Y9.ET1.3 Implementation of Secure Energy Management against Cyber/physical Attacks for FREEDM System

Project Leader: Dr. Bruce McMillin

Faculty: Dr. Mo-Yuen Chow

Students: Jie Duan

1. Project Goals

- Develop a resilient cyber-physical control strategy for FREEDM system to secure the energy scheduling and physical operation:
  - ○ Extending to cyber-physical security: Explore possible attacks on both energy scheduling and physical operation, analyze the impacts in terms of economic benefit and system stability.
  - ○ Interaction between cyber/physical layer detection: Investigate the correlation between the cyber layer behavior and physical layer response, inconsistent behaviors in both layer are helpful to disclose the misconduct devices.
  - ○ Collaborative cyber-physical countermeasures: Integrating the cyber layer resilience with physical layer secure algorithms to build a resilient cyber-physical control framework against possible attacks.
- Implement the resilient cyber-physical control framework in DGI platform and further test the algorithm in HIL and GEH testbeds.

2. Role in Support of Strategic Plan

This project closes the control loop for the distributed energy management to remain optimal in the presence of cyber attacks. Moreover, this project provides a tangible demonstration of how the cooperative distributed energy management should be implemented in HIL/GEH testbed.

3. Fundamental Research, Technological Barriers and Methodologies

The technology barrier is how to build a distributed monitoring system to fit for the distributed control framework. The methodology to deal with it is to develop a neighborhood-watch mechanism in which each node is responsible for monitoring its neighbors. A reputation index is introduced to reflect the credibility of the neighbors, if one neighbor is continuing sending out false information, the reputation index will indicate that it is compromised.

4. Achievements

4.1 New Data Integrity Attack on CoDES algorithm

The CoDES algorithm [1] is a fully distributed optimization method, in which the DGI nodes communicate with only neighbors and determine the local generation schedule. It brings some significant advantages to the system, in terms of scalability and robustness [2].

However, the fully distributed framework is also vulnerable to malicious cyber attacks, in which some devices might choose not to collaborate with neighbors, but to seek for selfish objectives [3]. For example, Fig.1 shows a selfish DESD (DESD 1) in the FREEDM GEH system who wants to maximize its own economic benefit:

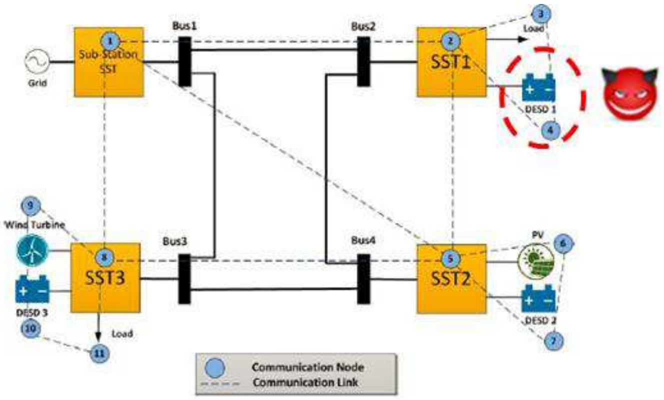


Figure 1: A malicious DESD in FREEDM system

Step 1: Determining the most profitable schedule for itself

The selfish objective of the malicious DESD is given as:

$$\max \quad \sum_{t=1}^{T} p(t) P_M^a(t) \Delta t$$

$$s.t. \quad \forall t \in \{1,...,T\}, x_0 - E_{full} \le \sum_{s=1}^{t} P_M^a(s) \Delta t \le x_0,$$

$$\forall t \in \{1,...,T\}, P_{M,B\min} \le P_M^a(t) \le P_{M,B\max}.$$

where $P_M^a(t)$ denotes the power command to the malicious storage device M at time step t, a positive value means a discharging command; $p(t)$ is the energy price at time step t; $E_{i,full}$ is the storage capacity and x_{i0} is initial value of the stored energy. $P_{i,Bmin}$ and $P_{i,Bmax}$ are the minimum and maximum power limits of the storage device.

