Securing Industrial Control Systems Using Physical Device Fingerprinting

Tsion Yimer
ECE Department
Morgan State University
Baltimore, Maryland
tsyim1@morgan.edu

Md Tanvir Arafin
ECE Department
Morgan State University
Baltimore, Maryland
mdtanvir.arafin@morgan.edu

Kevin Kornegay
ECE Department
Morgan State University
Baltimore, Maryland
kevin.kornegay@morgan.edu

***Abstract*—The advent of the Internet-of-Things (IoT) has introduced new connectivity modalities, communication protocols, and optimized architectures to coordinate Things on a network. BACnet MS/TP is a protocol that has the potential to deliver a reliable IoT back-end for industrial systems. However, conventional security threats can severely affect trust between the nodes in the network, leading to critical infrastructure failures. Hence, we analyze the opportunities and challenges for hardware solutions in securing BACnet controllers in this work. First, we look into the security threats and develop practical attack models. Then, we demonstrate how we model clocks in the network for fingerprinting. Next, we propose a distributed security monitor for deployment across the BACnet MS/TP nodes. We also illustrate how clock fingerprinting data can enable the security monitor to prevent intrusion and tampering. Finally, we experimentally verify our attacker model, attack scenarios, and the effectiveness of hardware-oriented security solutions for intrusion prevention and tampering on an industrial standard BACnet MS/TP network.**

***Index Terms*—the industrial control system, building automation system, intrusion detection, hardware oscillator, hardware-oriented security and trust**

I. Introduction

Industrial Control Systems (ICS) manage critical infrastructures ranging from power grids and environmental monitoring to commercial buildings' HVAC systems. These systems communicate over manufacturer-specific protocols, including Modbus, BACnet, and DNP3 [1]. ICS provides support for realizing dynamic and smart instances of the Internet-of-Things (IoT). However, the security of the IoT devices, network, control, and automation systems on the back-end, in general, are subject to vulnerabilities linked to new and legacy devices joining the networks [2]–[4]. Consequently, requiring a holistic approach that will allow the entire IoT ecosystem to protect itself with little interaction from the users. Since the IoT attack surface is increasing at a significant rate, the biggest challenge today is to develop efficient and budget-friendly techniques to ensure the security and reliability of ICS with IoT devices.

Remedies for vulnerabilities in the IoT ecosystem can be software-only solutions. However, bypassing software root-of-trust has proven to be a straightforward task. Thus, hardware-only and hardware-software solutions for IoT devices and systems security have become a popular paradigm for computer security research. Since physical properties such as device geometry, design, and fabrication signatures of a system are nearly unclonable, these physical attributes can be useful in providing novel solutions for security and trust in a densely connected IoT network.

In this work, we investigate how crystal oscillators' hardware imperfections can be useful in designing a tampering and intrusion detection system in an ICS setting. In [5], groundbreaking work on physical device fingerprinting using hardware oscillators was to show the promise of using hardware roots-of-trust in system security. This work excited the security research community; however, applications have not taken full advantage of this work. A critical implementation of Kohno's work was discovered by Murdoch [6] by applying oscillator drift to identify computers and their operating environment in a TOR network.

One of the critical reasons for the lack of application of Kohno's work in IoT is the sheer size of the IoT network. As more and more computers are connected to an extensive network, differentiating one clock from others becomes difficult due to the noise and limited resolution of the measurement system. However, for smaller networks, Kohno et al. 's' findings are significant because the system consists of a heterogeneous mixture of a low number of oscillators. Thus, for smaller IoT systems such as industrial control subsystems, the application of clock fingerprinting can serve as a powerful security solution.

