Multilayer Distributed Control over 5G Networks: Challenges and Security Threats

Nils Vreman
Department of Automatic Control,
Lund University
nils.vreman@control.lth.se

Martina Maggio
Department of Automatic Control,
Lund University
martina.maggio@control.lth.se

ABSTRACT

Modern control algorithms are frequently implemented in a distributed and decentralized way. Multiple fog devices communicate using paradigms borrowed from the Internet-of-Things. These control infrastructures are vulnerable to security threats. This position paper describes these threats and sketches future research on how to mitigate the security concerns that a modern distributed control infrastructure poses.

CCS CONCEPTS

• **Security and privacy**; • **Computer systems organization** → **Embedded and cyber-physical systems**; Fault-tolerant network topologies;

KEYWORDS

Distributed Control, 5G, Security, Latency-based Attacks

ACM Reference Format:
Nils Vreman and Martina Maggio. 2019. Multilayer Distributed Control over 5G Networks: Challenges and Security Threats. In *Workshop on Fog Computing and the IoT (IoT-Fog '19), April 15–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3313150.3313223

1 INTRODUCTION

In the past few years, there has been a trend towards distributed and decentralized systems. This is true even in domains like control. Historically, physical plants were designed with a given topology and interconnection. The elements of these plants were potentially complex objects, but their dynamics were well known. Currently, this is not true anymore. With the advent of the Internet of Things world, objects move around, interacting with one another.

Recently, [20] showed that a control plant can be controlled with code that is migrated from the proximity of the plant to two datacenters, located in various places and with different distances from the plant. This creates interesting opportunities for control, since much more can be done to regulate complex systems when data is shared and code is migrated. Considering the case where the plant has limited processing power, these results become even more interesting. 5G is here an enabling technology. It targets high data rate, reduced latency, energy saving, cost reduction, higher system capacity, and massive device connectivity. This creates new opportunities for control. For instance, with the reduced latency, time-critical control signals can be computed in a datacenter and reliably sent back to where it should be applied — something that was inconceivable a few years ago.

These new opportunities also come with significant new challenges, for example related to the security of these new systems. This position paper presents an overview of our research agenda in the domain of security for distributed and decentralized control systems. In particular, we are investigating problems with three distinct characteristics: (i) *different optimization levels*, (ii) *different security threats*, and (iii) *different latencies*. We argue that with the emergence of fog computing and Internet of Things, these characteristics can be found in many different problems. Here, we use a practical example, in a relevant domain, to motivate our proposal and goal.

1.1 Motivating Example

This section introduces our motivating example. A set $\mathcal{T} = \{t_1, \ldots, t_n\}$ of n trucks is circulating in a given area. The area is covered by a set $\mathcal{A} = \{a_1, \ldots, a_p\}$ of p 5G antennas, potentially with overlapping coverage. We assume that the antennas can communicate with a set $\mathcal{D} = \{d_1, \ldots, d_m\}$ of m data centers.

Figure 1 illustrates one possible scenario for our motivating example. The figure shows $m = 1$ data centre, $n = 7$ trucks, and $p = 4$ antennas with their respective coverage areas. As in a realistic case, some areas may not be covered by any signal. In the figure, truck t_6 and t_7 for example can be found in areas without coverage. On the contrary, truck t_2 is in a location that is covered by both a_1 and a_3.

We consider the following optimization problem: each truck has a given origin and destination, but no fixed path. We would like to optimize the path of all trucks in terms of fuel consumption in the presence of disturbances that can span from wind to the driver's need for a break. This means that we would like, for example, to create a platoon of vehicles when possible, reducing drag for trucks subsequent to the first one. In the figure, trucks t_2, t_3 and t_4 are proceeding together, therefore t_2 and t_3 can take advantage of t_4 and reduce their fuel consumption.

For this problem, we can identify three different *levels* for solving the route planning problem: (i) the *local* level (L), (ii) the *proximal* level (P), and (iii) the *global* level (G).