Step 2: Manipulating the power imbalance estimation

In the CoDES algorithm, all the devices estimate the system power imbalance in a collective sense, using a consensus network. Thus, attacker M could use false local power balance estimation to mislead the system to overestimate or underestimate the system power imbalance.

Assume attacker M sends out false local power imbalance estimation $\Delta\hat{P}_M^{k,f}(t)$ in iteration k, and the neighbors of device M use $\Delta\hat{P}_M^{k,f}(t)$ for their local update, while the attacker M still updates using the correct information $\Delta\hat{P}_M^k(t)$. In this case, due to the false information $\Delta\hat{P}_M^{k,f}(t)$, from iteration $k+1$:

$$\sum_{i} \Delta\hat{P}_i^{k+1}(t) = \Delta P_{sys}^{k+1}(t) + \sum_{j \in N_M} w_{Mj} \left(\Delta\hat{P}_M^{k,f}(t) - \Delta\hat{P}_M^{k}(t) \right),$$

which means the local estimation of system power imbalance deviates from the actual value.

When attacker M is launching the data integrity attack, it keeps the actual scheduling command $P_M^k(t)$ to be 0, appearing not to be contributing to the microgrid. At the same time, it manipulates the deviation to be exactly the same as $\mathbf{P}_M^a$. The impact of this attack is illustrated in Fig.2.

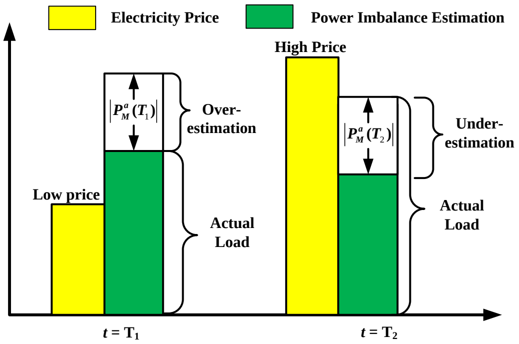


Figure 2: Manipulation of power imbalance estimation

When the electricity price is low ($t = T_1$), the system is misled to overestimate the system load, with a difference as $|P_M^a(T_1)|$. In contrast, when the electricity price is high ($t = T_2$), the system underestimates the system load, with a difference as $|P_M^a(T_2)|$. Consequently, while the normal devices adjust their power generations to support the false load, and attacker M charges excess power when the electricity price is low and discharges when the electricity price is high.

With the false load estimations, the normal devices have to adjust their schedule in order to meet the false load, while DESD 1 charges and discharges according to the malicious schedule $\mathbf{P}_M^a(t)$. We illustrate the impact of the data integrity attack by using the FREEDM system located at North Carolina State University, the detailed simulation setup could be found in [1]. The comparison between the normal schedule and the attacked one is given in Fig.3, where the green bars denote the schedule in the normal condition, and the red bars denote the schedule under data integrity attack.