Industrial control systems are a critical part of the Internet of Things. In this work, we focus on a specific subset of the industrial control systems for building automation called the Building Automation System (BAS). Security issues associated with the current practices of *smart* industrial control systems include (a) lack of tamper detection on automation level controllers; (b) adversarial physical access to the deployed controllers and sensors; (c) unauthorized access at higher levels (such as at the management layer); (d) lack of automated network traffic monitoring; and (e) common cyber-attacks to the controllers and sensors in the network. Smart buildings and industrial systems that adopt control systems for intelligent decision making are susceptible to malicious adversaries without effective and reliable security. Therefore, this work demonstrates the use of hardware-security techniques for protecting BAS. Overall, this paper makes the

following contributions:

- Vulnerabilities at the automation layer are studied and analyzed;
- We present a distributed security monitoring scheme for defending BACnet MS/TP nodes from frequent physical attacks. This monitoring scheme utilizes physically unclonable fingerprints of exiting crystal oscillators in the system for intrusion detection and tampering attacks on the BAS;
- Finally, concepts are validated experimentally and presented.

II. Background

In BAS, the field controllers allow users to control the HVAC, lighting, access control, and other systems within a building. All end-devices communicate via the Building Automation Control (BACnet) protocol. BACnet is a TCP/IP based protocol standard developed by the American Society of Heating, Refrigeration, and Air-Conditioning Engineers(ASHRAE) standard [7].

BACnet devices include profiles of the sensor, actuator, building controller, gateway, or any other devices on building automation networks. The BACnet protocol uses objects and services to communicate between these devices [8]. Each object represents either physical or virtual pieces of information in the system, such as binary and analog inputs or outputs, commands, and analog values. If an adversary gains access to the building automation network, they can perform a man-in-the-middle attack by injecting false commands or spoofed messages into the network [2].

A typical building automation system architecture may consist of supervisory controllers, various field controllers, actuators, and sensors as described below:

Supervisory devices are capable of providing standalone control for smaller systems. Field controllers and supervisory devices communicate through the Field Control (FC) bus [8].

Field controllers use inputs and outputs to monitor and control devices. The Sensor Actuator (SA) communication bus establishes communication between field controllers and networked sensors.

Sensors and actuators layer allows for monitoring and control of things such as humidity or temperature sensors, actuators, or relays.

The experimental system architecture includes high-performance workstations connected to a TCP/IP network, and network automation supervisory and control system connected using the BACnet protocol, as shown in Figure 1. The difference between a control system and a building automation system is that a control system describes the physical aspects. In contrast, a building automation system describes the control and interface elements of a system [8]. Thus a building automation system is not a control system but a layer on top of the control system. Field controllers are normalized and integrated into the supervisory layer required and responsible for the translation from a field-level network to an IT-based network. A supervisory device is a building automation system in a box that provides alarming points, visualization of the management, and control.

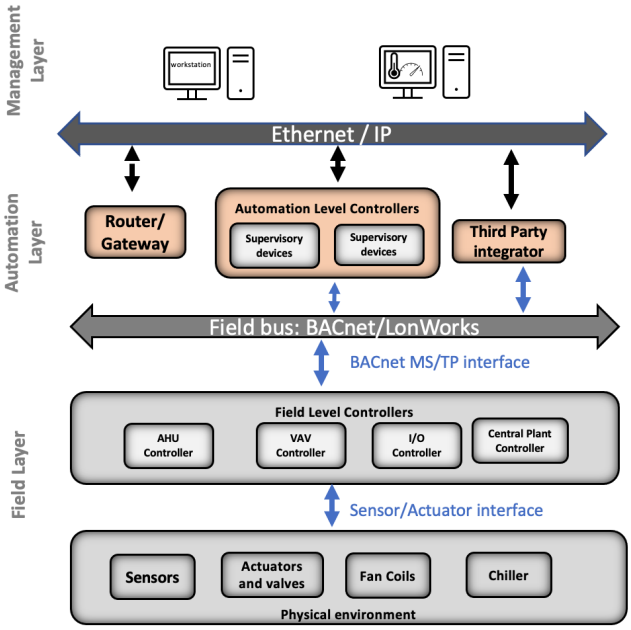

Fig. 1. Building Automation System Architecture is a typical three-tier structure.

III. Assumption and Attack Model

In this work, we consider two adversary models and multiple attack scenarios for the BAS, as shown in Figure 2. A detailed discussion on the adversary and attack model is given below.