At the local level (L), each truck can solve the route planning problem with the objective to minimize its own fuel consumption. The latency is (very often) negligible. Provided that the choice of hardware used for the control solution is appropriate, it is most

IoT-Fog '19, April 15–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6698-4/19/04…$15.00
https://doi.org/10.1145/3313150.3313223

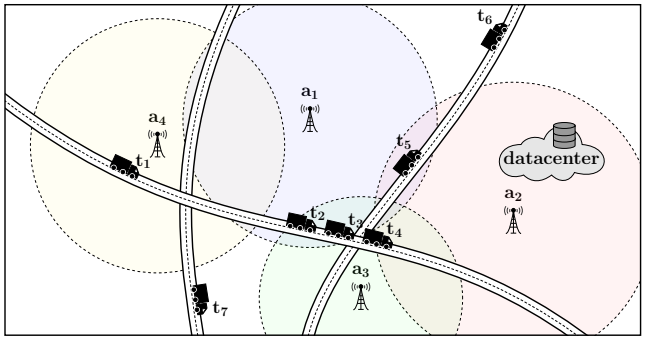

Figure 1: Motivating Example. A set of trucks is driving on some roads with source and destination points. The shipping company wants to optimize fuel consumption of all the trucks despite disturbances. Each truck can run a route control algorithm locally. If they are in the coverage areas of some 5G antennas, the trucks are connected to the closest antennas, where some computation can take place, taking advantage of additional information known at the proximal level. Finally, a data center with even more information can run a centralized optimization algorithm, at the global level.

likely possible to determine a Local Worst Case Execution Time ($WCET_L$) that meets control design requirements. The control solution is as secure as the hardware and software used to obtain it. Also, the lack of connectivity forces some trucks, like t_6 and t_7 to operate at the local level. We discuss the security threats at the local level in Section 2.1.

At the proximal level (P), each truck communicates with the antenna that is covering its location. The optimization problem now includes the routes of multiple trucks, which means that other features of the problem can be exploited, for example vehicle platooning. If it is possible to lower the sum of the fuel consumptions for all the trucks that belong to the coverage area, executing the control code at the antenna level is then beneficial. The Ultra-Reliable and Low-Latency Communication (URLLC) 5G network [19] allows for fast and reliable communication. However, when compared to the local version, the network latency increases, as the measurements have to be sent to the antenna and the control signal has to be retransmitted to the trucks. Also, the Proximal Worst Case Execution Time ($WCET_P$) is a function of the number of trucks in the coverage area and is different than in the local case. If many trucks are transiting in the same area, $WCET_P$ is most likely higher than $WCET_L$. However, in principle $WCET_P$ could also be lower than $WCET_L$, because the code might be executed on more powerful hardware. The additional communication and computation also provides additional opportunity for security threats. These are described in Section 2.2 for the antenna level and Section 2.4 for the communication channel.

Finally, at the global level (G), all the antennas communicate with a data center. The optimization problem now include a wider area and can return a better solution. However, the communication latency increases as the truck should communicate with an antenna, the antenna with the data center, and signals have to be sent back to the truck. If t_1 is willing to pay the additional latency cost, for example, it could discover the presence of t_2, t_3, and t_4 and speed up to join the platoon as soon as possible. The Global Worst Case Execution Time ($WCET_G$) also depends on the number of trucks and similar considerations as the ones done for the proximal level hold. The computation in the data center can also incur additional security problems. These are discussed in Section 2.3 for the data center level and Section 2.4 for the communication channel.

1.2 Problem Relevance

We have used a representative example in a relevant domain. A shipping company that would like to optimize the route of its trucks online, will benefit from running the controller at the highest possible level (G), but would not want to be expose to severe security threats. When distribution is becoming more and more common, this problem is not confined to our own motivating example, but emerge in many other domains. Additional knowledge is always beneficial when trying to optimize the behavior of a distributed system, but it comes at the cost of communication and centralized computation. We propose to precisely quantify this cost both in terms of control solution and in terms of security risks and threats, to be able to take informed decisions.

