Proceedings of the of the 17th Conference on Computer Science and Intelligence Systems pp. 627–634
DOI: 10.15439/2022F173
ISSN 2300-5963 ACSIS, Vol. 30

Secure Onboarding and Key Management in Federated IoT Environments

Krzysztof Kanciak
Cybernetics Faculty
Military University of Technology
Warsaw, Poland
krzysztof.kanciak@wat.edu.pl

Konrad Wrona
NATO Cyber Security Centre /
Military University of Technology
The Hague, Netherlands / Warsaw, Poland
konrad.wrona@[ncia.nato.int,wat.edu.pl]

Michał Jarosz
Cybernetics Faculty
Military University of Technology
Warsaw, Poland
michal.jarosz@wat.edu.pl

***Abstract*—Many high-impact Internet of Things (IoT) scenarios, such as humanitarian assistance and disaster relief, public safety, and military operations, require the establishment of a secure federated IoT environment. One of the critical challenges in the implementation of federated IoT solutions involves establishing a secure and authenticated key management mechanism. We propose and validate in a laboratory environment a novel federated IoT onboarding and key management solution. Our dl-mOT protocol integrates an efficient identity-based modified Okamoto-Tanaka (mOT) protocol with a distributed ledger in order to establish an anchor of trust between federation members.**

I. Introduction

MANY high-impact Internet of Things (IoT) scenarios, such as Humanitarian Assistance and Disaster Relief (HADR), public safety, and military operations, require the establishment of a secure federated IoT environment. Such a federation may involve entities with limited a priori trust relationships and generally rely on the integration and reuse of resources belonging to individual partners.

As an example, we consider a natural disaster, such as an earthquake, tornado, or flood, that affects a smart city. In such a scenario, military forces can be requested to assist local government agencies in delivering essentials, such as food and medical supplies, as well as medical and search and rescue support. One of the important operational priorities is to increase the number of information sources available to improve Situational Awareness (SA). To optimize the efficiency and effectiveness of their efforts, first responders can rely on data retrieved from the surviving smart city infrastructure. Such infrastructure may comprise sensors, actuators, and communication equipment, for example, traffic light posts with cameras for traffic flow monitoring, pollution and weather sensors, smart transportation networks, and smart power grids. To augment these capabilities, which can be degraded by a disaster, military forces can also deploy their own sensors at key locations, the data from which can be shared with local authorities and civilian responders and used to establish a common SA [1].

II. Federated IoT Environment

An example of a federated IoT environment that was proposed for use within military and HADR operations is presented in Fig. 1.

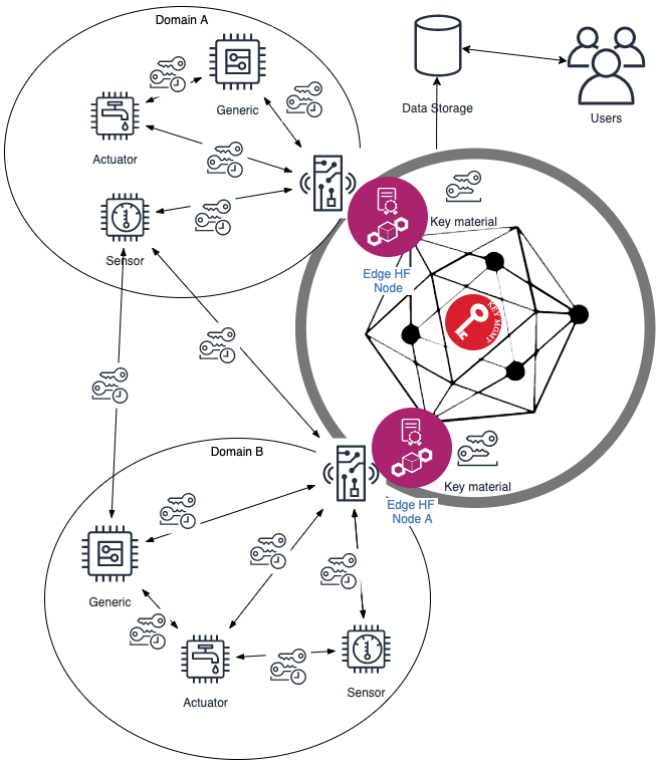

Fig. 1. Blockchain-based key management solution for IoT.

We can identify four types of components of such a federated IoT system:

1) IoT devices: Sensors and actuators, belonging to and operated by a specific organization and thus constituting a single security domain.
2) Edge nodes: These are gateway or sink nodes, facilitating communication within a single security domain and between devices belonging to different security domains. Edge nodes can also function as distributed ledger nodes.
3) Distributed ledger nodes: These are nodes participating in a permissioned distributed ledger. They represent different organizations participating in the federation. In the case of organizations operating an own IoT system, an

edge node can also play the role of a distributed ledger node. In the presented application, DL is responsible for authentication, authorization and for sharing the key for communication with IoT devices with each other. The ledger stores all the information necessary to perform the above operations. Each data saving transaction must be carried out only after fulfilling the conditions specified in the smart contract. In the case of Hyperledger Fabric, the smart contract is called chaincode. Because we use Hyperledger Fabric in our work, we will also rely on the naming convention used there.