A. Assumption

Hardware clocks consist of counters and oscillators, with the oscillators often containing crystals. A typical quartz crystal is cut and shaped for a specific frequency use [9, p.280]. However, due to the imperfections in the crystal cutting process, it has been observed that clocks may produce drifts from the actual time measured when compared to more precise clocks readings. At an atomic-level, cutting the crystals does not match; thus, each clock has its unique way of drifting in time. One can query these clocks over the internet through TCP timestamps. The basis of this idea was proposed and evaluated in [6], [10], but the hardware bases for the clocks were not considered. In this paper, we assumed that the TCP timestamp reflects the actual hardware system clock frequency. Therefore, our goal is to evaluate the automation layer devices using Kohno's remote physical device fingerprinting techniques and the TCP timestamp approach. Recently, a demonstration of how hardware oscillators can be useful in GPS spoofing detection is in [11].

B. Adversary Model

We assume that there exist two types of adversaries for this work: strong and weak adversaries. The adversaries can

have physical and/or remote access to the different layers of a BAS. Attacks on singular or multiple components at the Field level layer will mostly affect the control system decision for the devices and instruments near the attack's vicinity. The automation layer aggregates the data from the subsystem at the field level layer. Therefore, an attack on the automation layer can paralyze the intelligent decision making at the management layer. The management layer is forward and human-facing, and thus vulnerable to cyber-attacks. This work is particularly interested in preserving trust at all layers via physical device fingerprinting, ensuring reliable network communication between uncompromised devices in an ICS.

Strong Adversary Model: We consider a strong adversary has physical access to a subset of devices in any of the three layers of the BAS and can arbitrarily choose to tamper, replace, or remove any of the components. Hu et al. [12] have recently mentioned the possible existence of the attacker's chances to tamper with the industrial process data or destroy the operating rules of field devices. Critical requirements for a strong adversary include the following:

S1. The adversary has physical access to a section of the system and can tamper, replace, or remove the controllers.
S2. The adversary can swap field controllers or components in the automation layer to insert Trojans.
S3. The adversary can replace data frames or spoofed data frames in the network to seem similar to the data sent from a valid controller.

Weak Adversary Model: A semi-strong/weak adversary spoofs the ICS network using clever manipulation. A weak adversary knows the network ID of a given component in the system and is aware of BACnet MS/TP's particularities. For example, BACnet MS/TP does not allow simultaneous transmission from devices with the same network ID. We assume that a weak adversary listens and learns the communication pattern between devices in different layers and avoids simultaneous transmission, ID collisions, or simple ID-based intrusion detection techniques. Thus, the requirements for a weak adversary are:

W1. A weak adversary is an unauthorized node in the management or automation layer.
W2. It gains access to the network by spoofing the ID of a legitimate component in the system.
W3. The adversary can avoid standard software-defined intrusion detection techniques by learning the communication pattern and the identity of a legitimate component.

We do not consider the cases where a weaker attacker eavesdrops on the network communications.

C. Attack Scenarios

For this work, we have considered two attack scenarios: tampering and intrusion.

Tampering Attack: A strong adversary is capable of tampering with a physical device in the network. We assume that this attack aims to either disable the device or replace a controller with a different one.

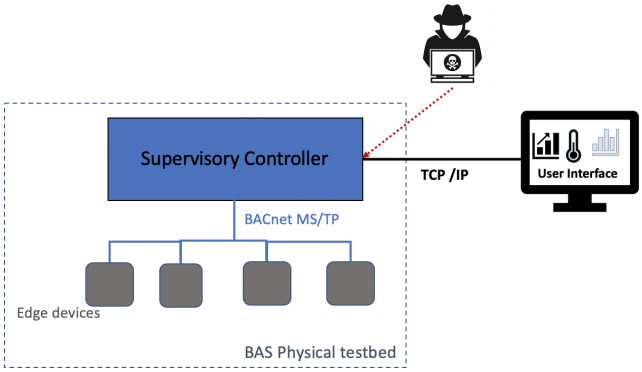

Fig. 2. Attack scenarios on the automation layer of a BAS architecture. A weak attacker will try to spoof the data from devices at the automation or the management layers. In contrast, a strong adversary will tamper or replace the automation layer devices to compromise sensors and field controllers at the field level layer.