In Section 2 we discuss some of the security threats that the system should defend itself against. In Section 3 we sketch some of the characteristics of our proposed solution and discuss our future research agenda.

2 SECURITY THREATS

This section introduces and classifies known threats that any computing infrastructure similar to the one described in our motivating example is vulnerable to. We analyze threats to each infrastructural element separately, classifying them into four categories. The first three categories are *local*, *proximal*, and *global*. The last category includes threats to the *communication* layer that allows any of these level to communicate with the others. We introduce attacks targeting each of these levels in the following subsections.

2.1 Local level (L)

The local level is vulnerable to *direct* attacks, in addition to *remote* attacks. For example, a truck is vulnerable to *physical attackers*, injecting malware through USB-connections or similar outlets. This malware takes control of the navigation system and can change the algorithm used for the computation of routes.

Another threat at the local level is the tampering of sensors and the corruption of actuator signals. These two can cause great damage for the system. For instance, *false data injection attacks* [12] alter the value of certain sensor measurements. The controller then executes using erroneous data. In our motivating example, presented in Section 1.1, the attack could alter the data read by a velocity sensor, returning a smaller velocity than the actual measurement. Over time, this results in a significant speed increase. False data injection attacks can be mitigated by introducing identification methods and adaptive anomaly detectors [8, 11, 28].

Actuator attacks are not as common as sensor attacks, as they are more difficult to perform. Actuator attacks alter the control signals calculated by the controllers to drive the system away from its desired state and make it follow a malicious trajectory. An example

of an actuator attack is the *zero-dynamics actuator attacks* [23]. Such attacks are incredibly hard to perform, due to the attack requiring complete knowledge of the plant dynamics. The attack is based on adding an attack signal to the actuator value computed by the controller such that the output signal is zero and the internal plant state is diverging or following an unwanted trajectory [2, 17]. In our example, if a controller is set to follow a given velocity profile for the trucks and the output of the system is the error, the attacker could achieve the perception of a zero output error while changing the actual speed of the truck.

2.2 Proximal level (P)

Computing at the proximal level introduces threats, arising from the presence of communication. Latency requirements are one of the most critical aspects when control code is executed remotely.

Latency-based attacks are a major concern for remote controller invocations. Here, we use the term latency-based attacks as a general concept to denote attacks whose main purpose is to significantly increase the latency between the actuation layer and the code execution layer. Such attacks include (but are not limited to) *botnets* [3, 18]. A botnet consists of multiple infected devices, synchronously sending requests to a node in a network, with the purpose of overloading it (ergo; increasing the service rate and therefore the latency). Renting a botnet is now very cheap, due to the sheer amount of infected devices. Usually botnets are rented to cause *distributed denial of service* (DDoS) [5] attacks, i.e., to make some components unavailable in a complex infrastructure.

Proximal level components like antennas are also vulnerable to physical attacks. Their availability and accessibility is crucial for the system's correct behavior.

2.3 Global level (G)

Cloud services can be used as a centralized control unit – e.g., in the trucks routing problem, it can collect the status of many different trucks and optimize their routes. However, cloud datacenters are also subject to attacks and security threats. Similarly to the proximal level, the global level is vulnerable to latency-based attacks. A latency increase can render the advantages of using the cloud service (instead of another level in the hierarchy) insignificant, or even damaging.

Furthermore, cloud computing provides us with the illusion of unlimited resources, at the cost of increased latency. To achieve this aim, usually applications running in the cloud are partitioned into virtual machines. These virtual machines are subjected to multiple attacks, such as *data inference* attacks [9, 16], and *rowhammer* attacks [10, 22].