4) End-user services: These are the primary consumers of sensor data and actuation capability offered by the IoT devices. Interaction between services and IoT devices is mediated via the federated distributed ledger. End-user services expose capabilities offered by a federated IoT system to the end-users.

In such an environment, IoT devices can communicate directly with each other, as well as with the edge nodes. A specific organization owns each IoT device, but to provide the required resilience and effectiveness of operation, it is desirable that once a federation is established, a device can communicate with any edge node belonging to the federation. Furthermore, a federation-wide federated access control policy can be maintained to define authorized direct communication patterns between IoT devices within and between organizations.

During the operational phases, IoT devices send data and requests to the edge node that acts as a mediator between IoT devices and distributed ledger nodes. Authorization to read and write data to the ledger is obtained by execution of a smart contract. The smart contracts can also be used to perform some processing of the data, e.g., data filtering, aggregation, or labeling.

III. Challenges

The most critical challenges related to the interconnection of civilian and military communications and information systems (CIS) are related to security. In particular, the federated CIS needs to support controlled and timely information sharing between federation partners, meeting both stringent need-to-know constraints defined by the military partners, as well as responsibility-to-share requirements of effective execution of joint emergency response activities. Ensuring interconnection of military systems, usually compliant with specific military standards such as NATO Standardization Agreements (STANAGs), with their civilian counterparts, largely relying on open or commercial standards, introduces another layer of complexity and specific challenges. However, these interoperability issues are likely to decrease in the future due to an increasing focus on dual use of many of the new technologies, such as 5G, and an increasing focus of the military on cost-effectiveness of CIS implementation through the use of commercial-of-the-shelf (COTS) technologies and related civilian and open standards.

Federated emergency response and disaster recovery operations can be conducted in a broad spectrum of settings: in form of a peacetime support to civilian authorities as well as response to terrorist activities and humanitarian disasters in conflict regions. Therefore, in context of the CIS security, we need to consider presence of a powerful and technically sophisticated adversary, interested in disrupting response or using it as an opportunity to infiltrate or damage CIS of military partners. Such an attacker can compromise an arbitrary number of devices belonging to a specific federated organization. In the context of resilience to attacks, we differentiate between two types of devices. First type consists of devices equipped with a secure element (or a trusted processing module) that can provide reasonable resistance to access to secret data stored in the device, so that it can be assumed that for duration of a specific military operation, the data remains secret. Second type of devices is not equipped with any hardware security mechanisms - therefore, once compromised, any secret data stored at the device can be read and modified by the attacker. We assume that although attacker can compromise an arbitrary number of IoT devices, edge nodes and distributed ledger nodes belonging to specific federated organization, the attacker cannot compromise majority of the federated organizations. In particular, we assume that an attacker is not able to control majority of the distributed ledger nodes, representing different federated organizations, and thus a secure consensus and secure execution of distributed application (or so-called chain code) within distributed ledger is possible. Furthermore, we assume that although an attacker can inject malicious input data from the compromised nodes but all data is verified at the distributed ledger by means of smart contracts before it is written to the ledger and also before it is read.

IV. Security Requirements

The security mechanism should satisfy several specific requirements of the federated CIMIC. In federated operations, exact trust relationships between federation members might not be known in advance of device initialization. Therefore, a device might need to be re-authorized for operation within the specific federated environment and reconfigured to enable end-to-end encrypted information sharing between a sensor or actuator of one nation and a command and control system of another nation. The proposed mechanisms must be able to function in an adversarial environment, where static or cloud infrastructure might not be accessible for prolonged periods of time. To support IoT devices, security mechanisms must be efficient with respect to computing power and memory required. Similarly, they must scale well to scenarios that involve hundreds of devices and adapt to different communication patterns. For extremely constrained devices or when modification of device configuration is not possible, the use of an edge node or a digital twin for security adaptation should be considered in order to ensure secure integration with data consumer applications. Finally, we also need to ensure some typical security requirements. Perfect forward secrecy stipulates that compromise of a session secret shall not affect

the security of any previous or future sessions. Furthermore, compromise of any device and a master secret stored on the device should not affect the security of the system as a whole and the security of other devices. The system design should also support a change of cryptographic mechanisms, providing the ability to withstand new and emerging threats, such as quantum computing.