Intrusion Attack: A weak adversary will intrude on the network and spoof the communication between two trusted nodes. We assume a somewhat smart intruder where they learn the basic communication pattern to avoid collision and simple identity-based intrusion detection.

Both attack cases are relatively common in TCP/IP based networks [13]. There are simple intrusion detection and tampering alert systems in the existing literature. A comprehensive survey of the various approaches is found in [14]. However, most techniques are software-based network-level management and detection strategies, and an attacker who fulfills W1-W3 can easily thwart most of the detection mechanisms discussed in [13], [14]. Hence, in this work, we have introduced a distributed security monitoring system that utilizes physical device signatures to detect malicious attempts in BACnet MS/TP based networks.

IV. Defense Against the Attacks

We propose a physical device fingerprint-based security monitoring system to defend against the strong and weak adversaries discussed in the previous section. We fingerprint the hardware clocks of different components in the network. In our BACnet MS/TP instances, there are fewer computers or supervisory devices connected at the automation layer. Therefore, it is easier to apply clock fingerprinting to this type of network. We discuss clock modeling in detail to develop a hardware-based security monitor.

A. Hardware Clocks

Embedded systems use real-time clocks(RTCs) for timekeeping and synchronization purposes. The common choice for BACnet MS/TP nodes on the automation layers is crystal-based RTCs used for precise synchronization between network components. These hardware oscillators are imperfect due to the accuracy in the crystal oscillator manufacturing process, and therefore deviate from ideal time. These systematic variations are observed as time and frequency offsets and frequency

drift. Therefore, at a given time t, the deviation of a clock from the ideal time can be expressed as [15]:

$$x(t) = x_0 + y_0 t + \frac{1}{2} D t^2 + \epsilon(t) \tag{1}$$

where x_0, y_0, D represents the time offset, frequency offset and frequency drift and $\epsilon(t)$ is captures non-deterministic random deviations. The frequency offsets and drifts vary for different clocks and changes in operating conditions. This is dependent on (1) design dissimilarity, (2) power supply variations, and (3) other operational and environmental factors. Based on these observations, we make the following assumptions for hardware oscillators [11]:

A1. Frequency drift and offset of a clock (measured with respect to a more precise clock) are unique for a given duration.
A2. The frequency states of a clock (measured with respect to a more precise clock) vary uniquely for different clocks.
A3. Frequency drift and offset of a clock (measured with respect to a more precise clock) of a known free-running local oscillator are predictable.

Therefore, if these assumptions hold, a given clock's frequency states will remain relatively constant for a given frame of reference. This information can help detect malicious activities by measuring the synchronization and clock properties of a trusted BACnet.

B. State-Space Model of Hardware Clocks

We use a state space mode for precisely calculating hardware clock states. The clock-state is defined by a column vector $x(t) = [x_1(t)\ y_1(t)\ D_1(t)]^T$, where, $x_1(t), y_1(t)$ and $D_1(t)$ represents the time offset state, frequency offset state, and frequency drift state. Then, the clock state follows the stochastic difference equations below [16]:

$$\frac{dx_1}{dt} = y_1 + w_1; \frac{dy_1}{dt} = D_1 + w_2; \frac{dD_1}{dt} = w_3; \tag{2}$$

here, $w_i(t)$ is the associated zero-mean white noise with spectral densities q_i. We can write the discrete-time equations for a system described by 2 as [16]:

$$\mathbf{X_n} = \mathbf{F_n X_{n-1}} + \mathbf{W_n} \tag{3}$$

$$\xi_\mathbf{n} = \mathbf{H_n X_n} + \mathbf{V_n} \tag{4}$$

here, $n = 0, 1, 2, \ldots$ corresponds to discrete-time t_n and measuring time interval $\Delta = t_n - t_{n-1}$, the $\mathbf{X_n} = [x_1, y_1, D_1]^T$ represents the state vector, $\xi_\mathbf{n}$ denotes the observation vector, and $\mathbf{F_n}$ is the state transition matrix which is computed using the following equations:

