Sushil Jajodia · George Cybenko
V.S. Subrahmanian · Vipin Swarup
Cliff Wang · Michael Wellman *Editors*

Adaptive Autonomous Secure Cyber Systems

Springer

Editors
Sushil Jajodia
Center for Secure Information Systems
George Mason University
Fairfax, VA, USA

George Cybenko
Thayer School of Engineering
Dartmouth College
Hanover, NH, USA

V.S. Subrahmanian
Department of Computer Science
Dartmouth College
Hanover, NH, USA

Vipin Swarup
MS T310
MITRE Corporation
McLean, VA, USA

Cliff Wang
Computing and Information Science Division
Army Research Office
Durham, NC, USA

Michael Wellman
Computer Science & Engineering
University of Michigan
Ann Arbor, MI, USA

ISBN 978-3-030-33431-4 ISBN 978-3-030-33432-1 (eBook)
https://doi.org/10.1007/978-3-030-33432-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Reference Architecture of an Autonomous Agent for Cyber Defense of Complex Military Systems

Paul Theron, Alexander Kott, Martin Drašar, Krzysztof Rzadca, Benoît LeBlanc, Mauno Pihelgas, Luigi Mancini, and Fabio de Gaspari

1 Future Military Systems and the Rationale for Autonomous Intelligent Cyber Defense Agents

Modern defense systems incorporate new technologies like cloud computing, artificial intelligence, lasers, optronics, electronics and submicronic processors, on-board power-generation systems, automation systems, sensors, software defined radios

This chapter reuses portions of an earlier paper: Theron, P., et al, "Towards an Active, Autonomous and Intelligent Cyber Defense of Military Systems: the NATO AICA Reference Architecture", Proceedings of the International Conference on Military Communications and Information Systems Warsaw, Poland, 22nd - 23rd May 2018; © 2018 IEEE.

P. Theron
Aerospace Cyber Resilience Chair, Paris, France
e-mail: paul.theron@thalesgroup.com

A. Kott (✉)
U.S. Army Research Laboratory, Adelphi, MD, USA
e-mail: alexander.kott1.civ@mail.mil

M. Drašar
Masaryk University, Brno, Czech Republic
e-mail: drasar@ics.muni.cz

K. Rzadca
University of Warsaw, Warsaw, Poland
e-mail: krzadca@mimuw.edu.pl

B. LeBlanc
Ecole Nationale Supérieure de Cognitique, Bordeaux, France
e-mail: benoit.leblanc@ensc.fr

S. Jajodia et al. (eds.), *Adaptive Autonomous Secure Cyber Systems*,
https://doi.org/10.1007/978-3-030-33432-1_1

and networks, etc. They are more and more relying on software, and will also embed new hardware technologies, including high performance computers and quantum technologies, nanoparticles, metamaterials, self-reconfigurable hardware, etc.

While defense infrastructures and systems engaged on the battle ground may not fail, the multitude of high-tech features and interconnections that they embed make cyber-attacks a good way to affect their functionality and the missions in which they are involved.

Today, five broad classes of systems coexist in Land, Sea and Air operations:

- Office and information management systems: these include web services, emailing systems, and information management applications ranging from human resource management to logistics through maintenance and project management;
- C4ISR systems for the command of war operations: they include associated Battlefield Management Systems that extend the C4ISR down to single vehicles and platoons;
- Communication systems: they include SATCOM, L16, line of sight networks, software defined radios, and the Internet of Battle Things (IoBT) can be seen as a major operational innovation and extension of communication capabilities;
- Platform and life automation systems: they are similar to industrial systems and provide sea vessels or armored vehicles, for instance, with capabilities such as air conditioning, refrigeration, lifts, video surveillance, etc.;
- Weapon systems: these include sensors and effectors of all kinds, operating in all kinds of situations and contested battle grounds.

On the battlefield, these platforms and technologies will operate together in complex large scale networks of massively interconnected systems.

Autonomy can be defined as the capacity of systems to decide by themselves on their course of action in uncertain and challenging environments without the help of human operators. It should not be confused with automation, the aptitude of systems to perform set tasks according to set rules in environments where uncertainty is low and characterized [11].

Despite the fact that “Full autonomy might not necessarily be the objective” and the existence of a variety of definitions [5], the number of autonomous military systems will grow [10]. They will be able to mitigate operational challenges such as needs for rapid decision-making, high heterogeneity and/or volumes of data, intermittent communications, the high complexity of coordinated actions, or the danger of missions, while they will require persistence and endurance [11, p. 12].

M. Pihelgas
NATO Cooperative Cyber Defence Centre of Excellence, Tallinn, Estonia
e-mail; mauno.pihelgas@ccdcoe.org

L. Mancini · F. de Gaspari
Sapienza University, Rome, Italy
e-mail: mancini@di.uniroma1.it; degaspari@di.uniroma1.it

Examples of autonomous capabilities and systems [11] include:

- Unmanned Air Systems, Unmanned Surface Vehicles and Unmanned Underwater Vehicles, will be able to carry out reconnaissance or attack missions stealthily, some of them with a large endurance. For instance, the Haiyan UUV [24] is a mini-submarine built on a civilian platform that China's People's Liberation Army Navy (PLAN) sponsors to create an autonomous UUV capable of carrying out dangerous missions like minesweeping and submarine detection operations without any human intervention. Weighing only 70 kg, energy efficient and fitted with advanced computing capacities, it has an endurance of up to 30 days.
- Today's Intelligence, Surveillance & Recognition (ISR) missions request more and more high-definition (HD) images and videos being captured and transmitted back to ground stations for analysis. As HD means large volumes of raw data (possibly encrypted, which adds to volumes), communication means cannot provide the ad hoc transmission throughput (and continuity in contested environments). Autonomous sensors equipped with artificial intelligence will be capable of generating on the ground aggregated, high-level information that can be more easily transmitted to command posts as they require much less bandwidth than raw data, also lowering the human workload needed to process high volumes of complex multi-source raw data.
- Autonomous Unmanned Ground Vehicles can be employed in dealing with chemical, biological, radiological and nuclear (CBRN) threats as well as with Improvised Explosive Devices (IED), as was the case in Iraq and Afghanistan conflicts.
- The US MK-18 Mod 2 program has demonstrated significant progress in utilizing Remote Environmental Monitoring UnitS (REMUS) Unmanned Underwater Vehicles for mine countermeasure missions, thus allowing pulling military personnel away from dangerous mine fields and reducing tactical reaction times.
- Unmanned Aircrafts (UA) could be used in anti-access and area denial (A2/AD) missions to perform functions that today require the intervention of personnel such as aerial refueling, airborne early warning, ISR, anti-ship warfare, command, offensive strike facilitation (electronic warfare, communications jamming, decoys) and actions supporting defense by creating confusion, deception or attrition through decoys, sensors and emitters, target emulators. Similar functions could be used in underwater combat.
- Agile ground forces could get local tactical support from Unmanned Aircraft Systems (UAS) embarking sensors, ISR capacities, communication means, electronic warfare functions and weapon systems. These UAS would reach greater levels of efficiency and could better deal with large numbers of ground, air and possibly sea sensors and actuators if they could themselves work in swarms or cohorts and collectively adapt their action dynamically on the basis of mission and environment-related data collected in real time.
- Logistics could be another area of utilization of autonomous land, sea and air vehicles and functions. Autonomous capabilities could be used in contested changeable environments either in support and defense of friendly logistic deployment and operation, or to disturb or strike enemy logistics.

Another two fundamental issues need to be taken into account.

The first of these issues is the fact that the level of interconnectedness, and therefore of interdependence and cross-vulnerability, of military systems will increase to unseen heights [42].

The Internet of Things is increasing rapidly in both numbers and types of smart objects, and this is a durable trend with regards to Defense [11] despite the massive scale of their deployment, their meager configurability and the new (cyber) risks they create. In effect, with the shift to the IPv6 addressing standard, the number of devices that can be networked is up to 340 undecillion unique devices (340 with 36 zeroes after it) and this immense network of interconnected devices could become a global platform for massively proliferated, distributed cyber-attacks [11].

This multitude of devices will work together in clusters, likely hard to map out, likely subject to unstable changing configurations of their dependencies. These changes will occur beyond our control because of the degrees of autonomy conferred to objects in shifting operative conditions.