One of the critical challenges when implementing federated IoT solutions consists of bootstrapping and maintaining the system's security. To ensure the confidentiality and integrity of communication channels between IoT devices, a secure and authenticated key management mechanism must be implemented. Such a key management mechanism needs to fulfill several specific requirements:

1) Federated operations: The mechanism shall be compatible with federated operations, where the exact trust relationships between federation members might not be known in advance of device initialization. Therefore, there might be a need for a device to be re-authorized for operation within the specific federated environment.
2) Resilience and fault tolerance: The mechanism should function also in adversary environment, where some of static or cloud infrastructure is not directly accessible from an IoT device.
3) Support for constrained devices: The part of the key management mechanisms executed at the end-devices should be computationally efficient with respect to required computing power and memory.
4) Scalability: the mechanism needs to scale well to scenarios involving potentially hundreds of devices and adapt to different communication patterns, including both fully distributed scenarios and edge scenarios where end-devices communicate to a gateway connected to a backbone network.
5) Perfect forward secrecy: compromise of a session secret does not affect the security of any previous and future sessions
6) Robustness against compromise of individual devices: compromise of any device and of a master secret stored in the device should not affect the security of a system as a whole and security of other devices. This implies in particular that any potential key generation and device authentication authority needs to be implemented using a threshold-based approach.
7) Cryptographic agility: The system should support the change of cryptographic mechanisms, providing the ability to withstand new and emerging threats, such as quantum computing.
8) Increased security guarantees for key material: none of the federated organizations is able to obtain the cryptosystem master key.

V. DL-MOT: DISTRIBUTED LEDGER IMPLEMENTATION OF THE MODIFIED OKAMOTO-TANAKA PROTOCOL

To meet the challenges defined in Section III, we propose a novel key management solution for IoT devices that integrates with a permissioned distributed ledger. The integration of the distributed ledger is a novel element that provides increased robustness against attacks on key generation and authentication authority.

In an IoT system, there are substantial differences between the computational capabilities of various devices. Therefore, our design aims at optimizing computational overhead induced on constrained IoT devices in exchange for much higher overhead on the edge nodes side, that are significantly more powerful.

The proposed solution is based on the Okamoto-Tanaka identity-based key management protocol, initially proposed in [2] and further extended in [3] and [4]. We further modify the protocol by implementing a distributed key generation authority based on a distributed ledger. Integration of a distributed ledger into identity-based cryptographic solutions can be seen as a more general and reusable security pattern; its applications in the context of privacy protection have been discussed in [5].

Since the concept of identity-based public-key cryptography was introduced in [6], many identity-based authenticated key exchange protocols have been proposed [7]–[9]. Most of the proposed protocols rely on elliptic curve pairings that are computationally very expensive and hard to implement in constrained devices. In the context of an IoT, a one-round ID-based key exchange with forward secrecy over the Rivest-Shamir-Adelman algorithm (RSA) group is a much more compelling choice. Moreover, although in data-centric federated systems, attribute- or identity-based encryption can be used to enable an efficient and effective enforcement of access control policies through data encryption, there is an inherent problem with pairing-based cryptography: There is no way to perform pairing operations (to generate private keys) in a decentralized and accountable manner. Therefore, it is difficult or impossible to maintain the principles of zero-trust architecture in the system.

When it is possible to securely store key material on an IoT device, e.g. using a Trusted Processing Module (TPM), we propose using *dl-mOT*, an enhanced version of the modified Okamoto-Tanaka (mOT) protocol [3]. The mOT protocol enables two parties to use their identities to establish their common secret keys without sending and verifying public key certificates. We further enhance the mOT protocol by integrating it with a distributed ledger. In this way, we address some inherit weaknesses related to the use of identity-based cryptography. In particular, we mitigate key, escrow, remove a single point of failure, and add strong accountability for key generation events.

The advantage of the dl-mOT protocol is its computational and communication efficiency and ease of implementation. The protocol can be implemented with short exponents, e.g., 160-bit exponents with a 1024 bit modulus. Moreover, operating over an RSA group enables the implementation of multiparty computation protocols, such as distributed exponentiation and multiparty generation of an RSA modulus.

Another essential feature of the dl-mOT protocol is the

support for perfect forward secrecy (PFS) [10] that was identified as one of the key security requirements in our federated environment. Perfect forward secrecy means that the leakage of a long-term key used by an entity does not compromise the security of session keys established by that entity. Moreover, dl-mOT allows spontaneous decentralized device-to-device interactions, without any request to the edge node or distributed ledger. Edge nodes are only involved in secure onboarding of the sensor into the federation. Device bootstrapping is a subprocess of onboarding and requires just enough information exchange between a device and the network to establish a secure channel.

A. Onboarding

In the mOT protocol, a Key Generation Center (KGC) chooses the RSA parameters $N = pq$, exponents d, e, and a random generator g of the cyclic subgroup of quadratic residues QR_N. The parameters p, q, d, and e are random and safe primes, that is, a prime p is safe iff $\frac{p-1}{2}$ is also a prime. KGC publishes values N, e, g, as well as two hash functions H and H' as a set of public parameters P. During the onboarding phase, every device receives from the KGC a unique identity id_D and a corresponding private key $S_D = H(id_D)^d \mod N$. It is important to note that the mOT software requires the storage of secure private keys on the sensors, which is not always possible.

