Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.835 – 840

RESEARCH ARTICLE

Optimizing the Performance and Secure Distributed Wireless Network in Unreliable D-NCS using CGA

Babu Pinjar
Second year M.Tech (CNE),
Computer Science & Engineering Department
The National Institute of Engineering College, Mysore, INDIA

C.N.Chinnaswamy
Associate Professor,
Information Science & Engineering Department
The National Institute of Engineering College, Mysore, INDIA

Abstract- Distributed network-control-systems (D-NCS) are a network structure and components that are capable of integrating sensors, actuators, communication, and control algorithms to suit real-time applications. Distributed networked control systems (D-NCS) are vulnerable to various network attacks when the network is not secured, thus, D-NCS must be well protected with security mech- anisms (e.g., cryptography), which may adversely affect the dynamic performance of the D-NCS because of limited system resources. This paper is concerned with the problem of designing a secure distributed control methodology that is capable of performing a secure consensus computation in a D-NCS in the presence of misbehaving nodes. It embeds four phases (detection, mitigation, identification, and update) into the distributed control process. We use a wireless network based, robot navigation path tracking system called Intelligent Space (iSpace) as a D-NCS test bed in this paper. Network security algorithms DES and 3DES are integrated with the application to secure the sensitive information flow. Then, the paper proposes a paradigm of the performance-security tradeoff optimization based on the Coevolutionary genetic algorithm (CGA) for D-NCS.

Index Terms—Coevolutionary genetic algorithm (CGA), distributed networked control systems (D-NCS), ISpace, secure distributed control, data security

I. INTRODUCTION

Distributed Networked Control Systems (D-NCS) consist of spatially distributed sensors, actuators, and controllers communicate with each other through a shared network. Such systems are widely deployed in the physical world. The widespread growth of wireless communication and embedded systems technologies have been creating a variety of applications for D-NCS, such as defense systems, autonomous vehicles, and smart structures of national critical infrastructures (e.g., electrical power systems, transportation systems, defense systems, traffic management). D-NCS are more vulnerable to cyber attacks with the increasingly usage of internet, embedded systems, and novel distributed control strategies. Many of these applications are time-sensitive, data-sensitive, and safety-critical. The potential consequences of compromising the D-NCS can be devastating to public health and safety, national security, and the economy. Compromised D-NCS can lead to extensive cascading power outages, dangerous toxic chemical releases, and explosions. It is, therefore, important to implement D-NCS with security controls that make reliable, safe, and flexible performance possible.

Recently, D-NCS have witnessed a paradigm shift from centralized to distributed control design, propelled by advances in wireless communication technologies and low cost, high performance embedded systems. More and more distributed control algorithms are developed and used in D- NCS due to their scalability and computation advantages. These algorithms, however, increase the vulnerability of D-NCS to malicious cyber attacks, as well. Because of the lack of a centralized entity that may monitor the activity of the nodes in the network, distributed control strategies are prone to attacks and component failures. Besides, most of the current effort for protecting D-NCS, e.g., Supervisory Control And Data Acquisition (SCADA) systems, has been accomplished in prevention mechanisms (e.g., cryptography and authentication) and limited to platform and communication security. There is an urgent growing concern to protect the control algorithms from malicious cyber attacks.