Data inference attacks get access to data stored in the system and gain information about its functioning. From a corporate viewpoint, this could result in loss of income due to corporate secrets or system information getting revealed. If a truck is carrying an important load, a data inference attack can reveal its position (together with the position of the other trucks) and allow a robber to steal the load when no other truck is around. An example of data inference attacks is *memory disclosure attacks*. They abuse leakage from the memory deduplication to infer private information [6]. In a datacenter, computational efficiency and power are essential. To reduce physical memory needs, deduplication is utilized, which means that identical physical pages are stored using the same memory address. This allows these types of attacks to succeed. Some mitigations strategies have been proposed [4, 21], although there is no clear consensus on whether they can completely eliminate the threat.

On the contrary, rowhammer attacks are used to flip bits on a DRAM row without accessing the data. With a rowhammer attack, the attacker does not only read the data, but also compromises it. In the trucks example, the position of the trucks can be altered, forcing the optimization algorithm to compute unreasonable control signals. A rowhammer attack can force the truck to an isolated position, where the robber mentioned above has the advantage. Rowhammer attacks have recently attracted great attention due to their versatility and potential for devastating effects as well as the absence of functional protection schemes against them. The proposed (and applied) short-term solution has been to increase the DRAM refresh rate, ergo; reducing the probability of bit flips being induced [15]. Increasing the refresh rate does however also increase the power consumption as well as decrease the performance. The authors do also discuss future prevention systems. However, the proposed solution is not easily implementable. In [7], the authors present a new rowhammer attack that is unaffected by the current state-of-the-art methods.

2.4 Communication (L↔P, P↔G)

Each level in the hierarchy has its security flaws. Moreover, the connection between the levels introduce additional security threats. We refer both to the communication between the *local* and *proximal* levels and the *proximal* and *global* levels, since the communicative channels are exposed to the same threats.

Communication channels are the target of attacks like *Man-in-the-Middle* (MITM). In MITM attacks, an attacker monitors a communication channel, listens and infers the messages being sent on the channel, and alters them before passing them on to the receiver. An encrypted and authenticated connection limits the power of these attacks.

3 PROPOSED RESEARCH

We would like to adapt the security level and the overall control latency simultaneously. The encryption, decryption, and control signal computation time is different depending on which level the operation is performed at. We define the set of encryption times $\Delta_e = \{\Delta_e^{\mathsf{L}}, \Delta_e^{\mathsf{P}}, \Delta_e^{\mathsf{G}}\}$, where the superscript represents the level. Correspondingly, we define the set of decryption times Δ_d and control signal computation times Δ_c.

Challenge 1: A challenging aspect of this research is the formal definition of what *security level* means. Intuitively, a bigger shared key is harder to crack compared to a smaller one. However, properly determining how secure a system is based on the properties of its encryption choice is still an open research topic.

Figure 2 shows an example of communication where the control code is executed at the global level G. Time advances from left to

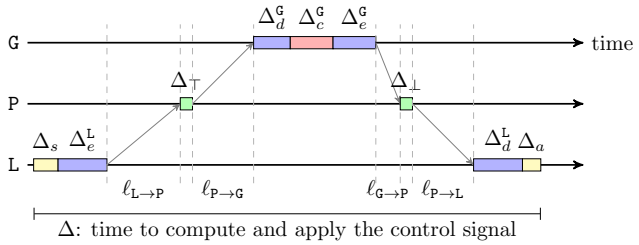

Figure 2: Example of sensing to actuating cycle.

right in the figure. At the local level L, sensors are activated, resulting in a sensing delay Δ_s. A message is prepared and encrypted, with an additional delay Δ_e^{L}. The message is then sent upwards in the hierarchy, reaching the antenna after $\ell_{\mathsf{L}\to\mathsf{P}}$ time units. The antenna forwards the message to the global level with a processing time $\Delta_\top$. At the global level, the message is received after $\ell_{\mathsf{P}\to\mathsf{G}}$ time units. Furthermore, the message is decrypted with a delay Δ_d^{G}, a control value is computed in Δ_c^{G} time units, and then the control signal is encrypted in Δ_e^{G} time units and forwarded to the antenna. The antenna receives the encrypted control signal after $\ell_{\mathsf{G}\to\mathsf{P}}$ time units and forwards it to the local level with an execution time of $\Delta_\bot$ and a latency of $\ell_{\mathsf{P}\to\mathsf{L}}$. Finally, at the local level the message is decrypted, spending Δ_d^{L} time. It then takes Δ_a time to properly actuate the control signal.