B. Operational phase — Point-to-point communication scenario

Communicating devices are already onboard, which means that they have their identities and secrets $S_A = H(id_A)^{d_i} \mod N_i$. Now, a dynamic asynchronous secure channel may be established. In the key agreement phase, devices A and B choose ephemeral private exponents x and y, respectively. Bar notation denotes single domain keys, and hats distinguish multi-domain keys establishment. Each device can calculate

$$A \xrightarrow{\quad \alpha = g^x S_A \mod N \quad} B$$

$$A \xleftarrow{\quad \beta = g^y S_B \mod N \quad} B$$

Fig. 2. mOT key agreement

the mOT session key in the following way:

$$\bar{K_A} = (\beta^e / H(id_B))^{2x} \mod N, \tag{1}$$

$$\bar{K_B} = (\alpha^e / H(id_A))^{2y} \mod N. \tag{2}$$

Both values are equal since:

$$\begin{aligned} \bar{K_A} &= (\beta^e / H(id_B))^{2x} = \\ &= \left(\left(g^y H(id_B)^d \right)^e / H(id_B) \right)^{2x} = \\ &= \left(g^{ey} H(id_B)^{ed} / H(id_B) \right)^{2x} = \\ &= \left(g^{ey} H(id_B)^1 / H(id_B) \right)^{2x} = g^{2xye} = \bar{K}. \end{aligned} \tag{3}$$

Analogically, $\bar{K_B} = \bar{K}$. The mOT session key K is established with a key derivation function H':

$$K = H'(\bar{K}, id_A, id_B, \alpha, \beta). \tag{4}$$

C. Multi-domain KGC setting

In our setting, each IoT device belongs to an organization that operates its own security domain and the KGC. In a federated IoT environment, devices from different organizations, and thus belonging to different security domains, should be able to execute the key agreement protocol and communicate.

Each organization i operates its own KGC_i with public parameters P_i, as previously described, and secret key d_i, such that $e_i d_i = 1 \mod \phi(N_i)$.

The device A belonging to the organization operating KGC_i receives during onboarding a secret $S_A = H(id_A)^{d_i} \mod N_i$ The identity of the device consists of $H(id_A)$ and P_i. Similarly, the device B belonging to the organization that operates KGC_j has a secret $S_B = H(id_B)^{d_j} \mod N_j$ with public parameters P_j. Consider

$$E = \mathrm{lcm}(e_1, e_2) \tag{5}$$

$$\hat{g} = (g_i \mod N_i, g_j \mod N_j) \tag{6}$$

$$\hat{N_i} = (N_i^{\frac{E}{e_i}} \mod N_i, 1 \mod N_j) \tag{7}$$

$$\hat{N_j} = (1 \mod N_i, N_j^{\frac{E}{e_j}} \mod N_j) \tag{8}$$

Both communicating sensors compute a pair of values $\mod N_i N_j$:

$$\hat{S_A} = (S_A \mod N_i, 1 \mod N_j) \tag{9}$$

$$\hat{S_b} = (1 \mod N_i, S_B \mod N_j) \tag{10}$$

respectively. Device A chooses an ephemeral random integer x and computes

$$\hat{X} = \hat{g}^x \mod N_1 N_2 \tag{11}$$

and sends to B value

$$\hat{\alpha} = \hat{X} \hat{S_A} \mod N_1 N_2. \tag{12}$$

The device B chooses a random secret y and sends to A the value:

$$\hat{\beta} = \hat{Y} \hat{S_B} \tag{13}$$

where

$$\hat{Y} = \hat{g}^y. \tag{14}$$

Finally, A computes the shared secret $\hat{K}$:

$$\begin{aligned} \hat{K} = (\frac{\hat{\beta}^E}{\hat{B}})^{2x} = (\frac{\hat{Y}^E \hat{S_B}^E}{\hat{B}})^{2x} = \hat{g}^{2xyE} (\frac{\hat{S_B}^E}{\hat{B}})^{2x} = \\ = \hat{g}^{2xyE} \mod N_1 N_2. \end{aligned} \tag{15}$$

The device B performs an analogous computation. Similarly as within a single organization, at the end, both devices make a hash of the shared secret and the identities of both parties:

$$K = H'(\hat{K}, id_A, id_B, \hat{\alpha}, \hat{\beta}). \tag{16}$$

The boot-strapping process (considered as a subprocess of onboarding) can be repeated during the lifetime of a device and requires direct communication to the edge node and identity registration in the distributed ledger.