A variety of distributed control algorithms, such as synchronized consensus algorithms and asynchronous gossip algorithms, have been proposed and studied in D-NCS. In D-NCS, to agree upon a certain performance measure among autonomous agents is a typical task, e.g., the computation load on a network of parallel computers, the velocity for a group of autonomous robots. Thus, the linear consensus algorithm is considered in this paper. In this algorithm, the state of each node is updated at each time step, which is a weighted average of its own state and those received from its neighbor nodes. The choice of topologies and weights influences the convergence speed toward the consensus value. In this algorithm, all the nodes are assumed to cooperate and follow the protocol exactly; otherwise, the consensus is not guaranteed to be reached. It is important to guarantee secure computation in the face of failures and intrusions of the linear consensus algorithm. The wireless NCS application on the basis of security effect on NCS performance to show the trade-off between security addition and real-time operation of D-NCS. To display the effect of addition of security in NCS, we developed a closed-loop network-controlled path tracking navigation system called intelligent space (iSpace) as a test bed. iSpace consists of different modules such as image processing, automation control, and communication for navigation of a differential drive unmanned ground vehicle (UGV) in an indoor 2-D environment. Determining how to achieve the optimized balance between performance and security on D-NCS is an open question. As a fast-developing optimization algorithm, the co evolutionary genetic algorithm (CGA) is an extension of conventional evolutionary algorithms. It models an ecosystem consisting of two or more species. Since most of the current effort for protecting D-NCS has been accomplished with prevention mechanisms (e.g., cryptography), this paper focuses on the confidentiality aspect of security service using secret key cryptography. We also suggest an optimization approach for a multi agent performance and security tradeoff based on the CGA.

II. SYSTEM DESCRIPTION AND IMPLEMENTATION

In the wireless systems, several security protocols such as wired equivalent privacy (WEP) and 802.Ix port access control with extensible authentication protocol (EAP) support are proposed to address security issues. Moreover, due to the strong security it provides in wired networks, the IP security protocol (IPsec) is considered as a good option for wireless systems as well. Some basic security features associated with these security protocols as defined by IETF for wireless systems. A distributed NCS consists of many different components such as sensors to collect data, controllers to process data in the desired manner and actuators to perform tasks asked by the controllers. We have used iSpace as a network based intelligent control platform for our experiments for two different security protocols so that we can characterize the system behavior. The more detailed explanation of iSpace therefore follows in this section.

A. iSpace as a D-NCS testbed

iSpace at the Advanced Diagnosis, Automation, and Control (ADAC) Lab at North Carolina State University (NCSU) has been implemented to perform basic research and education on time-sensitive and secure D-NCS with hardware-in-the-loop fast-prototyping capabilities. iSpace is a networked-controlled path tracking integrated navigation system for a UGV. The destination point is chosen by the user via any remote computing interface in the world by accessing the iSpace GUI. iSpace aggregates the status of the entire space from each agent's sensory information and responds intelligently to the system's goals. ISpace has several major components: distributed sensors and actuators, hybrid hierarchical and distributed controllers, the system's security manager, and the communication network.

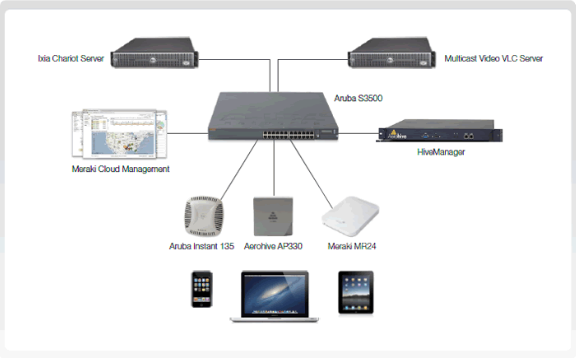


Fig: iSpace testbed

B. Data Encryption

Encryption algorithms for wireless systems can be broadly classified into two categories: symmetric and asymmetric ciphers. Encryption algorithms, which require a single cryptographic key to encrypt and decrypt the data, are termed as symmetric ciphers. The Data Encryption standard (DES) and the Advanced Encryption Standard (AES) are two symmetric ciphers which are considered secure for wireless systems. However, the issue associated with symmetric ciphers is how to transmit cryptographic keys securely to the legitimate recipient. On the other hand, asymmetric ciphers are the ones which require different cryptographic keys for encryption and decryption of data. For example, public key encryptions, such as RSA, employ asymmetric ciphers. In public key encryption, each user has a pair of public and private keys. The public key is published while the private key is kept confidential. Messages are encrypted using the public key of the recipient, and can be decrypted only by the private key of the recipient.   In this work also, we implement DES and 3DES (a stronger variant of DES) for encrypting and decrypting the data securely over NCS. Data Encryption Standard (DES) incorporates a 64- bit key to encrypt and decrypt the transmitted data. However, as the least significant bit of each byte in the key is a parity bit, that bit from each byte is ignored during actual encryption and decryption process, leading to reduction of 8 bits from 64-bit key. Therefore, DES uses 56-bit key effectively. Besides, DES, being a block cipher, operates on 64-bit blocks of plaintext as input and outputs 64-bit blocks of encrypted blocks. In 3DES, first each 64-bit block is encrypted by using first 64-bit key, then decrypted using second 64-bit key and then encrypted again by using third 64-bit key. For decryption at the receiver side, the entire process is reversed to obtain the original form. Consequently, 3DES is 3 times slower than DES.

