Distributed Genuine Intelligence: From Agent Integrity to Secure Inter-Agent Communications

A. Zahir* and M. Groshev
Universidad Carlos III de Madrid
Madrid, Spain
*Email: azahir@pa.uc3m.es

M. Angoustures and V. Lefebvre
sarl TAGES SOLIDSHIELD
Le Cannet, France

P. González and L. Velasco
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain

***Abstract*—The near real-time control of 6G network services requires decision-making to be placed as close to network devices as possible. A possible solution is to deploy a distributed system with intelligent agents that relieves the classical centralized control plane from such near-real-time control loops. However, distributing decision-making entails new vulnerabilities that attackers can exploit. This paper presents the solution devised by the DESIRE6G project to secure such distributed intelligence and includes remote attestation to verify agents' integrity, as well as secure communications, both supported by immutable transactions based on a blockchain system.**

I. Introduction

Autonomous network operation in near-real-time is essential to efficiently manage the expected high traffic variability and meet the stringent performance requirements of beyond 5G and 6G Network Services (NS). While centralized solutions can potentially decrease operational costs, they might lead to inefficient resource utilization due to slow response times. To minimize these delays, control algorithms, or (intelligent) *multi-agents*, can be executed close to the data plane devices. Implementing such distributed multi-agent systems (MAS) entails that agents interact with other peers, sharing data and knowledge for end-to-end service control. However, they also present several security concerns, as they are more vulnerable to some attacks than centralized approaches. Specifically, attacks can be crafted against such distributed systems that target: (i) the integrity of the agents, and (ii) the inter-agent communication. For the first type of attack, trusting agents is crucial and can be elaborated on first place through authentication, which assures the place of origin and integrity of the deployed agents. We have designed an ad-hoc remote attestation scheme adapted to the distributed nature of the MAS, featuring cumulative security above cumulative intelligence. For the second type of attack, we rely on encryption, where keys are distributed to the agents only after they successfully pass the attestation procedure. For both, attestation and key distribution, Distributed Ledger Technologies (DLT), such as *blockchain*, offer a promising solution by providing a secure, shared, immutable, and decentralized ledger of transactions.

In this paper, we extend our previous work in [1] and present a set of strategies designed to improve the security of MAS, focusing on their integrity and internal communications. The proposed solution has been developed as part of the DESIRE6G (D6G) project under Horizon Europe [2] and integrates the following components: (i) a Machine Learning Function Orchestrator (MLFO) that coordinates the deployment of agents and their configuration, with the help of a Service Management and Orchestration System (SMO) and an Infrastructure Management Layer (IML); (ii) Security-as-a-Service (SECaaS) automatic executable rewriting tool, that creates hardened, authenticable and self-monitored agents; and (iii) DLT, precisely the combination of blockchain and smart contracts.

II. Proposed solution

A. Architecture overview

Fig. 1 shows an overview of the essential DESIRE6G components required for executing the secure distributed intelligence. The architecture comprises multiple D6G sites in different geographical locations, each equipped with local networking, computing, and storage resources. Central to this architecture is the D6G SMO layer, which oversees the orchestration and lifecycle management aspects of the NS, providing guidelines for the agents and enabling them to operate autonomously. Within the SMO, an MLFO determines the deployment locations and connections of agents. A SECaaS tool is used to improve the security of the agents by adding distributed attestation sub-routines for measurement and verification to their software. The Service Orchestrator (SO) is the responsible component in the SMO to coordinate the deployment, where the IML is in charge of the interactions with the local virtual infrastructure.

Finally, a blockchain infrastructure is also part of the D6G solution that sets up two distinct smart contracts. The first contract dynamically manages the measurement and verification functions of agents, enabling mutual remote attestation mechanisms between agents. This allows each agent to check the integrity of another agent in the system, avoiding a centralized, Deny-of-Service-vulnerable verifier. The second contract ensures a secure exchange of keys/secrets for inter-agent communications. It is worth mentioning that the DESIRE6G architecture offers a multi-use case blockchain framework, which, besides securing the MAS, also supports a federation of services [3].