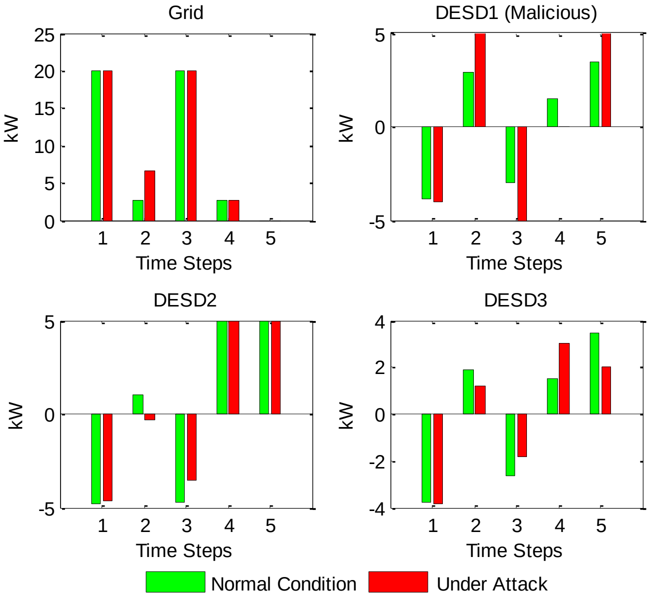


Figure 3: The impact of $\Delta\hat{P}_M^k(t)$ on the power generation

The economic impacts of the attack on three DESDs and the total electricity bill are calculated as **Error! Reference source not found.** and **Error! Reference source not found.**, respectively. The results are summarized in Table I. As we can see, after the attack, the total electricity bill increases as the attacked schedule is not the optimal result. In the meantime, only the attacker DESD 1 gains more economic benefit from the attack, while the other two DESDs make less money compared to the normal condition.

TABLE I. THE EXTRA MONEY OBTAINED BY LAUNCHING THE DATA INTEGRITY ATTACK

Benefit (cents)	Total Bill	DESD 1	DESD 2	DESD 3
Normal	187.02	26.08	38.56	22.35
Attacked	208.55	34.06	35.98	17.03
Difference	21.53	7.98	-2.58	-5.32
Impact (%)	+11%	+30%	-6%	-23.6%

4.2 Reputation-based neighborhood-watch algorithm

We proposed a reputation-based neighborhood-watch algorithm in which every node could monitor the correctness of the shared information from its neighbor and counteract the attacks [4]. Similar concept is available in our previous work [5], [6]. The objective of the proposed resilient control mechanism is twofold: 1) to detect the presence of any manipulated information; including λ_i^k and ΔP_i^k; 2) to recover the optimal energy schedule from the malicious impact of the attack. The framework of the reputation-based neighborhood-watch algorithm is shown as Fig.4. In the following the details of each steps are described.

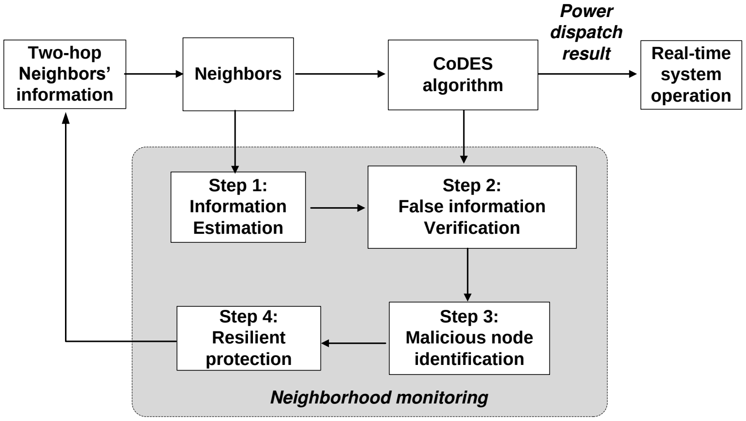

Two-hop Neighbors' information
Neighbors
CoDES algorithm
Power dispatch result
Real-time system operation
Step 1: Information Estimation
Step 2: False information Verification
Step 3: Malicious node identification
Step 4: Resilient protection
Neighborhood monitoring

Figure 4: The framework of distributed neighborhood-watch algorithm

- Step 1: Based on two-hop neighbors' shared information in iteration$k$, each node estimates its neighbors' shared information in next iteration $k+1$, where the estimated information includes λ_i^k and ΔP_i^k.
- Step 2: Each node detects the false information from its neighbors by comparing the estimated value with the actual received value.
- Step 3: Adjust the credibility of neighbors via the Local Reputation Index. Identify a malicious bus if the corresponding reputation drops below a threshold.
- Step 4: Information from the malicious bus is discarded by neighbors, and the other normal buses use estimated information to continue the iterative process

4.3 Algorithm implementation in DGI 2.0 framework

The CoDES algorithm has been implemented in a Linux PC environment running Ubuntu 16.04 LTS. The program is able to calculate the 24-hour charging/discharging schedule of DESDs of the 4-node FREEDM system as shown in Fig.5.