$$\mathbf{F_n} = \begin{bmatrix} 1 & \Delta & \Delta^2/2 \\ 0 & 1 & \Delta \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The process noise $\mathbf{W_n}$ is represented with covariance matrix $\mathbf{Q}$, which is given as [11]

$$\mathbf{Q} = \begin{bmatrix} q_1\Delta + q_2\frac{\Delta^3}{3} + q_3\frac{\Delta^5}{20} & q_2\frac{\Delta^2}{2} + q_3\frac{\Delta^4}{8} & q_3\frac{\Delta^3}{6} \\ q_2\frac{\Delta^2}{2} + q_3\frac{\Delta^4}{8} & q_2\Delta + q_3\frac{\Delta^3}{3} & q_3\frac{\Delta^2}{2} \\ q_3\frac{\Delta^3}{6} & q_3\frac{\Delta^2}{2} & q_3\Delta \end{bmatrix} \tag{6}$$

For calculating the clock's frequency states, one needs to define the noise covariance matrix $\mathbf{Q}$ properly. If the operating condition and the oscillator remains unchanged, and a proper noise covariance matrix is used in the calculation, one would not see any change in the clock states for the normal operating condition. This observation is the primary motivation for designing the clock fingerprinting-based security monitoring system.

C. State Estimation

We use this state-space model for hardware oscillators and employ a Kalman filter to estimate frequency states of a clock. The algorithm for this linear Kalman filter is given below [17]:

Prediction Step:

$$\mathbf{m_{n|n-1}} = \mathbf{F_n m_{n-1|n-1}} \tag{7}$$

$$\mathbf{P_{n|n-1}} = \mathbf{F_n P_{n-1|n-1} F_n^T} + \mathbf{Q} \tag{8}$$

Update Step:

$$\mathbf{K_n} = \mathbf{P_{n|n-1} H_n^T} (\mathbf{H_n P_{n|n-1} H_n^T})^{-1} \tag{9}$$

$$\mathbf{m_{n|n}} = \mathbf{m_{n|n-1}} + \mathbf{K_n}(\xi_\mathbf{n} - \mathbf{H_n m_{n|n-1}}) \tag{10}$$

$$\mathbf{P_{n|n}} = \mathbf{P_{n|n-1}} - \mathbf{K_n H_n P_{n|n-1}} \tag{11}$$

Here, the measurement matrix is givens as $\mathbf{H_n} = [1, 0, 0]$. $\mathbf{V_n}$ represents the zero-mean measurement noise with covariance $\mathbf{R} = \mathbf{0}$, $\mathbf{m_{n|n}}$, $\mathbf{P_{n|n}}$ are the Gaussian posterior mean and covariance at n^{th} time-step, and $\mathbf{K_n}$ is the Kalman gain. The clock states at n^{th} time-step is given by the components of $\mathbf{m_{n|n}}$ at that step .

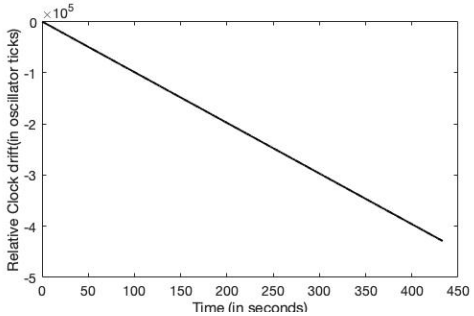

Fig. 3. Clock drift of the supervisory controller in the testbed measured with respect to the workstation clock. The drift data is reported as the difference in the values of the oscillator ticks in the timestamp data.