Massively interconnected military systems will become more and more difficult to engineer, test, maintain, operate, protect and monitor [42], which leads the authors to recommend "reducing the number of interconnections by reversing the default culture of connecting systems whenever possible" to improve cybersecurity. This recommendation, however intelligent it seems, is very likely never to be listened to . . .

Thus, cyber defending such complex systems will become arduous. For instance, they will not anymore allow for the sort of cybersecurity monitoring we currently deploy across IT and OT systems as they will prevent the implementation of classic, centralized, and even big data/machine learning-based security operations centers (SOCs).

They will also overwhelm human SOC operators' cognitive capacities as it will become impossible for the latter to get instantly a clear and adequate picture of the systems they defend, of their condition, of the adverse events taking place and of the remedies to apply and of their possible impacts.

To defend them against cyber-attacks, only locally implemented distributed and resilient swarms of cyber defense agents adapting to these frequent reconfigurations and emerging circumstances will be able to monitor and defend this vast fuzzy network, learning superior abilities from cumulated experience.

In this particular context, different from the previously exposed context of the cyber defense of a few well-identified and carefully managed autonomous mission systems, cyber defense agents will evolve themselves into more and more unknown, less and less controllable and maintainable states.

Given this last parameter, they may either show decreasing levels of efficiency or generate uncontrollable adverse effects.

The second issue stems from the fundamental military need to proceed successfully with defense missions while operational personnel of Air, Land and Sea forces are not primarily specialists of cybersecurity and cyber defense. This is not to mention that on the battlefield there will always be a scarcity of cyber competencies [22].

Cyber-attacks may cause human operators sometimes to be fooled, for instance when radar or GPS data are spoofed, or stressed, for instance when anomalies multiply while their cause appears unclear and their consequences detrimental. Studies in a variety of domains such as air, sea and ground transportation have drawn attention to this phenomenon. Attacks may trigger human errors of varying consequences. For instance, NAP [29] points out that "Inaccurate information sent to system operators, either to disguise unauthorized changes, or to cause the operators to initiate inappropriate actions, [] could have various negative effects".

The burden of cyber defending systems must therefore be relieved from unqualified operators' shoulders, while the lack of specialists of cybersecurity on the ground prohibits calling upon rapid response teams in case of trouble.

In this context, handling cyber-attacks occurring in the course of operations requires an embedded, undisturbing, seamless autonomous intelligent cyber defense technology [45]. Autonomous intelligent cyber defense agents should resolve (at -least most of) cyber-attacks without technology users being aware of issues at hand.

Only when they would reach their limits, i.e. when being unable to understand situations, to reconciliate disparate pieces of information, or to elaborate cyber defense counter-measures, such multiple agents should collaborate with human operators. NAP [28] provides inspiring examples of machine-human collaboration in a variety of contexts. Such a need for collaboration might also exist in the context of massively interconnected systems of systems evoked earlier.

2 NATO's AICA Reference Architecture: A Concept for Addressing the Need for an Autonomous Intelligent Cyber Defense of Military Systems

Inspired by the above rationale, NATO's IST-152 Research and Technology Group (RTG) is an activity that was initiated by the NATO Science and Technology Organization and was kicked-off in September 2016. The group has developed is developing a comprehensive, use case focused technical analysis methodology in order to produce a first-ever reference architecture and technical roadmap for active autonomous intelligent cyber defense agents. In addition, the RTG worked to identify and evaluate selected elements that may be eligible contributors to such capabilities and that begin to appear in academic and industrial research.

Scientists and engineers from several NATO Nations have brought unique expertise to this project. Only by combining multiple areas of distinct knowledge along with a realistic and comprehensive approach can such a complex software agent be provided.

The output of the RTG may become a tangible starting point for acquisition activities by NATO Nations. If based on a common reference architecture, software agents developed or purchased by different Nations will be far more likely to be interoperable.

Related research includes Mayhem (from DARPA Cyber Challenge, but also Xandra, etc.), agents from the Pechoucek's group, Professor Mancini's work on the AHEAD architecture [9] and the Aerospace Cyber Resilience research chair's research program [45], Anti-Virus tools (Kaspersky, Bitdefender, Avast, Norton, etc.), HBSS, OSSEC, Various host-based IDS/IPS systems, Application Performance Monitoring Agents, Anti-DDOS systems and Hypervisors. Also, a number of related research directions include topics such as deep learning (especially if it can be computationally inexpensive), Botnet technology (seen as a network of agents), network defense games, flip-it games, the Blockchain, and fragmentation and replication. The introduction of Artificial Intelligence into military systems, such as C4ISR, has been studied, for instance by Rasch et al. [35, 36]. Multi Agent Systems form an important part of AI.

Since the emergence of the concept of Multi Agent Systems, e.g., [46], MAS have been deployed in a number of contexts such as power engineering [25] and their decentralized automated surveillance [7], industrial systems [33], networked and intelligent embedded systems [16], collective robotics [19], wireless communication [21], traffic simulation and logistics planning [8], home automation [20], etc.

However, if the use of intelligent agents for the cyber defense of network-centric environments has already long been envisaged [43], effective research in this area is still new.

In the context of the cyber defense of friendly systems, an "agent" has been defined [45] as a piece of software or hardware, a processing unit capable of deciding on its own about its course of action in uncertain, possibly adverse, environments:

- With an individual mission and the corresponding competencies, i.e. in analyzing the milieu in which the agent is inserted, detecting attacks, planning the required countermeasures, or steering and adapting tactically the execution of the latter, or providing support to other agents like for instance inter-agent communication;
- With proactivity, i.e. the capacity to engage into actions and campaigns without the need to be triggered by another program or by a human operator;
- With autonomy, i.e. a decision making capacity of its own, the capacity to function or to monitor, control and repair itself on its own, without the need to be controlled by another program or by a human operator, and the capacity to evaluate the quality of its own work and to adjust its algorithms in case of deviance from its norm or when its rewards (satisfaction of its goals) get poor;
- Driven by goals, decision making and other rules, knowledge and functions fit for its purpose and operating circumstances;
- Learning from experience to increase the accuracy of its decisions and the power of its reactions;
- With memories (input, process, output, storage);
- With perception, sensing and action, and actuating interfaces;
- Built around the adequate architecture and appropriate technologies;
- Positioned around or within a friendly system to defend, or patrolling across a network;

- Sociable, i.e. with the capacity to establish contact and to collaborate with other agents, or to enter into a cyber cognitive cooperation when the agent requires human help or to cooperate with a central Cyber C2;
- Trustworthy, i.e. that will not deceive other agents nor human operators;
- Reliable; i.e. that do what they are meant to do, during the time specified and under the conditions and circumstances of their concept of operation;
- Resilient, i.e. both robust to threats (including cyber-threats aimed at disabling or destroying the agent itself; the agent being able to repel or withstand everyday adverse events and to avoid degrading), and resistant to incidents and attacks that may hit and affect the agent when its robustness is insufficient (i.e. the agent is capable of recovering from such incidents or attacks);
- Safe, i.e., conceived to avoid harming the friendly systems the agent defends, for instance by calling upon a human supervisor or central cyber C2 to avoid making wrong decisions or to adjust their operating mode to challenging circumstances, or by relocating when the agent is the target of an attack and if relocation is feasible and allows protecting it, or by activating a fail-safe mode, or by way of self-destruction when no other possibility is available.

In the same context (ibid), a multi agent system is a set of agents:

- Distributed across the parts of the friendly system to defend;
- Organized in a swarm (horizontal coordination) or cohort (vertical coordination);
- In which agents may have homogeneous or heterogeneous roles and features;
- Interoperable and interacting asynchronously in various ways such as indifference, cooperation, competition;
- Pursuing a collective non-trivial cyber defense mission, i.e. allowing to piece together local elements of situation awareness or propositions of decision, or to split a counter-attack plan into local actions to be driven by individual agents;
- Capable of self-organization, i.e. as required by changes in circumstances, whether external (the attack's progress or changes in the friendly system's health or configuration) or internal (changes in the agents' health or status);
- That may display emergent behaviors [26], i.e. performances that are not explicitly expressed in individual agents' goals, missions and rules; in the context of cyber defense, "emergence" is likely to be an interesting feature as, consisting in the "way to obtain dynamic results, from cooperation, that cannot easily be predicted in a deterministic way" [26]; it can be disturbing to enemy software in future malware-goodware "tactical" combats within defense and other complex systems;
- Extensible or not, i.e. open or closed to admitting new agents in the swarm or cohort;
- Safe, trustworthy, reliable and resilient as a whole, which is a necessity in the context of cyber defense whereas in other, less challenging contexts may be unnecessary. Resilience, here, may require maintaining a system of virtual roles as described in a human context by Weick [47].