D. Multiparty private exponentiation

In the scenarios presented, the KGCs are the only parties that own the master secrets of the cryptosystems. When onboarding an IoT device within a domain, an exponentiation using a secret exponent known to the KGC is required. With the help of a bit-decomposition [11], we have constructed a constant-round protocol for multiparty private exponentiation where several KGCs participate in secret key generation (since attaching a new device implies private key generation which is single integer exponentiation). Multiparty exponentiation in the IoT device setup phase eliminates the single point of storing master secrets and supports zero-trust architecture principles. We assume that federated organizations share the domain master secret and perform their part of the exponentiation process to achieve strong accountability of the onboarding process and to weaken requirements for trust into edge nodes. The bit-decomposition private exponentiation is computationally expensive and a significant amount of research has been devoted to improve its performance [12], [13]. However, in our scenario, it is performed by relatively powerful KGC nodes and only during the onboarding of an IoT device. The classical approach to distributed modular exponentiation on arithmetic circuits relies on bit-decomposition, which was first proposed in [11]. To achieve an exponentiation protocol with a public base b and secret exponent e, the authors of [11] propose a method for securely bit-decomposing the inputs (bd) secretly shared l-bit exponent e into bits, using the so-called fan-in multiplications. We assume that there exists a function:

$$[e]_{bits} bd([e]) \tag{17}$$

that receives a secret shared input $[e]$ and returns its shared bit-decomposition $[e]_{bits}$, so that it produces l shares:

$$([a_0], [a_1], ..., [a_{l-1}]) \tag{18}$$

where $a_i \in \{0, 1\}$. The final result of the exponentiation is then the product of the decomposed multiplications:

$$\prod_{i=0}^{l-1} ([a_i] b^{2^i} + [1] - [a_i]). \tag{19}$$

The identity of an IoT device is public, while its private key is generated using KGC secrets and multiparty private exponentiation. The objective is to improve the auditability of the onboarding process (or bootstrapping, understood as a subprocess of onboarding) and make it impossible for a single compromised or malicious KGC node to add devices to the domain. This was achieved through integration of bit-decomposition with a distributed ledger, ensuring that every decomposed multiplication, performed in support of private exponentiation, left a trail on the distributed ledger. It means that federated organizations share their KGC secrets, and each execution of multiparty exponentiation is registered in distributed ledger so that it is known among federation which organizations took part in the sensors onboarding process.

E. Use of trusted execution environment

In our initial proof-of-concept KGCs master-secrets escrow problem was solved by encapsulation into Intel Software Guard Extensions (SGX) enclave. Intel SGX is an implementation of the concept of a trusted execution environment in Intel CPUs that allows users to define secure enclaves. Secure enclaves are regions of memory whose content is protected and unable to be read or saved by any process outside the enclave itself. Combining chaincode with SGX makes it possible to run applications that demand privacy, such as distributed exponentiation combiners, which was first proposed in [14]. In our initial design, a KGC enclosed in a secure enclave acted as an oracle for distributed ledger chaincode, and there was no risk of keys compromise, but still the enclaves where only parties where keys were stored and enclave availability was necessary. But only the master-secrets distribution and multiparty exponentiation weakened the nodes availability requirement sufficiently, so that zero-trust architecture principles can be satisfied.

Intel SGX comes with two advantages. The first is that enclaves provide confidentially and integrity of any code and data inside the enclave. The second advantage of enclaves is a mechanism that allows remote parties to verify the identity of the enclaves via a process called attestation. Intel SGX provides applications with the ability to create areas in memory called enclaves that provide confidentiality and integrity for the code/data running inside it, even in the presence of buggy or malicious privileged software or actor. Attestation is a process that allows remote parties to verify the identity of a piece of software. The remote party can use Intel's SGX attestation hosted service to verify the quote before provisioning any secrets into the enclave. Intel SGX also supports cryptographic binding of secrets to an enclave, the so-called *sealing*. Any enclave can use a hardware-generated key, called a sealed key, to encrypt the secrets with the key. The key is available only to the owner enclave running on the same platform. This is crucial for key material management.

VI. Performance discussion

An important practical question is the prediction of the performance of the dl-mOT in various configurations of a federated IoT environment. The main aspects of performance analysis are related to the overheads introduced by onboarding, key management, and recording of transactions in the distributed ledger.

A. Onboarding overhead

dl-mOT protocol requires a secure channel or controlled environment for the onboarding phase. During device onboarding, an RSA group element, that is, a 1024-bit private key, and a hash result, that is, a 256-bit value, must be transmitted. These values (in total 160 bytes) must be stored securely on

the sensor device. There is no computational overhead for the sensor during the onboarding phase. However, there is a computational cost induced by a device onboarding on the KGC side. In the single-domain scenario, this computational overhead comprises one exponentiation and two hash operations. In a distributed multiparty scenario, the computational cost is significant, as described in V-D, but all computations occur on nodes without significant computational restrictions.