C. Formation Control Model

Consider there are n mobile robots in the network where all the robots exchange information with their neighbors through the communication network to maintain a certain formation in a distributed manner.

The linear dynamics of the robots are denoted by

$$x_i\,[k+1] = Ax_i\,[k] + Bu_i\,[k], \qquad (1)$$

where, xi and ui are the robot states and controls and i is the index for the robots in the group. Each robot receives the following measurements:

$$y_i\,[k] = Cx_i\,[k]. \qquad (2)$$

To maintain a certain formation of these robots, each robot implements a reference-based proportional controller on-board:

$$u_i\,[k] = K_p\, e_i\,[k] = K_p\,(r_i\,[k]\text{- } y\,[k]), \qquad (3)$$

where ri is the reference robot state and Kp is the proportional gain for the P controller to be designed.

B. Malicious Attack Models

The misbehaving robot in the system is modeled as one whose consensus manager is suffering a malicious input caused by cyber attacks. Thus the misbehaving robots in the system are formulated as:

$$z_i[k+1] = \sum\nolimits^{n}_{j=1} d_{ij} z_j[k] + B_M u_M[k], \qquad (4)$$

where u_M, the cyber attack input, is modeled as a malicious exogenous input to the consensus manager.

The malicious cyber attacks can be classified into two categories depending on their abilities. One type of malicious attacks simply makes the compromised robot stop updating its reference state and behave like a "faulty" robot. For example, the adversary stops the robot during its movement towards the goal point or even before it has started. The compromised robot will remain stationary indefinitely. Thus, this attack is denoted as the fault attack of which the exogenous input uM can be modeled as:

$U_M[k]=c-\sum^{n}_{j=1}d_{ij}z_j[k],$ (5)

Where, c is an arbitrary constant value and c R.

Another type of malicious attack is the false data injection attack. The robot under this attack may behave in an arbitrary manner. Thus, the malicious input u_M can be equal to any arbitrary value c[k] that the cyber attacker wants:

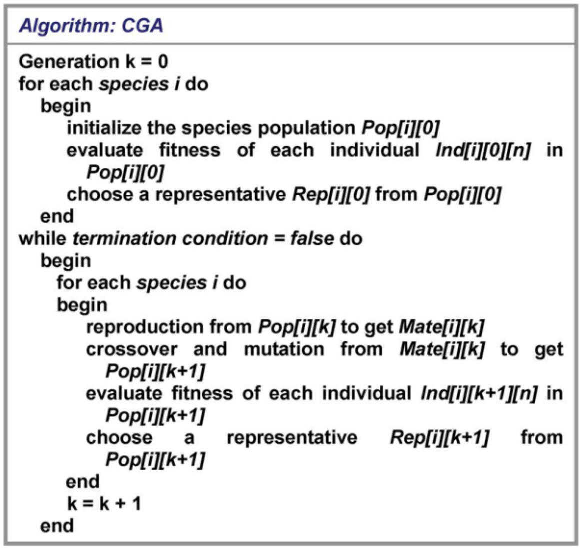
$U_M[k]=c[k],$ (6)

III. OPTIMIZATION USING THE CGA

A. Co evolutionary Genetic Algorithm

Genetic algorithms inspired by nature, the concept of co evolution—used as the foundation for the CGA comes from biological observations. Nature is composed of several species that co evolve