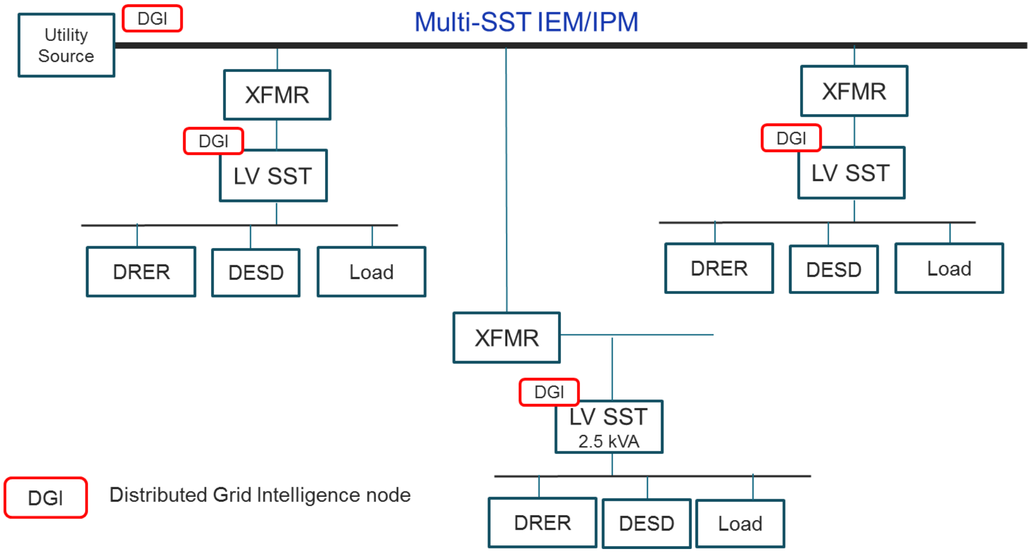
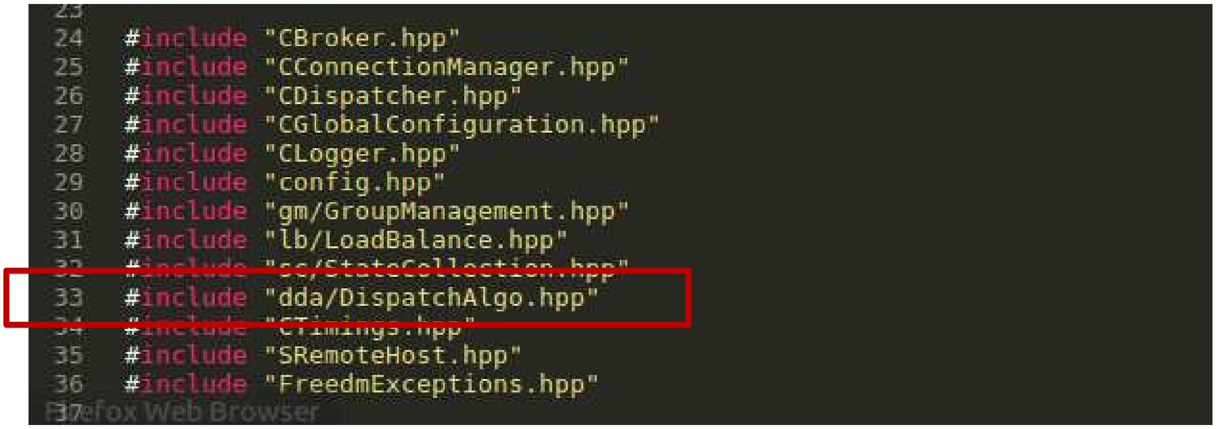


Figure 5: 4-node FREEDM system

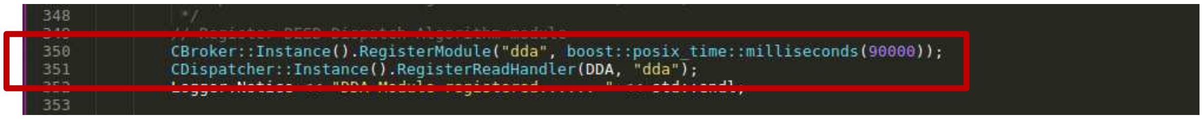
4.3.1 Configuration setup

The CoDES module is registered in the DGI main function (PosixMain.cpp) by adding "dda/DispatchAlgo.hpp" to the PosixMain.cpp.