The overall time to compute and apply the control signal, Δ, is the sum of all these quantities. When only local control is necessary, Δ is often negligible and the control engineer can synthesize a control system disregarding the delay. In our case, the engineer can optimize Δ_s and Δ_a on the local platform. Also, the forwarding time $\Delta_\top$ and $\Delta_\bot$ can be optimized. The control signal computation time Δ_c^{G} depends on the amount of information that is received and merged at the global level. We assume that an upper bound is provided.

Challenge 2: A research challenge is how to determine latency bounds. In our proposed solution, we use knowledge of both computational and network delays to optimize the choice of encryption algorithm. We will investigate how to bound computation time for encryption and decryption for the different levels in the hierarchy. We also need to have reliable estimates of control signal computation times at different levels (WCET_{L}, WCET_{P}, WCET_{G}), considering that the number of nodes providing information may vary.

Our goal is to have the highest level of security possible with the control computation being performed at a given level (L, P, or G). We propose to act on the encryption and decryption times Δ_e and Δ_d, adapting at the same time both the security level and the overall latency. To achieve this goal we can: (i) use different encryption algorithms, (ii) modify the key (or the "secret") size using the same encryption algorithm. For each of the alternatives, we expect corresponding upper bounds to be given. We are then able to make the most secure choice, controlling at the desired level. A lower level is selected when no choice is able to guarantee a timely computation. We assume that the controller at the local level is designed to fulfill the latency requirements. There is some ongoing work on this topic for Ethernet control that we plan to leverage on [1, 14].

Challenge 3: Alongside determining upper bounds, we will also investigate which encryption algorithms are *suitable* for our purposes. We need algorithms that are inherently adaptive, i.e., where changing parameters (like the key size) results in different computation times and different security levels. We will also investigate the limitations of these algorithms. For example, the security level provided by encryption is likely not a continuous function, i.e., a key smaller than a given size might be equivalent to no security at all for the system.

Furthermore, we propose to build on ongoing research on anomaly detection [13, 24, 26, 27]. Anomaly detectors are usually able to identify if a computing infrastructure is compromised, under some assumptions. For each of our levels, we should verify which assumptions hold and then try to detect attacks. As a response to the detection, we can then migrate the control code when necessary to avoid security bridges. In addition to classic anomaly detection techniques, we can take advantage of the distributed setting to reveal attacks that are targeting only certain nodes in the infrastructure [25]. Using information sent by other nodes we can infer faults and miscommunication. Going back to our motivating example, given a platoon of trucks, if all but one of them report a decrease in speed, the one vehicle with a divergent velocity might have been compromised.

The main purpose of our investigation is to obtain a scheme that enables *informed code migration*, for both security issues and latency requirements.

4 CONCLUSION

This position paper describes our future research in the field of computer security for distributed control systems over 5G networks. We provided a motivating example of why this research is timely and relevant. We believe that the advent of 5G opens possibilities to develop a new class of distributed control systems, that take advantage of both the local information and aggregated data. However, it is also evermore important to guarantee the security of such complex control systems.

We provided a brief survey of attacks that target this special class of systems and existing countermeasures. However, we believe that combining latency requirements and security concerns is still an open problem, that could enable much better control solutions. Finally, we highlighted research challenges that we would like to address in the future and described our planned research path.

Acknowledgements: This work was supported by the ELLIIT Strategic Research Area and by the Nordforsk Nordic Hub on Industrial IoT (HI2OT).