THE PSEUDO-CODE OF THE CGA



```
Algorithm: CGA
Generation k = 0
for each species i do
  begin
      initialize the species population Pop[i][0]
      evaluate fitness of each individual Ind[i][0][n] in
         Pop[i][0]
      choose a representative Rep[i][0] from Pop[i][0]
  end
while termination condition = false do
  begin
     for each species i do
     begin
         reproduction from Pop[i][k] to get Mate[i][k]
         crossover and mutation from Mate[i][k] to get
            Pop[i][k+1]
         evaluate fitness of each individual Ind[i][k+1][n] in
            Pop[i][k+1]
         choose a representative Rep[i][k+1] from
            Pop[i][k+1]
     end
     k = k + 1
  end
```

The pseudo-code of the CGA is shown above, in which the evolution of each species is handled by a standard GA, while the evaluation of an individual from each species is handled through collaboration with representatives from other species.

B. Performance-Security Model Based on CGA

The performance security model of D-NCS based on the CGA. Since all the objectives in the tradeoff objective function are equally weighted, all the agents interact with each other through the environment and form a non cooperative game, each agent submits its decision variables to the system environment model and takes its following actions based on the knowledge of itself and the system response from the environment. Here, the system environment consists of the total bandwidth of system has.

The CGA optimization process of D-NCS, each agent is represented by a species (a species means a population of GA in this algorithm) in the ecosystem. Each species evolves a bundle of individuals that represent the candidate competing strategies—decision variables of the corresponding agent. Each species is evolved through the repeated application of a con ventional GA. For the representative selection, there are many possible methods for choosing the representatives with which to col- laborate. In order to facilitate the fast convergence of the evolutionary process, we use a "greedy" method for selecting representatives. From the evaluation process above, we can see that the species are coordinated by the system environment response. When one agent changes its decision variables to gain a better fitness value, it will change the system response according to system dynamics and, in turn, change the fitness values of the other agents. Other agents will behave in the same way. The adjustment process will continue until no agent can gain better fitness value by changing its own decision variables without changes of the decision variables of other agents.

IV. SECURE DISTRIBUTED CONTROL

A secure distributed control methodology is proposed in this section to achieve attack-resilience in D- NCS in a fully distributed fashion.

A. Detection Phase

A Neighborhood Monitor is an embedded monitor for an agent to observe the behaviors of its neighbors.

In this Neighborhood Monitor, the robot carries out a real-time anomaly detection mechanism for all its neighbors (similar to the watchdog design in the security mechanism of ad hoc networks). It redundantly calculates and stores its neighbor robot j's reference state r_j and compares it with the state value z_j received from the neighbor robot j in time step k.

$G_{ij}[k] = G_{ij}[k-1]+1, r_j[k] – z_j[k] = 0,$
$G_{ij}[k -1], r_j[k] – z_j[k] \neq 0$ (7)

where Gij is the total number of verifiably correct robot states of neighbor robot j up to time-step k monitored by robot i.

B. Mitigation Phase

A Local Reputation Manager is an onboard system for an agent that updates the reputation values of the neighboring agents and records them in its local reputation table.

Reputation is an index for the reliability of a node in the network; it is widely used in the cooperation issues among the nodes of an ad hoc network. The reputation metric is introduced to quantitatively measure the credibility of the neighbor robots in this paper and the Bayesian Reputation function is used to calculate the reputation values. If misbehaviors of one neighboring robot are detected, the Neighborhood Monitor reports them to the Local Reputation Manager. In this phase, a good behavior or misbehavior as the output result of the Neighborhood Monitor is used to update the neighboring robot's reputation value in each agent's local Reputation Manager.

C. Identification Phase

In this phase, if the reputation value of one neighboring robot falls below a certain level, defined as the malicious threshold, that robot is identified as a compromised robot and then will be isolated. In that case, all the information from the misbehaving neighboring agent will be rejected. Actually, the reputation manager acts as a confirmation mechanism that one agent is confirmed as misbehaving if it is detected with abnormal values consecutively during a certain period of time, which is interpreted as its reputation value drops below the malicious threshold.