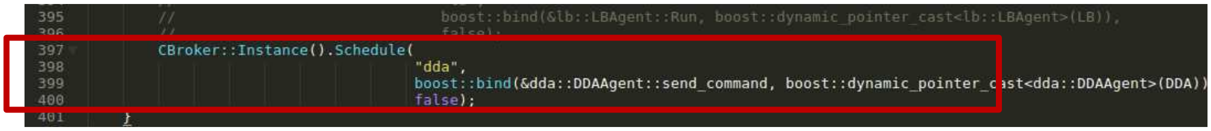


```
#include "CBroker.hpp"
#include "CConnectionManager.hpp"
#include "CDispatcher.hpp"
#include "CGlobalConfiguration.hpp"
#include "CLogger.hpp"
#include "config.hpp"
#include "gm/GroupManagement.hpp"
#include "lb/LoadBalance.hpp"
#include "sc/StateCollection.hpp"
#include "dda/DispatchAlgo.hpp"
#include "CTimings.hpp"
#include "SRemoteHost.hpp"
#include "FreedmExceptions.hpp"
```

The dda module has already been registered with 90 seconds phase time. Detailed code are shown in the source code between line 326 to line 407.



```
         */
        // Register DESD Dispatch Algorithm module
        CBroker::Instance().RegisterModule("dda", boost::posix_time::milliseconds(90000));
        CDispatcher::Instance().RegisterReadHandler(DDA, "dda");
        Logger.Notice << "DDA Module registered...... " << std::endl;
```

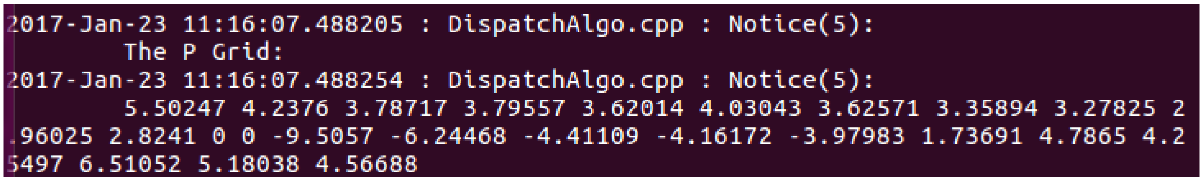


```
    //                          boost::bind(&lb::LBAgent::Run, boost::dynamic_pointer_cast<lb::LBAgent>(LB)),
    //                          false);
    CBroker::Instance().Schedule(
                                "dda",
                                boost::bind(&dda::DDAAgent::send_command, boost::dynamic_pointer_cast<dda::DDAAgent>(DDA))
                                false);
}
```

4.3.2 Execution result example

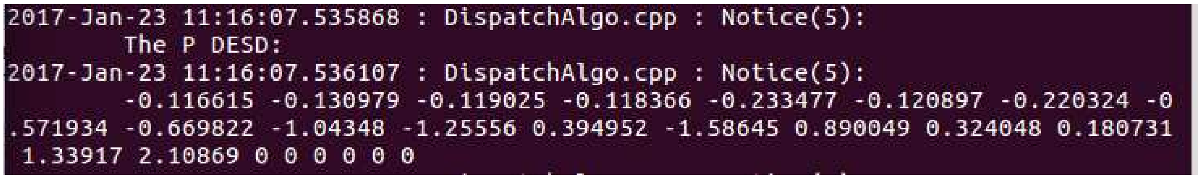
By executing (./ProsixBroker) the executable file one by one, the CoDES algorithm will run and each terminal (represent each DGI node) session will print out algorithm information as the algorithm executes.

