” *Internet of Things*, vol. 5, pp. 12–33, 2019.

[4] NATO Standardization Office, “Confidentiality metadata label syntax,” NATO Standardization Office, NATO Standardization Agreement (STANAG) 4774, 2016.

[5] ——, “Metadata binding mechanism,” NATO Standardization Office, NATO Standardization Agreement (STANAG) 4478, 2018.

[6] NATO Consultation Command and Control Board (C3B), “NATO Core Metadata Specification (NCMS) Version 1.0,” NATO, Brussels, Belgium, Tech. Rep., 2015.

[7] I. Allan, J. Althuis, A. Averin, G. Conci, S. Dooley, E. Duffy, D. Gray, L. Haiden, M. Ilbury, N. Kantovich, C. Mcmanus, E. Moore, K. Ranautta-sambhi, and S. Strand, *Fake News: A Roadmap*, J. Althuis and L. Haiden, Eds. Riga: NATO Strategic Communications Centre of Excellence, 2018.

[8] V. Dhillon, D. Metcalf, and M. Hooper, “The Hyperledger Project,” in *Blockchain Enabled Applications*, 2017, pp. 139–149.

[9] “Hyperledger Fabric,” https://www.hyperledger.org/projects/fabric.

[10] N. Haur, L.Desrosiers, V. Ramakrishna, P. Novotny, S.A.Baset, and A. O’Dowd, *Building decentralized applications with Hyperledger Fabric and Composer*. Packt Publishing, 2018.

[11] https://hyperledger-fabric.readthedocs.io/en/latest/.

[12] P. Thakkar, S. N. N, and B. Viswanathan, “Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform,” 2018. [Online]. Available: https://arxiv.org/pdf/1805.11390.pdf

[13] https://github.com/hyperledger/fabric-sdk-go.

[14] N. E. Ioini and C. Pahl, “A Review of Distributed Ledger Technologies,” in *Proc. of the OTM Conferences*. Springer International Publishing, 2018, pp. 277–288.

[15] C. Maple and J. Jackson, “Selecting Effective Blockchain Solutions,” in *Proc. of the Euro-Par 2018 Workshops*. Springer International Publishing, 2019, pp. 392–403.

[16] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, “Town Crier : An Authenticated Data Feed for Smart Contracts,” in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 270–282.

[17] F. Casino, T. K. Dasaklis, and C. Patsakis, “A systematic literature review of blockchain-based applications: current status, classification and open issues,” *Telematics and Informatics*, vol. 36, pp. 55–81, 2019.

[18] L. Zhu, Y. Wu, K. Gai, and K. K. R. Choo, “Controllable and trustworthy blockchain-based cloud data management,” *Future Generation Computer Systems*, vol. 91, pp. 527–535, 2019.

[19] M. Brandenburger, C. Cachin, R. Kapitza, and A. Sorniotti, “Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric,” 2018. [Online]. Available: http://arxiv.org/abs/1805.08541

[20] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, “Blockchain’s adoption in IoT: The challenges, and a way forward,” *Journal of Network and Computer Applications*, vol. 125, pp. 251–279, 2019.

[21] M. Pustišek and A. Kos, “Approaches to Front-End IoT Application Development for the Ethereum Blockchain,” *Procedia Computer Science*, vol. 129, pp. 410–419, 2018.

[22] C.-F. Liao, C.-C. Hung, and K. Chen, “Blockchain and the Internet of Things: A Software Architecture Perspective,” in *Business Transformation through Blockchain*, H. Treiblmaier and R. Beck, Eds. Springer, 2019, pp. 53–75.

[23] D. Burkhardt, P. Frey, S. Hiller, A. Neff, and H. Lasi, “Distributed Ledger Enabled Internet of Things Platforms: Symbiosis Evaluation,” in *Business Transformation through Blockchain*, 2019, pp. 77–118.

[24] L. Shuling, “Application of Blockchain Technology in Smart City Infrastructure,” in *IEEE International Conference on Smart Internet of Things Application*. IEEE, 2018, pp. 276–282.

[25] T. M. Fernández-Caramés and P. Fraga-Lamas, “A Review on the Use of Blockchain for the Internet of Things,” *IEEE Access*, vol. 6, pp. 32 979–33 001, 2018.

[26] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On blockchain and its integration with IoT. Challenges and opportunities,” *Future Generation Computer Systems*, vol. 88, no. 2018, pp. 173–190, 2018.

[27] J. Wang, M. Li, Y. He, H. Li, K. Xiao, and C. Wang, “A Blockchain Based Privacy-Preserving Incentive Mechanism in Crowdsensing Applications,” *IEEE Access*, vol. 6, pp. 17 545–17 556, 2018.

[28] S. Zoican, M. Vochin, R. Zoican, and D. Galatchi, “Blockchain and Consensus Algorithms in Internet of Things,” *International Symposium on Electronics and Telecommunications ISETC 2018*, pp. 1–4, 2018.

[29] M. A. Khan and K. Salah, “IoT security: Review, blockchain solutions, and open challenges,” *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[30] M. Banerjee, J. Lee, and K. K. R. Choo, “A blockchain future for internet of things security: a position paper,” *Digital Communications and Networks*, vol. 4, no. 3, pp. 149–160, 2018.

[31] D. Minoli and B. Occhiogrosso, “Blockchain mechanisms for IoT security,” *Internet of Things*, vol. 1-2, pp. 1–13, 2018.

[32] N. Kshetri, “Can Blockchain Strengthen the Internet of Things?” *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[33] M. Nuss, A. Puchta, and M. Kunz, “Towards Blockchain-Based Identity and and Access Management for Internet of Things in Enterprises,” in *Proc. of the TrustBus*. Springer International Publishing, 2018, pp. 167–181.

[34] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, “Bubbles of Trust: A decentralized blockchain-based authentication system for IoT,” *Computers and Security*, vol. 78, no. 2018, pp. 126–142, 2018.