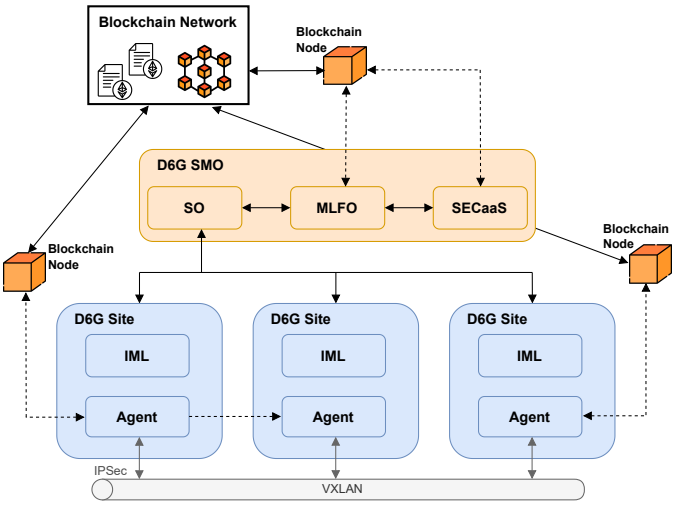

Fig. 1. Proposed DESIRE6G architecture for secure distributed intelligence

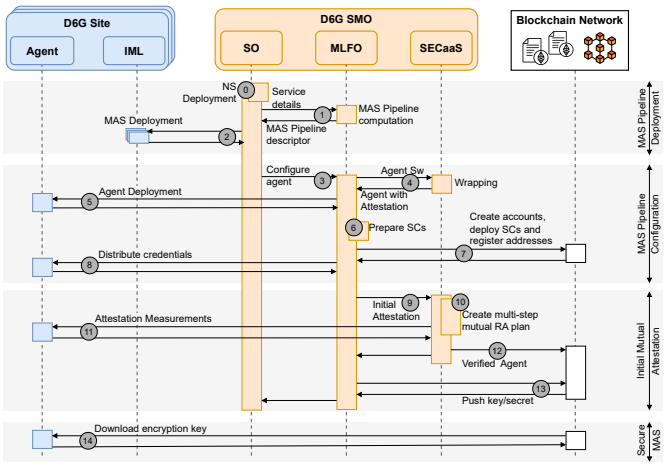

Fig. 2. MAS pipeline deployment with initial mutual attestation workflow

B. High-level description of the MAS deployment workflow

Fig. 2 outlines the proposed workflow for the MAS Pipeline deployment, which includes the MAS agents, their initial mutual attestation process, and their connectivity. This workflow consists of four phases: (i) deployment of the MAS pipeline; (ii) configuration of protected agents through a SECaaS binary hardening tool; (iii) mutual attestation; and (iv) secret/key exchange. The workflow is initiated by the SO during the deployment of the NS (step 0 in Fig. 2).

The SO starts with deploying the MAS pipeline and requests the computation of the topology of a MAS pipeline given the relevant details of the NS, i.e., the location of the VNFs, the performance required for that service, etc. (1). The MLFO computes the optimal MAS pipeline and returns a descriptor that specifies the agents': deployment location, booting image, and connectivity. The SO then interacts with the IML components at different D6G sites to deploy each agent (2). Inter-agent connectivity based on a VXLAN that connects the D6G sites is also deployed in this step (not shown in Fig. 2). Once the agents are running and connectivity is established, the configuration of the agents starts (3). In this phase, the SECaaS system processes the native agent software (e.g., x86) and improves its security properties by including measurement, verification, and communication subroutines. Through this *wrapping* technique, SECaaS generates a reference measurement of the agent and stores it in a secure repository, ensuring its integrity for future verifications (4). The MLFO then deploys the wrapped images of the agents in the sites (5). Next, it creates smart contracts that will be used to record the state of agents during the remote attestation workflow and store the secrets/keys for the MAS pipeline (6). It also creates blockchain accounts for the agents, including a public address and private key, deploys smart contracts to the blockchain network, and registers the agents' addresses (7). This allows the MLFO to control access to the smart contracts and add or revoke permissions in case of MAS pipeline reconfiguration or detection of infected agents.

Once the addresses are registered, the MLFO distributes the blockchain credentials to the agents involved in the MAS pipeline for interaction with the smart contracts (8).

To face group attacks, where a corrupted agent verifies another corrupted agent, an initial *root of trust* (RoT) is established for the first agent verification. To that end, the first attestation is triggered by the SECaaS when the MLFO confirms that the MAS configuration has finished (9). For the initial attestation, SECaaS prepares a remote attestation plan and asks the agents to run their attestation sub-routines for measurement and verification (10). The resulting hash value from the measurements (11) is compared to the one stored in the SECaaS repository and if it matches the agent is verified. A transaction is created in the blockchain for each remote attestation, including the test result, the identifiers of the involved agents (i.e., measured and verifier ID), a timestamp, and the updated RoT (i.e., last verified agent) in the first smart contract (12). Once the NS operation phase starts, the smart contract automatically manages the remote attestation process through an internal function that receives a group attestation scheduling plan (the sequence of agents for verification) and notifies the RoT with the reference measurement of the agent, without compromising its confidentiality to other potentially infected agents. These steps are performed periodically throughout the NS lifecycle. Finally, the MLFO pushes to the second smart contract the key/secret that the agents need to use to encrypt the inter-agent communication through the VXLAN (13-14). See the details in [1].

ACKNOWLEDGMENT

This work has been partially funded by the Smart Networks and Services Joint Undertaking under the European Commission Horizon Europe DESIRE6G project (G.A. 101096466).

REFERENCES

[1] L. Velasco *et al.*, "Securing multi-agent systems for near real-time control of 6G services," *European Conf. on Ntw and Comm (EuCNC)*, 2023.
[2] C. Papagianni, "DESIRE6G: Deep Programmability & Secure Distributed Intelligence for Real-Time E2E 6G Networks," 2023. [Online]. Available: https://doi.org/10.5281/zenodo.10351667
[3] A. Zahir *et al.*, "Performance evaluation of private and public blockchains for multi-cloud service federation," *25th International Conference on Distributed Computing and Networking (ICDCN)*, 2024.