Grid Power (output as a 24X1 array):



```
2017-Jan-23 11:16:07.488205 : DispatchAlgo.cpp : Notice(5):
        The P Grid:
2017-Jan-23 11:16:07.488254 : DispatchAlgo.cpp : Notice(5):
        5.50247 4.2376 3.78717 3.79557 3.62014 4.03043 3.62571 3.35894 3.27825 2
.96025 2.8241 0 0 -9.5057 -6.24468 -4.41109 -4.16172 -3.97983 1.73691 4.7865 4.2
5497 6.51052 5.18038 4.56688
```

DESD schedule (output as a 24X1 array):



```
2017-Jan-23 11:16:07.535868 : DispatchAlgo.cpp : Notice(5):
        The P DESD:
2017-Jan-23 11:16:07.536107 : DispatchAlgo.cpp : Notice(5):
        -0.116615 -0.130979 -0.119025 -0.118366 -0.233477 -0.120897 -0.220324 -0
.571934 -0.669822 -1.04348 -1.25556 0.394952 -1.58645 0.890049 0.324048 0.180731
 1.33917 2.10869 0 0 0 0 0 0
```

5. Other Relevant Work Being Conducted Within and Outside of the ERC

NSF program: Secure Algorithms for Cyber-Physical Systems under Award Number 1505610.

6. Milestones and Deliverables

Q3 (9/30/2016) –

- In process of implementing the neighborhood-watch algorithm in DGI platform to secure the energy scheduling algorithm.

Q4 (12/31/2016) –

- A working implementation of the neighborhood-watch algorithm in DGI platform, in process of exploring possible cyber/physical attacks on energy scheduling and real-time operation.

Q1 (3/31/2017) –

- Build the resilient cyber-physical control framework, develop the collaborative cyber/physical countermeasures against potential attacks.

Q2 (6/30/2017) –

- The implementation of the resilient cyber-physical control framework in DGI platform.

Deliverable for SV (04/2017):

- A working implementation of the resilient cyber-physical control in DGI platform
- Related publications and reports.

Final Deliverable (08/2017):

- System level demonstration of the cyber/physical attacks and the corresponding countermeasures.
- Related publications and reports.

7. Plans for Next Five Years

- Consider collude misbehaving nodes in the system
- Adapt the detecting threshold considering the effect of communication noise and disturbance
- Analyze the optimal number of hops required to exchange information under different attack scenarios.

8. Member Company Benefits

By demonstrating the attacks and the resilient operation of the CoDES algorithm, the member company will be able to see the potential challenges of the distributed technologies and a promising solution to deal with the cyber attacks. This secure technology could be used for other similar applications.

9. References

[1] N. Rahbari-asr, Y. Zhang, and M.-Y. Chow, "Cooperative Distributed Scheduling for Storage Devices in Microgrids using Dynamic KKT Multipliers and Consensus Networks," *IEEE Power and Energy Society General Meeting*, pp. 1–5, 2015.

[2] N. Rahbari-Asr, Y. Zhang, and M.-Y. Chow, "Consensus-based distributed scheduling for cooperative operation of distributed energy resources and storage devices in smart grids," *IET Generation, Transmission & Distribution*, vol. 10, no. 5, pp. 1268–1277, 2016.

[3] J. Duan and M.-Y. Chow, "Data Integrity Attack on Consensus-based Distributed Energy Management Algorithm," in *IEEE Power and Energy Society General Meeting*, 2017, pp. 1–5 (submitted).

[4] J. Duan, S. Member, W. Zeng, and M. Chow, "Resilient Cooperative Distributed Energy Scheduling against Data Integrity Attacks," *2016 - 42th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pp. 4941–4946, 2016.

[5] W. Zeng and M.-Y. Chow, "Resilient distributed control in the presence of misbehaving agents in networked control systems.," *IEEE Transactions on cybernetics*, vol. 44, no. 11, pp. 2038–49, 2014.

[6] J. Duan, W. Zeng, and M.-Y. Chow, "Resilient Distributed DC Optimal Power Flow against Data Integrity Attacks," *IEEE Transactions on Smart Grid*, vol. 99, no. 99, pp. 1–10 (in press), 2016.