REFERENCES

[1] Amir Aminifar, Petru Eles, and Zebo Peng. 2018. Optimization of Message Encryption for Real-Time Applications in Embedded Systems. *IEEE Trans. Computers* 67, 5 (2018), 748–754. https://doi.org/10.1109/TC.2017.2778728

[2] J. Back, J. Kim, C. Lee, G. Park, and H. Shim. 2017. Enhancement of security against zero dynamics attack via generalized hold. In *Conference on Decision and Control (CDC)*. 1350–1355. https://doi.org/10.1109/CDC.2017.8263842

[3] E. Bertino and N. Islam. 2017. Botnets and Internet of Things Security. *Computer* 50, 2 (Feb 2017), 76–79. https://doi.org/10.1109/MC.2017.62

[4] David Bigelow, Thomas Hobson, Robert Rudd, William Streilein, and Hamed Okhravi. 2015. Timely Rerandomization for Mitigating Memory Disclosures. In *Conference on Computer and Communications Security (CCS)*. 268–279. https://doi.org/10.1145/2810103.2813691

[5] Christos Douligeris and Aikaterini Mitrokotsa. 2004. DDoS Attacks and Defense Mechanisms: Classification and State-of-the-art. *Computer Networks* 44, 5 (April 2004), 643–666. https://doi.org/10.1016/j.comnet.2003.10.003

[6] Daniel Gruss, David Bidner, and Stefan Mangard. 2015. Practical Memory Deduplication Attacks inźSandboxed Javascript. In *Proceedings, Part I, of the 20th European Symposium on Computer Security – ESORICS 2015 - Volume 9326*. 108–122. https://doi.org/10.1007/978-3-319-24174-6_6

[7] Daniel Gruss, Moritz Lipp, Michael Schwarz, Daniel Genkin, Jonas Juffinger, Sioli O'Connell, Wolfgang Schoechl, and Yuval Yarom. 2017. Another Flip in the Wall of Rowhammer Defenses. *CoRR* (2017).

[8] J.M. Hendrickx, K.H. Johansson, R.M. Jungers, H. Sandberg, and K.C. Sou. 2014. Efficient Computations of a Security Index for False Data Attacks in Power Networks. *IEEE Trans. Automat. Control* 59, 12 (Dec 2014), 3194–3208. https://doi.org/10.1109/TAC.2014.2351625

[9] Jinyuan Jia, Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. 2017. AttriInfer: Inferring User Attributes in Online Social Networks Using Markov Random Fields. In *International Conference on World Wide Web (WWW)*. 1561–1569. https://doi.org/10.1145/3038912.3052695

[10] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. 2014. Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors. In *Annual International Symposium on Computer Architecuture (ISCA)*. 361–372.

[11] L. Liu, M. Esmalifalak, Q. Ding, V.A. Emesih, and Z. Han. 2014. Detecting False Data Injection Attacks on Power Grid by Sparse Optimization. *IEEE Transactions on Smart Grid* 5, 2 (March 2014), 612–621. https://doi.org/10.1109/TSG.2013.2284438

[12] Yao Liu, Peng Ning, and Michael K. Reiter. 2009. False Data Injection Attacks Against State Estimation in Electric Power Grids. In *Conference on Computer and Communications Security (CCS)*. 21–32. https://doi.org/10.1145/1653662.1653666

[13] Rouhollah Mahfouzi, Amir Aminifar, Petru Eles, Zebo Peng, and Mattias Villani. 2016. Intrusion-Damage Assessment and Mitigation in Cyber-Physical Systems for Control Applications. In *Conference on Real-Time Networks and Systems (RTNS)*. 141–150. https://doi.org/10.1145/2997465.2997478

[14] R. Mahfouzi, A. Aminifar, S. Samii, A. Rezine, P. Eles, and Z. Peng. 2018. Stability-aware integrated routing and scheduling for control applications in Ethernet networks. In *Design, Automation Test in Europe Conference (DATE)*. 682–687. https://doi.org/10.23919/DATE.2018.8342096

[15] O. Mutlu. 2017. The RowHammer problem and other issues we may face as memory becomes denser. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*. 1116–1121. https://doi.org/10.23919/DATE.2017.7927156