AICA will not be simple agents. Their missions, competencies, functions and technology will be a challenging construction in many ways.

Among many such challenges, we can mention [45] working in resource-constrained environments, the design of agents' architecture and the attribution of roles and possible specialization to each of them, agents' decision making process [3], the capacity to generate and execute autonomously plans of counter-measures in case of an attack, agents' autonomy, including versus trustworthiness, MAICA's safety to defense systems, cyber cognitive cooperation [23], agents' resilience in the face of attacks directed at them by enemy software, agents' learning capacities and the development of their functional autonomy, the specification and emergence of collective rules for the detection and resolution of cyber-attacks, AICA agents' deployment concepts and rationale, their integration into host hardware as [33] showed in industrial system contexts, etc.

To start the research with an initial assumption about agents' architecture, the IST-152-RTG designed the AICA Reference Architecture [22] on the basis of classical perspective reflected in [37].

At the present moment, it is assumed to include the following functional components (Fig. 1).

The AICA Reference Architecture delivers five main high-level functions (Fig. 2):

- Sensing and world state identification.
- Planning and action selection.
- Collaboration and negotiation.
- Action execution.
- Learning and knowledge improvement.

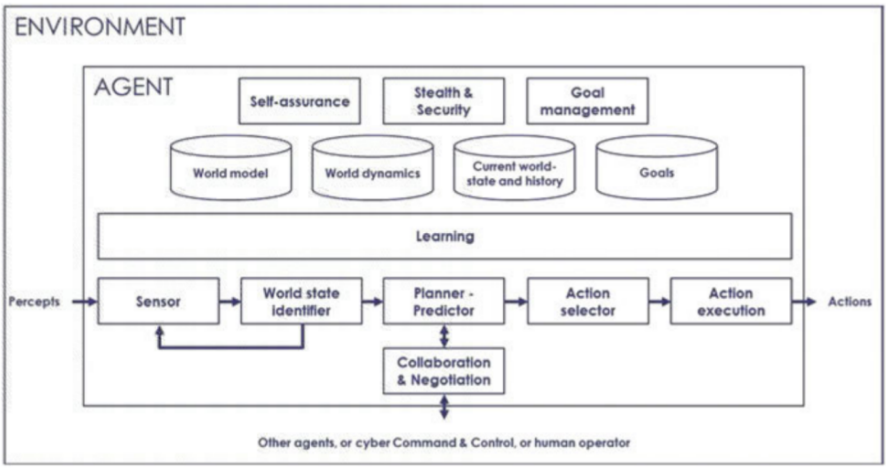

Fig. 1 Assumed functional architecture of the AICA

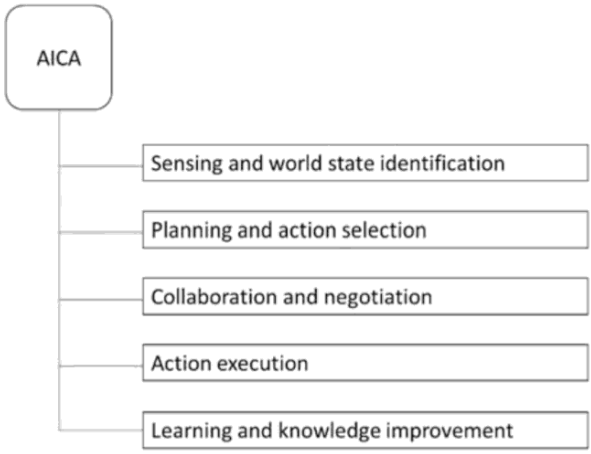

Fig. 2 The AICA's main five high-level functions

2.1 Sensing and World State Identification

Definition: Sensing and World state identification is the AICA's high-level function that allows a cyber-defense agent to acquire data from the environment and systems in which it operates as well as from itself in order to reach an understanding of the current state of the world and, should it detect risks in it, to trigger the Planning and Action selection high-level function. This high-level function relies upon the "World model", "Current world state and history", "Sensors" and "World State Identifier" components of the assumed functional architecture.

The Sensing and World state identification high-level function includes two functions: (1) Sensing; (2) Word state identification.

2.1.1 Sensing

Description: Sensing operates from two types of data sources: (1) External (system and device-related) current world state descriptors; (2) Internal (agent-related) current state descriptors.

Current world state descriptors, both external and internal, are captured on the fly by the agent's Sensing function. They may be double-checked, formatted or normalized for later use by the World state identification function (to create processed current world state descriptors).

2.1.2 World State Identification

Description: The World state identification function operates from two sources of data: (1) Processed current world state descriptors; (2) Learnt world state patterns.

Learnt world state patterns are stored in the agent's world knowledge repository. Processed current world state descriptors and Learnt world state patterns are compared to identify problematic current world state patterns (i.e. presenting an anomaly or a risk). When identifying a problematic current world state pattern, the World state identification function triggers the Planning and Action selection high-level function.

2.2 Planning and Action Selection

Definition: Planning and action selection is the AICA's high-level function that allows a cyber-defense agent to elaborate one to several action proposals and to propose them to the Action selector function that decides the action or set of actions to execute in order to resolve the problematic world state pattern previously identified by the World state identifier function. This high-level function relies upon the "World dynamics" that should include knowledge about "Actions and effects", "Goals", "Planner - Predictor" and "Action selector" components of the assumed functional architecture.

The Planning and action selector high-level function includes two functions: (1) Planning; (2) Action selector.

2.2.1 Planning

Description: The Planning function operates on the basis of two data sources: (1) Problematic current world state pattern; (2) Repertoire of actions (Response repertoire).

The Problematic current world state pattern and Repertoire of actions (Response repertoire) are concurrently explored in order to determine the action or set of actions (Proposed response plan) that can resolve the submitted problematic current world state pattern. The action or set of actions so determined are presented to the Action selector. It may be possible that the Planning function requires some form of cooperation with human operators (cyber cognitive cooperation, C3).

It may alternatively require cooperation with other agents or with a central cyber C2 (command and control) in order to come up with an optimal set of actions forming a global response strategy. Such cooperation could be either to request from other agents or from the cyber C2 complementary action proposals, or to delegate to the cyber C2 the responsibility of coordinating a global set of actions forming the wider response strategy.

It may be possible that the Planning function requires some form of cooperation with human operators (cyber cognitive cooperation, C3). It may alternatively require cooperation with other agents or with a central cyber C2 (command and control) in order to come up with an optimal set of actions forming a global response strategy. Such cooperation could be either to request from other agents or from the cyber C2

complementary action proposals, or to delegate to the cyber C2 the responsibility of coordinating a global set of actions forming the wider response strategy.

These aspects have been the object of an initial study in [3] where options such as offline machine learning, pattern recognition, online machine learning, escalation to a human operator, game theoretic option search, and failsafe have been envisaged, and in [23] for cyber cognitive cooperation processes.

2.2.2 Action Selector

Description: The Action selector function operates on the basis of three data sources: (1) Proposed response plans; (2) Agent's goals; (3) Execution constraints and requirements, e.g., the environment's technical configuration, etc.

The proposed response plan is analyzed by the Action selector function in the light of the agent's current goals and of the execution constraints and requirements that may either be part of the world state descriptors gained through the Sensing and World state identifier high-level function or be stored in the agent's data repository and originated in the Learning and Knowledge improvement high-level function. The proposed response plan is then trimmed from whatever element does not fit the situation at hand, and augmented of prerequisite, preparatory or precautionary or post-execution recommended complementary actions. The Action selector thus produces an Executable Response Plan, and then submitted to the Action execution high-level function.