D. Update Phase

In order to embed the above security mechanisms into each iteration of the control computation process, an update phase is proposed to adaptively update the consensus computation weights d_{ij} based on the reputation values from the Local Reputation Manager. The update rule is shown as:

$D_{ij}[k] = rep_{ij}[k]/\sum^{n}_{j=1} rep_{ij}[k],$ (8)

Thus, the robot reference state update rule in the consensus manager is correspondingly changed to:

$Z_i[k+1] = \sum^{n}_{j=1} d_{ij}[k] z_j[k], I =1,….,n,$ (9)

In summary, the consensus manager of each robot will gradually decrease the corresponding consensus computation weights when the neighboring robot's reputation drops, in order to slow down the speed at which the malicious effects of that potentially compromised robot spreads. Furthermore, when a neighboring robot is identified as a compromised agent, the consensus manager will set the consensus computation weight of that robot to zero and cut off the connection to isolate that misbehaving robot.

V. CONCLUSION

This paper considers the distributed control problem in an unreliable D-NCS. This paper proposes a secure distributed control methodology that embeds the distributed security mechanism internally. It allows all the well-behaving robots to reach the consensus state in the presence of a misbehaving robot. This paper addresses the performance and security tradeoff problem of D-NCS and proposes a tradeoff model for performance and security in the D-NCS, as well as a paradigm for multi agent tradeoff optimization based on the CGA. The CGA paradigm provides satisfactory modeling and optimization results for the performance-security tradeoff on the iSpace system.

REFERENCES

[1] R. A. Gupta and M.Y. Chow, "Networked Control System: Overview and Research Trends," IEEE Transactions on Industrial Electronics, vol. 57, pp. 2527-2535, 2010.

[2] A. Ulusoy, O. Gurbuz, and A. Onat, “Wireless model-based predictive networked control system over cooperative wireless network,” IEEE Trans. Ind. Informat., vol. 7, no. 1, pp. 41–51, Feb. 2011.

[3] “Common cyber security vulnerabilities observed in control system assessments by the INL NSTB program,” U.S. Department of Energy Office of Electricity, Delivery, and Energy Reliability, Washington, DC, 2008 [Online]. Available: http://www.controlsystemsroadmap.net/pdfs/INL_Common_Vulnerabilties.pdf

[4] M. Lin, L. Xu, L. T. Yang, X. Qin, N. Zheng, Z. Wu, and M. Qiu, “Static security optimization for real-time systems,” IEEE Trans. Ind. Informat., vol. 5, no. 1, pp. 22–37, Feb. 2009.

[5]W.Granzer,F.Praus,andW.Kastner,“Securityinbuildingautomation systems,” IEEE Trans. Ind. Electron., vol. 57, no. 11, pp. 3622–3630, Nov. 2010.

[6] P. Marti, C. Lin, S. A. Brandt, M. Velasco, and J. M. Fuertes,“Draco:Efficientresourcemanagementforresource-constrainedcontroltasks,” IEEE Trans. Comput., vol. 58, no. 1, pp. 90–105, Jan. 2009.

[7] C. K. Goh and K. C. Tan, “A competitive-cooperative coevolutionary paradigm for dynamic multi objective optimization,”IEEE Trans.Evol. Comput. vol. 13, no. 1, pp. 103–127, Feb 2009.

[8] M. A. Potter and K. A. De Jong, “Cooperative coevolution: An architecture for evolving coadapted subcomponents,”J.Evol.Comput.,vol. 8, no. 1, pp. 1–29, 2000.

[9] D. He, C. Chen, S. Chan, J. Bu, and L. T. Yang, "Security Analysis and Improvement of a Secure and Distributed Reprogramming Protocol for Wireless Sensor Networks," IEEE Transactions on Industrial Electronics, vol. 60, pp. 5348-5354, 2013.

[10] S. X. Ding, Z. Ping, Y. Shen, and E. L. Ding, "An Integrated Design Framework of Fault-Tolerant Wireless Networked Control Systems for Industrial Automatic Control Applications," IEEE Transactions on Industrial Informatics, vol. 9, pp. 462-471, 2013.