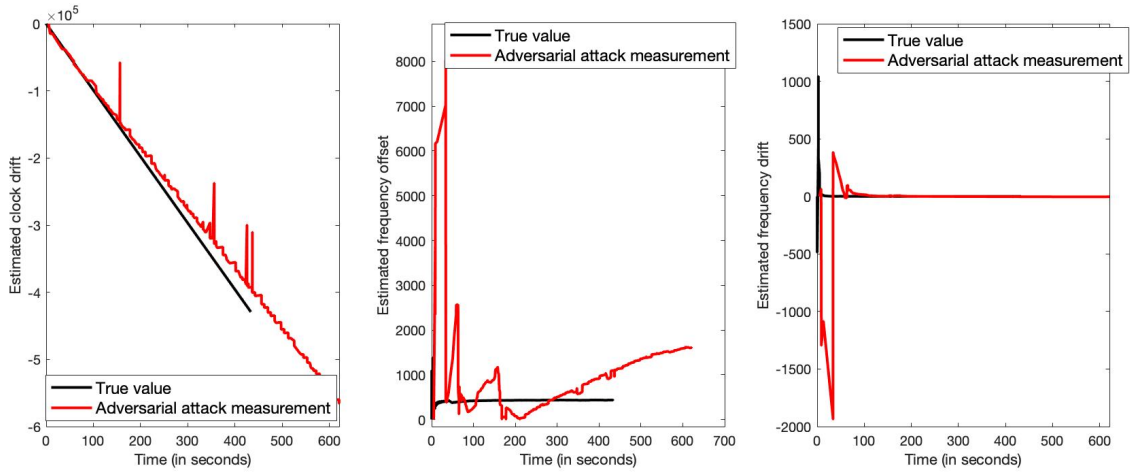

Fig. 4. Difference in clock offset drift, frequency offset and frequency drift for two clocks in the network. The frequency offset and drift are computed using the optimized noise parameters from the controller.

V. Experimental Results & Discussions

We validate the assumptions A1-A3 presented in section 4.1 using real measurements from an industrial standard building automation testbed. In this experiment, we use a supervisory controller, a building automation system in a box that provides alarming points, and a visualization of the management and control of the field devices.

The clock properties of the supervisory controller are measured with respect to a workstation for validating our assumption at A1-A3. We noticed that for noise parameters $q1 = 1e-3, q2 = 1e-9$, and $q3 = 1e-12$, a gradually increasing clock offset between the worker and the manager. This offset is shown as the true offset in Figure 3, and the steady frequency values are shown as 'True Value' in Figure 4.

A. Attack Detection

The workers in the security monitor system utilize the clock states and scrutinize any unusual behavior to raise an alert. For strong adversaries performing tampering attacks, the security monitor will observe sudden changes in clock offsets. For example, suppose the supervisory controller is replaced with an adversarial controller with a different clock, as shown in Figure 4. In that case, the monitor will experience a sudden change in the offset values of the new clock. The result will cause the worker-monitor to trigger a TAMPER alarm. Furthermore, any unexpected change in the environmental condition (temperature, vibration, *etc.*) will change the relative clock properties between a worker and a manager; thus, physical tampering would also be reported from the worker instance. Since physical environmental changes can impact the clock offset, we proposed a distributed security monitoring system to test the accuracy, false positives, and detection rate.

The weak adversaries with the capabilities mentioned in W1-W3 in section 3.1 will also find it difficult to hide their clock signature from the TCP/IP timestamps mandated by the security monitors. As shown in Figure 4, any replaced controller or the spoofed data packets will fail to demonstrate the same clock frequency and offset as the true clock. Since the worker is trained with the correct noise parameters (with respect to a given manager) that generate a stable value for clock drift, changing the oscillator and/or spoofing the TCP/IP timestamp data would create noise in the frequency offset measurement and will trigger an INTRUSION alarm.

Since the attacks are triggered based on the historical values of the clock drift, a windowed averaging techniques can be used to store the moving average values s_n of the relative clock parameters. An anomalous event can be detected when z_n crosses a predefined threshold. Here,

$$s_{i,n} = \alpha s_{i,n-1} + (1-\alpha)\ln(p(m_{i,n})) \tag{12}$$

where $p(m_{i,n})$ is the loglikelihood of an estimate, and α is the smoothing factor.