Like with the Planning function, it is possible that the Action selector function requires to liaise with human operators, other agents or a central cyber C2 (command and control) in order to come up with an optimal Executable Response Plan forming part of and being in line with a global response strategy. Such cooperation could be to exchange and consolidate information in order to come to a collective agreement on the assignment of the various parts of the global Executable Response Plan and the execution responsibilities to specific agents. It could alternatively be to delegate to the cyber C2 the responsibility of elaborating a consolidated Executable Response Plan and then to assign to specific agents the responsibility of executing part(s) of this overall plan within their dedicated perimeter. This aspect is not yet studied in the present release of the AICA Reference Architecture.

2.3 Collaboration and Negotiation

Definition: Collaboration and negotiation is the AICA's high-level function that allows a cyber-defense agent (1) to exchange information (elaborated data) with other agents or with a central cyber C2, for instance when one of the agent's functions is not capable on its own to reach satisfactory conclusions or usable results, and (2) to negotiate with its partners the elaboration of a consolidated

conclusion or result. This high-level function relies upon the "Collaboration & Negotiation" component of the assumed functional architecture.

The Collaboration and negotiation high-level function includes, at the present stage, one function: Collaboration and negotiation.

Description: The Collaboration and negotiation function operates on the basis of three data sources: (1) Internal, outgoing data sets (i.e. sent to other agents or to a central C2); (2) External, incoming data sets (i.e. received from other gents or from a central cyber C2); (3) Agents' own knowledge (i.e. consolidated through the Learning and knowledge improvement high-level function).

When an agent's Planning and action selector function or other function needs it, the agent's Collaboration and negotiation function is activated. Ad hoc data are sent to (selected) agents or to a central C2. The receiver(s) may be able, or not, to elaborate further on the basis of the data received through their own Collaboration and negotiation function. At this stage, when each agent (including possibly a central cyber C2) has elaborated further conclusions, it should share them with other (selected) agents, including (or possibly not) the one that placed the original request for collaboration. Once this (these multiple) response(s) received, the network of involved agents would start negotiating a consistent, satisfactory set of conclusions. Once an agreement reached, the concerned agent(s) could spark the next function within their own decision making process.

When the agent's own security is threatened the agent's Collaboration and negotiation function should help warning other agents (or a central cyber C2) of this state.

Besides, the agent's Collaboration and negotiation function may be used to receive warnings from other agents that may trigger the agent's higher state of alarm.

Finally, the agent's Collaboration and negotiation function should help agents discover other agents and establish links with them.

2.4 Action Execution

Definition: The Action execution is the AICA's high-level function that allows a cyber-defense agent to effect the Action selector function's decision about an Executable Response Plan (or the part of a global Executable Response Plan assigned to the agent), to monitor its execution and its effects, and to provide the agents with the means to adjust the execution of the plan (or possibly to dynamically adjust the plan) when and as needed. This high-level function relies upon the "Goals" and "Action execution" components of the assumed functional architecture.

The Action execution high-level function includes four functions:

- Action effector;
- Execution monitoring;
- Effects monitoring;
- Execution adjustment.

2.4.1 Action Effector

Description: The Action effector function operates on the basis of two data sources:

- Executable Response Plan;
- Environment's Technical Configuration.

Taking into account the Environment's Technical Configuration, the Action effector function executes each planned action in the scheduled order.

2.4.2 Execution Monitoring

Description: The Execution monitoring operates on the basis of two data sources:

- Executable Response Plan;
- Plan execution feedback.

The Execution monitoring function should be able to monitor (possibly through the Sensing function) each action's execution status (for instance: done, not done, and wrongly done). Any status apart from "done" should trigger the Execution adjustment function.

2.4.3 Effects Monitoring

Description: The Effects monitoring function operates on the basis of two data sources: (1) Executable Response Plan; (2) Environment's change feedback.

It should be able to capture (possibly through the Sensing function) any modification occurring in the plan execution's environment. The associated dataset should be analyzed or explored. The result of such data exploration might provide a positive (satisfactory) or negative (unsatisfactory) environment change status. Should this status be negative, this should trigger the Execution adjustment function.

2.4.4 Execution Adjustment

Description: The Execution adjustment function operates on the basis of three data sources: (1) Executable Response Plan; (2) Plan execution feedback and status; (3) Environment's change feedback and status.

The Execution adjustment function should explore the correspondence between the three data sets to find alarming associations between the implementation of the Executable Response Plan and its effects. Should warning signs be identified, the Execution adjustment function should either adapt the actions' implementation to circumstances or modify the plan.

2.5 *Learning and Knowledge Improvement*

Definition: Learning and knowledge improvement is the AICA's high-level function that allows a cyber-defense agent to use the agent's experience to improve progressively its efficiency with regards to all other functions. This high-level function relies upon the Learning and Goals modification components of the assumed functional architecture.

The Learning and knowledge improvement high-level function includes two functions: (1) Learning; (2) Knowledge improvement.

2.5.1 Learning

Description: The Learning function operates on the basis of two data sources: (1) Feedback data from the agent's functioning; (2) Feedback data from the agent's actions.

The Learning function collects both data sets and analyzes the reward function of the agent (distance between goals and achievements) and their impact on the agent's knowledge database. Results feed the Knowledge improvement function.

2.5.2 Knowledge Improvement

Description: The Knowledge improvement function operates on the basis of two data sources: (1) Results (propositions) from the Learning function; (2) Current elements of the agent's knowledge.

The Knowledge improvement function merges Results (propositions) from the Learning function and the Current elements of the agent's knowledge.

3 Use Cases

The use-case of military UAVs that operate in teams illustrates a possible deployment of the AICA Reference Architecture. It is based on the AgentFly project developed within the Agent Technology Center [44].

The AgentFly project facilitates the simulation of multi agent Unmanned Aerial Vehicles (UAV). Its features include flight path planning, decentralized collision avoidance and models of UAVs, physical capabilities and environmental conditions [41]. In addition to simulation, AgentFly was implemented on a real fixed-wing Procerus UAV [32].

The basis of this use-case is the set of missions selected for the AgentFly project. It is here extended to include an adversarial cyber-attack activity against the AgentFly UAV to disrupt its mission. The use case is that a swarm of AgentFly

UAVs perform a routine tactical aerial surveillance mission in an urban area. Collaboration between AgentFly UAVs aims at collision avoidance, trajectory planning, automatic distributed load-balancing and mission assurance.

The AgentFly UAVs use case is built around the following assumptions:

- AgentFly UAVs self-assess and share information with neighboring UAVs.
- When setting up a communication channel, AgentFly UAVs have to determine whether they trust their correspondent.
- Network-wide collaboration and negotiation is affected by timing, range, and reachability issues.
- The AgentFly UAV lacks modern cyber defense capabilities and is thus vulnerable to potential cyberattacks.
- Due to environmental conditions, AgentFly UAVs might be offline for some time and later re-join the swarm when connectivity allows.
- A single AICA agent is implemented within each AgentFly UAV.
- The AICA connects with the modules of the UAV and can supervise the activity and signals in and between various UAV modules (e.g., sensors, navigation, and actuators).
- The AICA can function in isolation from other AgentFly UAVs' AICA agents present in the AgentFly UAV swarm.

Attackers have acquired a technology similar to that used in AgentFly UAVs' COMMS module. They have discovered a zero-day vulnerability that can be exploited remotely over the radio link from the ground and they plan to use the vulnerability in order to gain control over the swarm of UAVs and cut them off from the theatre's Command & Control (C2) system. The UAVs are using the COMMS module to collaborate among themselves and report to the C2 when needed.

The vulnerability lies in the functionality responsible for dynamically registering new UAV agents in the swarm upon due request. The COMMS module is interconnected with other intrinsic modules of the AgentFly UAV via a central control unit.

The adversary has set up a ground station in the area of the surveillance mission. When AgentFly UAVs enter the area, the cyberattack is launched.

The AICA detects a connection to the COMMS module and allows the incoming connection for the dynamic registration of a new UAV agent into the swarm. Due to the nature of zero-day attacks, an Intrusion Detection System (IDS) would not have any corresponding signatures to detect a compromised payload.

The AICA's Sensor monitors the entire set of modules of the AgentFly UAV.