B. Key management overhead

The communication overhead of the operational phase of dl-mOT is as in the original Diffie-Hellman protocol and requires two messages with a single RSA group element. After these messages have been exchanged, the symmetric key with full Perfect Forward Secrecy against active attackers is established and remains confidential to all parties, including the federated organization and KGC. Furthermore, due to the identity-based properties of mOT, there is no communication overhead related to certificate transmission that is typical for the authenticated Diffie-Hellman (DH) key exchange protocol. The computational cost of key establishment using the dl-mOT protocol, when used with short exponents, is very low. dl-mOT is more efficient than any RSA-based key agreement protocol and any authenticated DH protocol over Z_p^* for large prime and much more efficient than any of the ID-based protocols based on elliptic curve pairings. The dl-mOT protocol is less efficient than protocols using elliptic curves, but only for keys longer than 2048 bits.

C. Distributed ledger overhead

In the dl-mOT scenario considered, a distributed ledger is used to generate identity-based keys for IoT devices, which can be further used for secure communication with the distributed ledger nodes and the IoT peer nodes. Our target environment consists of a distributed ledger based on Hyperledger Fabric. Furthermore, we assume the use of a policy that specifies that any two organizations are sufficient to sign the transaction. Our prediction is that more nodes result in shorter delay times; that is, the median delay related to the generation of private keys and the recording of public key parameters in the distributed ledger is shorter for more nodes.

The performance of the dl-mOT solution directly depends on the performance and scalability of the distributed ledger, and in the case of our design on Hyperledger Fabric performance, which acts as a trustworthy storage for validated and reliable data.

When writing data to the ledger, the number of requests (or transactions) that can be processed increases with the number of nodes that participate in the ledger from each organization. However, the transaction processing time will also increase as the number of federated organizations that need to accept (sign) a transaction increases. To counteract the performance penalty introduced with the increasing number of federated organizations, an appropriate endorsement policy can be used in Hyperledger Fabric. The endorsement policy determines how many organizations are needed to approve the transaction. If the endorsement policy requires fewer organizations to accept a transaction, the ledger performance and, therefore, the performance of dl-mOT will increase.

When reading data from a ledger, no consensus among organizations is required, and therefore, the performance depends only on the number of available ledger nodes; i.e., the more nodes are available, the higher performance of the ledger.

VII. Related work

The interoperability aspects of civilian IoT platforms have been discussed in several earlier papers. A high-level architecture for semantic and syntactic interoperability between cloud-based IoT platforms has been proposed in [15]. A multiauthority access control framework for federated cloud IoT platforms has been presented [16]. This earlier work differs significantly from our contribution and is not directly applicable to military and civil-military scenarios. First, we consider the federation and interoperability of IoT systems in multiple layers. In HADR and military operations, we cannot rely on universal availability of cloud connectivity, but we also target disadvantaged and adversary environments, which my rely on locally deployed IoT enclaves and edge nodes interconnected via private and proprietary links. Moreover, most of the earlier work does not consider explicitly resilience and survavability aspects - the distribution is interpreted rather in context of preserving control, than providing reliability and fault tolerance. In our solution, we focus on ensuring that during the operational phase, i.e. after onboarding, the IoT devices and end users can be authenticated and authorized by any other node belonging to federation, thus mitigating some of the risks related to defects, environmental faults, and adversarial activity.

Several ongoing activities focus on the development of secure and interoperable onboarding and configuration management mechanisms for IoT devices [17], [18]. Our proposed key management protocols are aligned with the frameworks presented in [19] and [20].

The use of blockchains in the context of IoT applications was a topic of several recent surveys, for example, [21]–[23]. However, in the context of our solution, the most relevant work is related to key and access management, as well as the performance evaluation of blockchain-based applications.

The use of blockchain to implement a key management approach in a federated environment with smart vehicles was investigated in [24]. A pairwise key management scheme based on a transitory system-wide secret has been discussed in [25]. The use of randomness obtained from IoT devices for the generation of cryptographic material has been mainly investigated in the context of taking advantage of Physical Unclonable Functions (PUF) [26], [27]. An excellent study of various sources of secure PUFs is provided in [28].

The performance of blockchain, when used to store data obtained from IoT devices, has been studied in [29]. Similarly, the performance of Solidity smart contracts implemented in Ethereum for the management of access to IoT devices is investigated in [30]. This investigation was further extended

in [31], where a comparison of the performance of blockchain-based access management with an alternative CoAP-based solution is presented. The applicability of the Ethereum smart contract for access control in the IoT is also discussed in [32].

We also chose to provide identity-based authenticated key exchange in our device network as a solution for environments where the operation of traditional public key infrastructure (including its various phases of a life cycle, such as certificates dissemination and revocation) is too costly in terms of bandwidth, on-device storage, and computation. Since it is possible to store additional key material in an IoT device, we propose using a modified Okamoto-Tanaka (OT) protocol [2] adapted to the decentralized setting. Our solution is based on [3], which introduced a protocol called mOT. The OT protocol enables two parties to use their identities to establish their common secret keys without sending and verifying public key certificates.