[16] S. Narain, T.D. Vo-Huu, K. Block, and G. Noubir. 2016. Inferring User Routes and Locations Using Zero-Permission Mobile Sensors. In *Symposium on Security and Privacy (SP)*. 397–413. https://doi.org/10.1109/SP.2016.31

[17] G. Park, H. Shim, C. Lee, Y. Eun, and K.H. Johansson. 2016. When adversary encounters uncertain cyber-physical systems: Robust zero-dynamics attack with disclosure resources. In *Conference on Decision and Control (CDC)*. 5085–5090. https://doi.org/10.1109/CDC.2016.7799047

[18] A. Sagala, R. Pardosi, A. Lumbantobing, and P. Siagian. 2016. Industrial control system security-malware botnet detection. In *International Conference on Computer, Control, Informatics and its Applications*. 125–130. https://doi.org/10.1109/IC3INA.2016.7863036

[19] Mansoor Shafi, Andreas F. Molisch, Peter J Smith, Thomas Haustein, Peiying Zhu, Prasan De Silva, Fredrik Tufvesson, Anass Benjebbour, and Gerhard Wunder. 2017. 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *IEEE Journal on Selected Areas in Communications* 35, 6 (June 2017), 1201–1221. https://doi.org/10.1109/JSAC.2017.2692307

[20] Per Skarin, William Tärneberg, Karl-Erik Årzén, and Maria Kihl. 2018. Towards Mission-Critical Control at the Edge and Over 5G. In *IEEE International Conference on Edge Computing (EDGE)*. 50–57. https://doi.org/10.1109/EDGE.2018.00014

[21] Adrian Tang, Simha Sethumadhavan, and Salvatore Stolfo. 2015. Heisenbyte: Thwarting Memory Disclosure Attacks Using Destructive Code Reads. In *Conference on Computer and Communications Security (CCS)*. 256–267. https://doi.org/10.1145/2810103.2813685

[22] Andrei Tatar, Radhesh Krishnan Konoth, Elias Athanasopoulos, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. 2018. Throwhammer: Rowhammer Attacks over the Network and Defenses. In *Usenix Annual Technical Conference (ATC)*. 213–225.

[23] André Teixeira, Daniel Pérez, Henrik Sandberg, and Karl Henrik Johansson. 2012. Attack Models and Scenarios for Networked Control Systems. In *Proceedings of the 1st International Conference on High Confidence Networked Systems (HiCoNS '12)*. 55–64. https://doi.org/10.1145/2185505.2185515

[24] A. Teixeira, I. Shames, H. Sandberg, and K.H. Johansson. 2012. Revealing stealthy attacks in control systems. In *Allerton Conference on Communication, Control, and Computing (Allerton)*. 1806–1813. https://doi.org/10.1109/Allerton.2012.6483441

[25] A. Teixeira, I. Shames, H. Sandberg, and K.H. Johansson. 2014. Distributed Fault Detection and Isolation Resilient to Network Model Uncertainties. *IEEE Transactions on Cybernetics* 44, 11 (Nov 2014), 2024–2037. https://doi.org/10.1109/TCYB.2014.2350335

[26] Nikola Trcka, Mark Moulin, Shaunak Bopardikar, and Alberto Speranzon. 2014. A Formal Verification Approach to Revealing Stealth Attacks on Networked Control Systems. In *International Conference on High Confidence Networked Systems (HiCoNS)*. 67–76. https://doi.org/10.1145/2566468.2566484

[27] David I. Urbina, Jairo A. Giraldo, Alvaro A. Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. 2016. Limiting the Impact of Stealthy Attacks on Industrial Control Systems. In *Conference on Computer and Communications Security (CCS)*. 1092–1105. https://doi.org/10.1145/2976749.2978388

[28] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao. 2014. On False Data-Injection Attacks against Power System State Estimation: Modeling and Countermeasures. *IEEE Transactions on Parallel and Distributed Systems* 25, 3 (March 2014), 717–729. https://doi.org/10.1109/TPDS.2013.92