The AICA's World-state identifier module flags the connection from a newly connected UAV agent as anomalous since it does not follow the baseline pattern that has been established out of previous connections with legitimate UAVs. It also detects a change in the UAV's system configuration and deems it anomalous because no new configurations have been received from the C2. The AICA launches, through its Sensor module, a system integrity check. A compromise within the UAV's COMMS module is detected.

The AICA decides (Planner-Selector and Action selection modules) to isolate (Action execution module) the COMMS module from other UAV modules in order

to prevent further propagation. Alerting the C2 is not possible because of the compromised COMMS module.

In order to reduce the attack surface, the AICA requests (Action execution module) that the UAV's central control unit resets the COMMS module, raises the security level and disables auxiliary functions (among others, the dynamic inclusion of new UAVs into the swarm).

The AICA performs another integrity check to verify that no other compromise exists. It keeps its Sensor and World-state identifier modules on a high-level of vigilance in relation to integrity monitoring. The AICA adds the signature of the payload that caused the anomaly into its knowledge base. And it sends out an alert along with malware signature updates to other agents as well as to the C2.

This basic, single AICA agent, use case should be expanded to Multi AICA agents deployed across the AgentFly UAV's architecture and modules. Future research will benchmark Multi AICA agents versus Single AICA agent deployments in order to assess the superiority and context of Multi AICA agent solutions.

4 Discussion and Future Research Directions

The AICA Reference Architecture (AICARA) [22] was elaborated on the basis of [37, 38].

Since the end of 70's and the early works on Artificial Intelligence (AI), the concept of agent was used by different authors to represent different ideas. This polymorphic concept was synthesized by authors such as [30, 48]. Since 1995, Russell and Norvig [38] proposed an architecture and functional decomposition of agents widely regarded as reference work in the ever-growing field of AI.

Their agent architecture can be seen as an extension of the developments in object-oriented methods for software development that culminated in the Unified Modeling Language [4] and design patterns [17]. Both concepts form the basis of modern software development.

The concept of cooperating cognitive agents [38] perfectly matches requirements for AICA agents.

First, AICA agents need to prove trustworthy, and therefore the AICA Reference Architecture is conceived as a white-box. The agent's architecture involves a set of clearly defined modules and specifies the links connecting information perception to action actuation or else the agent to external agents or a central cyber defense C2.

Second, the AICA agents must go beyond merely reactive agents because in situations of autonomy they will need to make decisions by themselves. Reactive agents are today widely used in cybersecurity and are based on rule sets in the form of "if X suspicious, then trigger Y".

Third, Russell and Norvig [38] has attributes highly required by AICA agents: autonomous decision making, learning and cooperation. This is important because these agents may operate for prolonged periods of time if deployed in autonomous weapon systems. The latter may face multiple and unknown cyber-attacks and AICA

agents, by learning and cooperating with one another, will sustain their capacity to equip the weapon system with an autonomous intelligent cyber defense.

Applied to the field of the autonomous cyber defense of military systems [38], well-known concepts must be reassessed, prototypes must be built and tested, and the superiority of the concept must now be benchmarked.

Developing the concepts described here also presents many other challenges that require research in the coming years.

Agents' integrity, agent communications' security, the inclusion of cyber defense techniques such as deception, or else identifying and selecting the right actions, are only a few of them.

4.1 Agents' Integrity

A compromise of agents can potentially threaten the entire military platform they are supposed to defend. It is paramount to harden the agents' architecture in order to minimize the chance of such compromise. Methods that assess the integrity of the agent during runtime are required.

Virtualization techniques have been successfully employed to improve systems' resiliency [2, 18]. For instance, systems such as [18] allow providing security guarantees to applications running on untrusted operating systems. It is possible to build upon such techniques in order to harden AICA agents and to maintain their functionality even under attack or in case of partial compromise. Furthermore, periodical assessment of agents' integrity can be performed through attestation techniques [15], based on a trusted hardware core (Trusted Platform Module, TPM). Such techniques allow ensuring that the software of the agent has not been altered at any time, even during the operations of the platform, and can easily scale up to millions of devices [1]. Finally, while the topic of protecting machine learning systems from adversarial examples is still relatively new, techniques such as distillation [31] could be leveraged to increase robustness.

4.2 Agent Communications' Security

Sensors are the fundamental building blocks providing the agents with a consistent world view. As such, they are a part of the AICA architecture most exposed to adversarial tampering. The AICA architecture needs to provide secure communications to ensure that the agent's world view is not corrupted.

To this end, cryptographic protocols such as random key pre-distribution [12, 13], can be employed to provide secure agent-sensor communication even when one or more sensor channels are compromised.

4.3 The Inclusion of Cyber Defense Techniques Such as Deception

Deception is a key component of active defense systems and, consequently, could be part of the AICA architecture. Active defense deception tools can be used to thwart an ongoing attack. To provide this functionality, the AICA architecture can employ deception techniques such as honeyfiles [6, 49], mock sensors [14] and fake services [34]. Moreover, implementing dynamic tools deployment and reconfiguration is required for actuating functions. To this end container technologies can be employed, such as in [9] to provide isolation and configuration flexibility.

4.4 Identifying and Selecting the Right Actions

Identifying the appropriate actions to take in response to external stimuli is one of the key requirements for the AICA architecture. The AICA agent should include autonomous decision making that can adapt to the current world state. Machine learning-based techniques can be employed [39] to this end, to devise complex plans of action [40] to mitigate an attack, and to learn from previous experiences. However, Blakely and Theron [3] have shown that a variety of techniques may be called upon by AICA agents to elaborate their decisions.

5 In Conclusion

AICA agents are required by foreseeable evolutions of military systems, and it is likely that civil systems, such as the wide-scale deployment of the Internet of Things, will generate similar demands.

The AICA Reference Architecture (AICARA) [22] is a seminal proposition to answer the needs and challenges of the situation.

NATO's IST-152 Research and Technology Group (RTG) has initiated this piece of work and in a recent meeting held in Warsaw, Poland, has evaluated that future research is likely to span over the next decade before efficient solutions be operated.

The AICARA opens discussions among the scientific community, from computer science to cognitive science, Law and moral philosophy.

Autonomous intelligent cyber defense agents may change the face of the fight against malware. This is our assumption.