As noted in [3] in terms of computational effort, the OT protocol is more efficient than any RSA-based key agreement protocol and any authenticated Diffie-Hellman protocol over Z_p^* for large prime. OT protocol is incomparably more efficient than any of the ID-based protocols based on elliptic curve pairings. OT is pretty close in terms of computational efficiency to the certificate-based MQV protocol, which runs over elliptic curves, while mOT runs over RSA composites. The cost of MQV on-line is slightly larger than that of OT, but OT requires a setup phase. For moderate security parameters (1024-bit RSA), OT has a computational advantage over MQV, but for larger modulus (above 2000 bits), the advantage is on the elliptic-curve side.

Current key management protocols [15], [16] do not offer such a combination of properties.

We aim to provide a two-message identity-based key exchange, maintaining low computational and communication overhead, achieving the PFS property, and allowing sensors to communicate without interaction with edge nodes. Since the federated IoT environment includes trusted edge nodes, our protocol also allows the implementation of a multiauthority case in which IoT devices from two separate domains (connected to a single edge node) can establish a secret key. In such a case, each edge node plays the role of an independent key generation center (since it belongs to a trusted network or at least a semi-trusted network).

OT framework can also be used as an enabler to implement the Self-Sovereign Identities (SSI). The objective of SSI is to provide subjects with complete control of their own digital identities [33]. There are two standard approaches to SSI, namely, Decentralized Identifiers (DID) and Verifiable Credentials (VC) [34] - both of them rely on public-key cryptography. To control a specific DID, a subject just has to own a private key associated with a public key in the DID document. Although DID focuses on cryptographic identification, VC provides authenticated and privacy-aware attribute disclosure. The mentioned SSI approaches mean that devices need to be able to execute encryption algorithms based on asymmetric keys, which can be challenging in devices with limited processing and energy resources, and cope with the communication overhead of transmitting metadata, such as DID and VC. Furthermore, in the IoT world, low communication overhead plays a vital role, as wireless communication protocols often offer relatively small packet sizes. In particular, low-energy protocols such as Long Range Radio (LoRa) and Bluetooth Low Energy (BLE) have maximum packet sizes of 222 and 244 bytes, respectively, and the mentioned approaches require at least 512 bytes or more, which means that additional mechanisms are required, for example, for message partitioning and lost packet control.

Therefore, although SSI may improve security and privacy protection for IoT devices, new, more efficient cryptographic methods need to be identified to ensure its successful deployment in the IoT environment. In particular, power and communication constraints of small devices in large distributed networks suggest the benefits of using a one-round authenticated key exchange (AKE) protocol in the IoT environment. Our proposed approach revisits the classic Okamoto-Tanaka protocol [3] that realizes single-round key agreement and exchange with perfect forward secrecy, using a single group element per message and maintaining low computational and communication overhead, avoiding the need to distribute large public key certificates.

VIII. Conclusions and Future Work

We have presented dl-mOT, a novel key management solution for federated IoT environments. Our solution integrates an efficient identity-based mOT protocol with a distributed ledger. It relies on a distributed ledger to establish an anchor of trust between federation members.

Our work focused only on a subset of challenges related to the implementation of an operational federated IoT environment. In particular, we have not discussed how the resources available in the federation can be discovered or how to achieve secure time synchronization. We plan to investigate the possible integration of our key management approach with other open frameworks, such as Teserakt E4[1], to implement a solution that can be validated in practice. Moreover, public-key algorithms that we use in our current implementation of the distributed ledger are susceptible to quantum computing attacks. Although it is possible to modify Hyperledger Fabric to use customized public-key cryptography, the integration of quantum-safe algorithms into our solution is left for future work.

Acknowledgment

This work has been partially funded by the NATO Allied Command Transformation Innovation Programme of Work and by the SEMACITI project, sponsored by the Ministry of Defense of Republic of Poland as part of the Kościuszko Programme.

[1]https://teserakt.io/

REFERENCES

[1] F. T. Johnsen, Z. Zieliński, K. Wrona, N. Suri, C. Fuchs, M. Pradhan, J. Furtak, B. Vasilache, V. Pellegrini, M. Dyk, M. Marks, and M. Krzysztoń, "Application of iot in military operations in a smart city," in *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, May 2018, pp. 1–8.

[2] E. Okamoto and K. Tanaka, "Key distribution system based on identification information," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 481–485, May 1989.

[3] R. Gennaro, H. Krawczyk, and T. Rabin, "Okamoto-Tanaka revisited: Fully authenticated Diffie-Hellman with minimal overhead," in *Proc. of Applied Cryptography and Network Security (ACNS)*, vol. 6123 LNCS, 2010, pp. 309–328.

[4] B. Tian, F. Wei, and C. Ma, "mOT+: An efficient and secure identity-based diffie-hellman protocol over RSA group," in *INTRUST 2014: Revised Selected Papers of the 6th International Conference on Trusted Systems*, vol. 9473, 2015, pp. 407–421.