References

1. Ambrosin, M. et al., 2016. *SANA: Secure and Scalable Aggregate Network Attestation.* New York, NY, USA, ACM, pp. 731–742.
2. Baumann, A., Peinado, M. & Hunt, G., 2015. Shielding Applications from an Untrusted Cloud with Haven. *ACM Trans. Comput. Syst.,* 8, Volume 33, pp. 8:1–8:26.
3. Blakely, B. & Theron, P., 2018. *Decision flow-based Agent Action Planning.* Prague, 18–20 October 2017: https://export.arxiv.org/pdf/1804.07646.
4. Booch, G., 1991. *Object-Oriented Analysis and Design with Applications.* The Benjamin Cummings Publishing Company ed. San Francisco, CA: Pearson Education.
5. Boulanin, V. & Verbruggen, M., 2017. *Mapping the development of autonomy in weapon systems,* Solna, Sweden, available at https://www.sipri.org/publications/2017/other-publications/mapping-development-autonomy-weapon-systems: SIPRI.
6. Bowen, B. M., Hershkop, S., Keromytis, A. D. & Stolfo, S. J., 2009. *Baiting Inside Attackers Using Decoy Documents.* s.l., Springer, Berlin, Heidelberg, pp. 51–70.
7. Carrasco, A. et al., 2010. Multi-agent and embedded system technologies applied to improve the management of power systems. *JDCTA,* 4(1), pp. 79–85.
8. Chen, B. & Cheng, H. H., 2010. A review of the applications of agent technology in traffic and transportation systems. *Trans. Intell. Transport. Sys.,* 11(2), pp. 485–497.
9. De Gaspari, F., Jajodia, S., Mancini, L. V. & Panico, A., 2016. *AHEAD: A New Architecture for Active Defense,* Vienna, Austria: SafeConfig'16, October 24 2016.
10. Defense Science Board, 2012. *Task Force Report: The Role of Autonomy in DoD Systems,* Washington, D.C.: Office of the Under Secretary of Defense for Acquisition, Technology and Logistics.
11. Defense Science Board, 2016. *Summer Study on Autonomy,* Washington, D.C.: Office of the Under Secretary of Defense for Acquisition, Technology and Logistics.
12. Di Pietro, R., Mancini, L. V. & Mei, A., 2003. *Random Key-assignment for Secure Wireless Sensor Networks.* New York, NY, USA, ACM, pp. 62–71.
13. Di Pietro, R., Mancini, L. V. & Mei, A., 2006. Energy Efficient Node-to-node Authentication and Communication Confidentiality in Wireless Sensor Networks. *Wireless Networks,* 11, Volume 12, pp. 709–721.
14. Disso, J. P., Jones, K. & Bailey, S., 2013. *A Plausible Solution to SCADA Security Honeypot Systems.* IEEE, Eighth International Conference on Broadband, Wireless Computing, Communication and Applications, pp. 443–448.
15. Eldefrawy, K., Francillon, A., Perito, D. & Tsudik, G., 2012. *SMART: Secure and Minimal Architecture for (Establishing a Dynamic) Root of Trust.* 19th Annual Network and Distributed System Security Symposium, February 5–8 ed. San Diego, CA: NDSS 2012.
16. Elmenreich, W., 2003. Intelligent methods for embedded systems. In: J. 2. Vienna University of Technology 2003, ed. *Proceedings of the First Workshop on Intelligent Solutions in Embedded Systems.* Austria: Vienna: Vienna University of Technology, pp. 3–11.
17. Gamma, E., Helm, R., Johnson, R. & Vlissides, J., 1994. *Design patterns: elements of reusable object-oriented software.* Reading, Massachusetts: Addison-Wesley.
18. Hofmann, O. S. et al., 2013. *InkTag: Secure Applications on an Untrusted Operating System.* New York, NY, USA, ACM, pp. 265–278.
19. Huang, H.-P., Liang, C.-C. & Lin, C.-W., 2001. Construction and soccer dynamics analysis for an integrated multi-agent soccer robot system. *Natl. Sci. Counc. ROC(A),* Volume 25, pp. 84–93.
20. Jamont, J.-P. & Occello, M., 2011. A framework to simulate and support the design of distributed automation and decentralized control systems: Application to control of indoor building comfort. In: *IEEE Symposium on Computational Intelligence in Control and Automation.* Paris, France: IEEE, pp. 80–87.
21. Jamont, J.-P., Occello, M. & Lagrèze, A., 2010. A multiagent approach to manage communication in wireless instrumentation systems. *Measurement,* 43(4), pp. 489–503.

22. Kott, A. et al., 2019. *Autonomous Intelligent Cyber-defense Agent (AICA) Reference Architecture, Release 2.0*, Adelphi, MD: US Army Research Laboratory, ARL SR-0421, September 2019, available from https://arxiv.org/abs/1803.10664.
23. LeBlanc, B., Losiewicz, P. & Hourlier, S., 2017. *A Program for effective and secure operations by Autonomous Agents and Human Operators in communications constrained tactical environments.* Prague: NATO IST-152 workshop.
24. Lin, J. & Singer, P. W., 2014. *University Tests Long-Range Unmanned Mini Sub.* [Online] Available at: https://www.popsci.com/blog-network/eastern-arsenal/not-shark-robot-chinese-university-tests-long-range-unmanned-mini-sub [Accessed 11 May 2018].
25. McArthur, S. D. et al., 2007. Multi-Agent Systems for Power Engineering Applications - Part I: Concepts, Approaches, and Technical Challenges. *IEEE TRANSACTIONS ON POWER SYSTEMS,* 22(4), pp. 1743–1752.
26. Muller, J.-P., 2004. Emergence of collective behaviour and problem solving. In: A. Omicini, P. Petta & J. Pitt, eds. *Engineering Societies in the Agents World IV.* volume 3071: Lecture Notes in Computer Science, pp. 1–20.
27. NAP, 2012. *Intelligent Human-Machine Collaboration: Summary of a Workshop,* available at http://nap.edu/13479: National Academies Press.
28. NAP, 2014. *Autonomy Research for Civil Aviation: Toward a New Era of Flight,* available at http://nap.edu/18815: National Academies Press.
29. NAP, 2016. *Protection of Transportation Infrastructure from Cyber Attacks: A Primer,* Available at http://nap.edu/23516: National Academies Press.
30. Nwana, H. S., 1996. Software agents: An overview. *The knowledge engineering review, 11(3),* pp. 205–244.
31. Papernot, N. et al., 2016. *Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks.* IEEE, 37th IEEE Symposium on Security & Privacy, pp. 582–597.
32. Pěchouček, M., Jakob, M. & Novák, P., 2010. Towards Simulation-Aided Design of Multi-Agent Systems. In: R. Collier, J. Dix & P. Novák, eds. *Programming Multi-Agent Systems.* Toronto, ON, Canada: Springer, 8th InternationalWorkshop, ProMAS 2010, 11 May 2010, Revised Selected Papers, pp. 3–21.
33. Pechoucek, M. & Marík, V., 2008. Industrial deployment of multi-agent technologies: review and selected case studies. *Autonomous Agents and Multi-Agent Systems,* Volume 17, p. 397–431.
34. Provos, N., 2004. *A Virtual Honeypot Framework.* Berkeley, USENIX Association, pp. 1–1.
35. Rasch, R., Kott, A. & Forbus, K. D., 2002. AI on the battlefield: An experimental exploration. *AAAI/IAAI.*
36. Rasch, R., Kott, A. & Forbus, K. D., 2003. Incorporating AI into military decision making: an experiment. *IEEE Intelligent Systems,* 18(4), pp. 18–26.
37. Russell, S. J. & Norvig, P., 2003. *Artificial Intelligence: A Modern Approach.* 2nd ed. Upper Saddle River, New Jersey: Prentice Hall.
38. Russell, S. J. & Norvig, P., 2010. *Artificial Intelligence: a Modern Approach.* 3rd ed. Upper Saddle River, NJ: Pearson Education.
39. Seufert, S. & O'Brien, D., 2007. *Machine Learning for Automatic Defence Against Distributed Denial of Service Attacks.* IEEE, ICC 2007 proceedings, pp. 1217–1222.
40. Silver, D. et al., 2017. Mastering the game of Go without human knowledge. *Nature,* 10, Volume 550, p. 354.
41. Sislak, D., Volf, P., Kopriva, S. & Pěchouček, M., 2012. AgentFly: Scalable, High-Fidelity Framework for Simulation, Planning and Collision Avoidance of Multiple UAVs. In: P. Angelov, ed. *Sense and Avoid in UAS: Research and Applications.* Wiley Online Library: Wiley: John Wiley&Sons, Inc., https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119964049.ch9, pp. 235-264.
42. Snyder, D. et al., 2015. *Improving the Cybersecurity of U.S. Air Force Military Systems Throughout Their Life Cycles,* Santa Monica, CA: RAND Corporation.

encrypted and anonymized. In side-channel attacks [22, 23], an attacker can infer a system's cryptographic keys via leveraging ML to analyze the power consumption, processing time, and access patterns. In membership inference attacks [24–26], an attacker can infer whether a data record is in a classifier's training dataset via leveraging ML to analyze the confidence scores of the data record predicted by the classifier or the gradient of the classifier with respect to the data record. In sensor-based location inference attacks [27, 28], an attacker can infer a user's locations via leveraging ML to analyze the user's smartphone's aggregate power consumption as well as the gyroscope, accelerometer, and magnetometer data available from the user's smartphone. In feature inference attacks [29, 30],[2] an attacker can infer a data point's missing features (e.g., an individual's genotype) via analyzing an ML model's prediction for the data point. In CAPTCHA breaking attacks [31–33], an attacker can solve a CAPTCHA via ML.

2.2 Defenses

Game-Theoretic Methods and Differential Privacy Shokri et al. [35], Calmon et al. [38], and Jia and Gong [39] proposed game-theoretic methods to defend against inference attacks. These methods rely on optimization problems that are computationally intractable when the public data is high dimensional. Salamatian et al. [47] proposed *Quantization Probabilistic Mapping (QPM)* to approximately solve the game-theoretic optimization problem formulated by Calmon et al. [38]. Specifically, they cluster public data and use the cluster centroids to represent them. Then, they approximately solve the optimization problem using the cluster centroids. Huang et al. [58] proposed to use generative adversarial networks to approximately solve the game-theoretic optimization problems. However, these approximate solutions do not have formal guarantees on utility loss of the public data. Differential privacy or local differential privacy [40–46] can also be applied to add noise to the public data to defend against inference attacks. However, as we discussed in Introduction, they achieve suboptimal privacy-utility tradeoffs because they aim to provide privacy guarantees that are stronger than needed to defend against inference attacks.

Other Methods Other methods [2, 59] leveraged heuristic correlations between the entries of the public data and attribute values to defend against attribute inference attacks in online social networks. Specifically, they modify the k entries that have large correlations with the attribute values that do not belong to the target user. k is a parameter to control privacy-utility tradeoffs. For instance, Weinsberg et al. [2] proposed BlurMe, which calculates the correlations based on the coefficients of a logistic regression classifier that models the relationship between public data entries

[2]These attacks are also called attribute inference attacks [30]. To distinguish with attribute inference attacks in online social networks, we call them feature inference attacks.

and attribute values. Chen et al. [59] proposed ChiSquare, which computes the correlations between public data entries and attribute values based on chi-square statistics. These methods suffer from two limitations: (1) they require the defender to have direct access to users' private attribute values, which makes the defender become a single point of failure, i.e., when the defender is compromised, the private attribute values of all users are compromised; and (2) they incur large utility loss of the public data.

3 Problem Formulation

We take attribute inference attacks in online social networks as an example to illustrate how to formulate the problem of defending against inference attacks. However, our problem formulation can also be generalized to other inference attacks. We have three parties: *user*, *attacker*, and *defender*. Next, we discuss each party one by one.

3.1 User

We focus on protecting the private attribute of one user. We can protect different users separately. A user aims to publish some data while preventing inference of its private attribute from the public data. We denote the user's public data and private attribute as $\mathbf{x}$ (a column vector) and s, respectively. For instance, an entry of the public data vector $\mathbf{x}$ could be the rating score the user gave to an item or 0 if the user did not rate the item; an entry of the public data vector could also be 1 if the user liked the corresponding page or 0 otherwise. For simplicity, we assume each entry of $\mathbf{x}$ is normalized to be in the range $[0, 1]$. The attribute s has m possible values, which we denote as $\{1, 2, \cdots, m\}$; $s = i$ means that the user's private attribute value is i. For instance, when the private attribute is political view, the attribute could have two possible values, i.e., democratic and republican. We note that the attribute s could be a combination of multiple attributes. For instance, the attribute could be $s =$ (politicalview, gender), which has four possible values, i.e., (democratic, male), (republican, male), (democratic, female), and (republican, female).

Policy to Add Noise Different users may have different preferences over what kind of noise can be added to their public data. For instance, a user may prefer modifying its existing rating scores, while another user may prefer adding new rating scores. We call a policy specifying what kind of noise can be added a *noise-type-policy*. In particular, we consider the following three types of noise-type-policy.

- **Policy A: Modify_Exist.** In this policy, the defender can only modify the non-zero entries of $\mathbf{x}$. When the public data are rating scores, this policy means that the defender can only modify a user's existing rating scores. When the public data

4 Design of AttriGuard

4.1 Overview

The major challenge to solve the optimization problem in Eq. (1) is that the number of parameters of the mechanism $\mathcal{M}$, which maps a given vector to another vector probabilistically, is exponential to the dimensionality of the public data vector. To address the challenge, Jia and Gong [39] proposed AttriGuard, a *two-phase framework* to solve the optimization problem approximately. The intuition is that, although the noise space is large, we can categorize them into m groups depending on the defender's classifier's output. Specifically, we denote by G_i the group of noise vectors such that if we add any of them to the user's public data, then the defender's classifier will infer the attribute value i for the user. Essentially, the probability q_i that the defender's classifier infers attribute value i for the user is the probability that $\mathcal{M}$ will sample a noise vector in the group G_i, i.e., $q_i = \sum_{\mathbf{r} \in G_i} \mathcal{M}(\mathbf{r}|\mathbf{x})$. AttriGuard finds one representative noise vector in each group and assumes $\mathcal{M}$ is a probability distribution concentrated on the representative noise vectors.

Specifically, in Phase I, for each group G_i, AttriGuard finds a minimum noise $\mathbf{r}_i$ such that if we add $\mathbf{r}_i$ to the user's public data, then the defender's classifier predicts the attribute value i for the user. AttriGuard finds a minimum noise in order to minimize utility loss. In *adversarial machine learning*, this is known as *adversarial example*. However, existing adversarial example methods [50, 52–55] are insufficient to find the noise vector $\mathbf{r}_i$, because they do not consider the noise-type-policy. AttriGuard optimizes the adversarial example method developed by Papernot et al. [54] to incorporate noise-type-policy. The noise $\mathbf{r}_i$ optimized to evade the defender's classifier is also likely to make the attacker's classifier predict the attribute value i for the user, which is known as *transferability* [50–53] in adversarial machine learning.

In Phase II, AttriGuard simplifies the mechanism $\mathcal{M}^*$ to be a probability distribution over the m representative noise vectors $\{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_m\}$. In other words, the defender randomly samples a noise vector $\mathbf{r}_i$ according to the probability distribution $\mathcal{M}^*$ and adds the noise vector to the user's public data. Under such simplification, $\mathcal{M}^*$ only has at most m non-zero parameters, the output probability distribution $\mathbf{q}$ of the defender's classifier essentially becomes $\mathcal{M}^*$, and we can transform the optimization problem in Eq. (1) to be a convex problem, which can be solved efficiently and accurately. Moreover, Jia and Gong derived the analytical forms of the solution using the *Karush–Kuhn–Tucker (KKT) conditions* [56], which shows that the solution is intuitive and interpretable.

learning to defend against inference attacks in various domains. For the adversarial machine learning community, there are opportunities to develop new adversarial machine learning methods that consider the unique privacy and utility-loss challenges. For the privacy community, adversarial machine learning brings new opportunities to achieve better privacy-utility tradeoffs. For the security community, adversarial machine learning brings new opportunities to enhance system security such as designing more secure and usable CAPTCHAs as well as mitigating side-channel attacks. Specifically, we envision that AttriGuard's two-phase framework can be applied to defend against other inference attacks, e.g., the ones we discussed in Sect. 2.1. However, Phase I of AttriGuard should be adapted to different inference attacks, as different inference attacks may have their own unique privacy, security, and utility requirements on the representative noise vectors. Phase II can be used to satisfy the utility-loss constraints via randomly sampling a representative noise vector according to a certain probability distribution. We note that some recent studies [61, 62] have tried to leverage adversarial examples to defend against website fingerprinting attacks and side-channel attacks. However, they did not consider the utility-loss constraints, which can be satisfied by extending their methods using Phase II of AttriGuard. Moreover, recent studies [63, 64] have explored adversarial example based defenses against author identification attacks for programs.

Data Poisoning Attacks Based Defenses Other than adversarial examples, we could also leverage data poisoning attacks [65–72] to defend against inference attacks. Specifically, an attacker needs to train an ML classifier in inference attacks. For instance, in attribute inference attacks on social networks, an attacker may train a classifier via collecting a training dataset from users who disclose both public data and attribute values. In such scenarios, the defender could inject fake users with carefully crafted public data and attribute values to poison the attacker's training dataset such that the attacker's learnt classifier is inaccurate. In other words, the defender can perform data poisoning attacks to the attacker's classifier. For instance, an online social networking service provider could inject such fake users to defend against inference attacks performed by third-party attackers.