[5] K. Kanciak and K. Wrona, "Towards an Auditable Cryptographic Access Control to High-value Sensitive Data," *Int. J. Electron. Telecommun.*, vol. 66, no. 3, pp. 449–458, 2020.

[6] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," in *Proc. of the Annual Int. Cryptology Conf. (Crypto)*, 1984.

[7] A. Kate and I. Goldberg, "Distributed Private-Key Generators for Identity-based Cryptography," in *Int. Conf. Secur. Cryptogr. Networks*, 2010.

[8] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (Without random oracles)," in *Adv. Cryptol. - CRYPTO*, 2006.

[9] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[10] R. Canetti and H. Krawczyk, "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels," Cryptology ePrint Archive, Report 2001/040, 2001, available at: https://eprint.iacr.org/2001/040.

[11] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally Secure Constant-Rounds Multi-party Computation for Equality, Comparison, Bits and Exponentiation," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 285–304.

[12] C. Ning and Q. Xu, "Constant-rounds, linear multi-party computation for exponentiation and modulo reduction with perfect security," in *Advances in Cryptology – ASIACRYPT 2011*, D. H. Lee and X. Wang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 572–589.

[13] ——, "Multiparty computation for modulo reduction without bit-decomposition and a generalization to bit-decomposition," in *Advances in Cryptology - ASIACRYPT 2010*, M. Abe, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 483–500.

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

[15] I. P. Zarko, S. Mueller, M. Plociennik, T. Rajtar, M. Jacoby, M. Pardi, G. Insolvibile, V. Glykantzis, A. Antonic, M. Kusek, and S. Soursos, "The symbIoTe solution for semantic and syntactic interoperability of cloud-based IoT platforms," in *Global IoT Summit, GIoTS 2019 - Proceedings*. Aarhus, Denmark: IEEE, 2019, pp. 1–6.

[16] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi, "On the Design of a Decentralized and Multiauthority Access Control Scheme in Federated and Cloud-Assisted Cyber-Physical Systems," *IEEE Internet of Things J.*, vol. 5, no. 6, pp. 5190–5204, 2018.

[17] S. Symington, W. Polk, and M. Souppaya, "Trusted Internet of Things (IoT) Device Network-Layer Onboarding and Lifecycle Management," NIST, Working Paper, 2020.

[18] M. Sethi, B. Sarikaya, and D. Garcia-Carrillo, "Secure IoT Bootstrapping: A Survey," IETF, Internet Draft, 2020.

[19] M. Vucinic, G. Selander, J. Mattsson, and D. Garcia, "Requirements for a Lightweight AKE for OSCORE," IETF, Internet Draft, 2020.

[20] F. Palombini, L. Seitz, G. Selander, and M. Gunnarsson, "OSCORE Profile of the Authentication and Authorization for Constrained Environments Framework," IETF, Internet Draft, 2020.

[21] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the internet of things: Research issues and challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2188–2204, 2019.

[22] M. Wu, K. Wang, X. Cai, S. Guo, M. Guo, and C. Rong, "A Comprehensive Survey of Blockchain: From Theory to IoT Applications and beyond," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8114–8154, 2019.

[23] W. Viriyasitavat, L. D. Xu, Z. Bi, and D. Hoonsopon, "Blockchain Technology for Applications in Internet of Things - Mapping from System Design Perspective," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8155–8168, 2019.

[24] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. Ogah, and Z. Sun, "Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, 2017.

[25] F. Gandino, R. Ferrero, B. Montrucchio, and M. Rebaudengo, "Fast Hierarchical Key Management Scheme with Transitory Master Key for Wireless Sensor Networks," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1334–1345, 2016.

[26] B. Chen and F. M. Willems, "Secret Key Generation over Biased Physical Unclonable Functions with Polar Codes," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 435–445, 2019.

[27] P. Gope and B. Sikdar, "Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2019.

[28] NATO STO IST-ET-104, "Physical Unclonable Functions (PUFs) in Military IoT," NATO STO, Tech. Rep., 2019.

[29] M. Alaslani, F. Nawab, and B. Shihada, "Blockchain in IoT Systems: End-to-End Delay Evaluation," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8332–8344, 2019.

[30] O. Novo, "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[31] ——, "Scalable access management in IoT using blockchain: A performance evaluation," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4694–4701, 2019.

[32] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.

[33] G. Fedrecheski, J. Rabaey, L. Costa, P. Ccori, W. Pereira, and M. Zuffo, "Self-Sovereign Identity for IoT environments: A Perspective," in *Global Internet of Things Summit (GIoTS)*, 2020.

[34] M. Sporny, D. Longley, and D. Chadwick, "Verifiable credentials data model 1.0," W3C, Tech. Rep., 2019, https://www.w3.org/TR/2019/REC-vc-data-model-20191119/.