Adaptive Inference Attacks We envision that there will be an arms race between attackers and defenders. Specifically, an attacker could adapt its attacks when knowing the defense, while a defender can further adapt its defense based on the adapted attacks. For instance, an attacker could first detect the noise added to the public data or detect the fake users, and then the attacker performs inference attacks. Jia and Gong tried a low-rank approximation based method to detect the noise added by AttriGuard and AttriGuard is still effective against the method. However, this does not mean an attacker cannot perform better attacks via detecting the noise. An attacker could also leverage fake-user detection (also known as Sybil detection) methods (e.g., [73–82]) to detect and remove the fake users when the defender uses data poisoning attacks as defenses. We believe it is an interesting future work to systematically study the possibility of detecting noise and fake users

References

1. Jahna Otterbacher. Inferring gender of movie reviewers: exploiting writing style, content and metadata. In *CIKM*, 2010.
2. Udi Weinsberg, Smriti Bhagat, Stratis Ioannidis, and Nina Taft. Blurme: Inferring and obfuscating user gender based on ratings. In *RecSys*, 2012.
3. E. Zheleva and L. Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *WWW*, 2009.
4. Abdelberi Chaabane, Gergely Acs, and Mohamed Ali Kaafar. You are what you like! information leakage through users' interests. In *NDSS*, 2012.
5. Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *PNAS*, 2013.
6. Neil Zhenqiang Gong, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui Chul Richard Shin, Emil Stefanov, Elaine(Runting) Shi, and Dawn Song. Joint link prediction and attribute inference using a social-attribute network. *ACM TIST*, 5(2), 2014.
7. Neil Zhenqiang Gong and Bin Liu. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *USENIX Security Symposium*, 2016.
8. Jinyuan Jia, Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. AttriInfer: Inferring user attributes in online social networks using markov random fields. In *WWW*, 2017.
9. Neil Zhenqiang Gong and Bin Liu. Attribute inference attacks in online social networks. *ACM TOPS*, 21(1), 2018.
10. Yang Zhang, Mathias Humbert, Tahleen Rahman, Cheng-Te Li, Jun Pang, and Michael Backes. Tagvisor: A privacy advisor for sharing hashtags. In *WWW*, 2018.
11. Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *IEEE S&P*, 2012.
12. Mathias Payer, Ling Huang, Neil Zhenqiang Gong, Kevin Borgolte, and Mario Frank. What you submit is who you are: A multi-modal approach for deanonymizing scientific publications. *IEEE Transactions on Information Forensics and Security*, 10(1), 2015.
13. Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. De-anonymizing programmers via code stylometry. In *USENIX Security Symposium*, 2015.
14. Aylin Caliskan, Fabian Yamaguchi, Edwin Tauber, Richard Harang, Konrad Rieck, Rachel Greenstadt, and Arvind Narayanan. When coding style survives compilation: De-anonymizing programmers from executable binaries. In *NDSS*, 2018.
15. Rakshith Shetty, Bernt Schiele, and Mario Fritz. A4nt: Author attribute anonymity by adversarial training of neural machine translation. In *USENIX Security Symposium*, 2018.
16. Mohammed Abuhamad, Tamer AbuHmed, Aziz Mohaisen, and DaeHun Nyang. Large-scale and language-oblivious code authorship identification. In *CCS*, 2018.
17. Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *ACM Workshop on Cloud Computing Security*, 2009.
18. Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *ACM workshop on Privacy in the Electronic Society*, 2011.
19. Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *CCS*, 2012.
20. Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. In *CCS*, 2014.
21. Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting. In *USENIX Security Symposium*, 2014.


image
not
available


image
not
available


image
not
available

Fig. 1 Overview of coevolutionary adversarial AI framework concept. The coevolutionary component performs search over the actions of adversary controllers. The engagement component evaluates the strategies of the adversaries and returns the measurements of the engagement

Coevolutionary search methods results in population-wide adversarial dynamics. Such dynamics can expose adversarial behaviors for a defense to anticipate.

We present an extension of a framework called RIVALS that we previously have used to generate robust defensive configurations [31]. It is composed of different coevolutionary algorithms to help it generate diverse behavior. The algorithms, for further diversity, use different "solution concepts", i.e. measures of adversarial success and quality measures.

One way to evaluate solutions in a multi-player setting is to consider Nash equilibria. These are points which satisfy every player's optimizing condition given the other players' choices. That is, a player does not have incentive to deviate from its strategy given the other players' strategies. This concept has been used to understand the strategic actions of multiple players in a deterministic gaming environment [28]. We can model different threat scenarios in RIVALS and Nash equilibria may offer insight into possible outcomes in the attacker-defender coevolution.

The RIVALS framework supports a number of threat scenario use-cases using simulation and emulation of varying model granularity. These include:

(a) Defending a peer-2-peer network against Denial of Service (DOS) attacks [13, 40]
(b) Defenses against spreading device compromise in a segmented enterprise network [15], and
(c) Deceptive defense against the internal reconnaissance of an adversary within a software defined network [16]

The RIVALS framework is linked to a decision support module named `ESTABLO` [35, 40]. The engagements of every run of any of the coevolutionary algorithms are cached and, later, `ESTABLO` collects adversaries from the cache for its *compendium*. It then evaluates all the adversaries of each side against those of the other side inn the environment and ranks them according to multiple criteria. It can also provide comparisons of adversarial behaviors. This information can be incorporated in the decision process of a defensive manager.

14. S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving target defense: creating asymmetric uncertainty for cyber threats*. Springer Science & Business Media, 2011, vol. 54.
15. G. S. Kc, A. D. Keromytis, and V. Prevelakis, "Countering code-injection attacks with instruction-set randomization," in *Proceedings of the 10th ACM conference on Computer and communications security*. ACM, 2003, pp. 272–280.
16. A. Clark, Q. Zhu, R. Poovendran, and T. Başar, "Deceptive routing in relay networks," in *International Conference on Decision and Game Theory for Security*. Springer, 2012, pp. 171–185.
17. H. Maleki, S. Valizadeh, W. Koch, A. Bestavros, and M. van Dijk, "Markov modeling of moving target defense games," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*. ACM, 2016, pp. 81–92.
18. C. R. Hecker, "A methodology for intelligent honeypot deployment and active engagement of attackers," Ph.D. dissertation, 2012.
19. Q. D. La, T. Q. Quek, J. Lee, S. Jin, and H. Zhu, "Deceptive attack and defense game in honeypot-enabled networks for the internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1025–1035, 2016.
20. J. Pawlick, T. T. H. Nguyen, and Q. Zhu, "Optimal timing in dynamic and robust attacker engagement during advanced persistent threats," *CoRR*, vol. abs/1707.08031, 2017. [Online]. Available: http://arxiv.org/abs/1707.08031
21. J. Pawlick and Q. Zhu, "A Stackelberg game perspective on the conflict between machine learning and data obfuscation," in *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*. IEEE, 2016, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7823893/
22. Q. Zhu, A. Clark, R. Poovendran, and T. Basar, "Deployment and exploitation of deceptive honeybots in social networks," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 212–219.
23. Q. Zhu, H. Tembine, and T. Basar, "Hybrid learning in stochastic games and its applications in network security," *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, pp. 305–329, 2013.
24. Q. Zhu, Z. Yuan, J. B. Song, Z. Han, and T. Başar, "Interference aware routing game for cognitive radio multi-hop networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 10, pp. 2006–2015, 2012.
25. Q. Zhu, L. Bushnell, and T. Basar, "Game-theoretic analysis of node capture and cloning attack with multiple attackers in wireless sensor networks," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 3404–3411.
26. Q. Zhu, A. Clark, R. Poovendran, and T. Başar, "Deceptive routing games," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 2704–2711.
27. Q. Zhu, H. Li, Z. Han, and T. Basar, "A stochastic game model for jamming in multi-channel cognitive radio systems." in *ICC*, 2010, pp. 1–6.
28. Z. Xu and Q. Zhu, "Secure and practical output feedback control for cloud-enabled cyber-physical systems," in *Communications and Network Security (CNS), 2017 IEEE Conference on*. IEEE, 2017, pp. 416–420.
29. ——, "A Game-Theoretic Approach to Secure Control of Communication-Based Train Control Systems Under Jamming Attacks," in *Proceedings of the 1st International Workshop on Safe Control of Connected and Autonomous Vehicles*. ACM, 2017, pp. 27–34. [Online]. Available: http://dl.acm.org/citation.cfm?id=3055381
30. ——, "Cross-layer secure cyber-physical control system design for networked 3d printers," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 1191–1196. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7525079/
31. M. J. Farooq and Q. Zhu, "Modeling, analysis, and mitigation of dynamic botnet formation in wireless iot networks," *IEEE Transactions on Information Forensics and Security*, 2019.
32. Z. Xu and Q. Zhu, "A cyber-physical game framework for secure and resilient multi-agent autonomous systems," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 5156–5161.