B. Designing a Distributed Security Monitoring System (DSMS) for BACnet MS/TP devices at the automation layer

Using the clock-fingerprinting technique described in the previous section and the common BACnet MS/TP network architecture described in section 2, we design a distributed security monitoring system for BACnet devices at the automation layer. This system consists of a manager-worker architecture, where every node in the automation and field-level layers has an instance of a light-weight worker system. The managers are deployed at the supervisory layer and report to the security alarm. The workers maintain noise profiles for their RTCs measured with respect to all of the manager clocks. When a node receives TCP/IP timestamp data from a manager, it computes the corresponding Gaussian posterior mean and covariance for that timestep using the Kalman filter in 4.3. The managers are deployed in the management layer, whereas the workers can be deployed at the automation and

field-level layers. The duties of the managers and workers are listed below:

Manager

- Enable the TCP/IP timestamp for all the worker nodes;
- Aggregate ALERTS from the workers and report it at the management layer;
- For new controllers joining the network, alert as an INTRUDER and ask permission to set up a Kalman filter profile and enable TCP/IP timestamps.

The worker instances perform the following tasks.

Worker

- Monitor the traffic or register TCP/IP timestamps for all controllers and workstations;
- Maintain a Kalman filter with trained noise parameters that computes the frequency offsets and drifts with respect to the managers in the system.
- For the existing device, if the clock parameters calculated with respect to a given manager are changing quickly, send a TAMPER to the corresponding manager.

C. Shortcomings of the Approach

The monitoring system will introduce computational and hardware overhead for the nodes in the network. For example, the matrix-vector computation for the Kalman filter is in the order of $O(N^3)$, which would require arithmetic capabilities at the worker nodes. Furthermore, the memory requirement for the anomaly detection will add additional memory overhead on the worker nodes. The detection scheme also suffers during sudden changes in the relative clock drift due to rapid environmental variation, network disruption, and changes in the operating condition.

VI. Conclusions and Future Work

In this paper, we have illustrated the application of the physical device fingerprinting techniques to detect tampering and intrusions on BACnet MS/TP devices at the automation layer. Our experimental results demonstrate that such hardware-based approaches are promising for use as novel security primitives in this era of the Internet-of-Things. For future work, we plan to conduct different experiments to test the proposed method's detection rate.

References

[1] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. A. Halderman *et al.*, "An internet-wide view of ics devices," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2016, pp. 96–103.

[2] Z. Zheng and A. N. Reddy, "Safeguarding building automation networks: The-driven anomaly detector based on traffic analysis," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2017, pp. 1–11.

[3] P. Ciholas, A. Lennie, P. Sadigova, and J. Such, "The security of smart buildings: a systematic literature review," *arXiv preprint arXiv:1901.05837*, 2019.

[4] C. B. Jones and C. Carter, "Trusted interconnections between a centralized controller and commercial building hvac systems for reliable demand response," *Ieee Access*, vol. 5, pp. 11 063–11 073, 2017.

[5] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, April 2005.

[6] S. J. Murdoch, "Hot or not: Revealing hidden services by their clock skew," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 27–36.

[7] A. STANDARD, "Ansvshrae standard 135-2008," *A data communication protocol for building automation and control network," ASHRAE Standing Standard Project Committee*, vol. 135, 2008.

[8] P. Zito, *Building Automation Systems a to Z: How to Survive in a World Full of Bas*. CreateSpace Independent Publishing Platform, 2016. [Online]. Available: https://books.google.com/books?id=aJr3MAAACAAJ

[9] K. McGowan, *Semiconductors: From Book to Breadboard*. Cengage Learning, 2012.

[10] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, 2005.

[11] M. T. Arafin, D. Anand, and G. Qu, "A low-cost gps spoofing detector design for internet of things (iot) applications," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*. ACM, 2017, pp. 161–166.

[12] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, p. 1550147718794615, 2018.

[13] W. Li, "Using genetic algorithm for network intrusion detection," *Proceedings of the United States department of energy cyber security group*, vol. 1, pp. 1–8, 2004.

[14] A. A. Gendreau and M. Moorman, "Survey of intrusion detection systems towards an end to end secure internet of things," in *2016 IEEE 4th international conference on future internet of things and cloud (FiCloud)*. IEEE, 2016, pp. 84–90.

[15] D. W. Allan, "Time and frequency(time-domain) characterization, estimation, and prediction of precision clocks and oscillators," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 34, no. 6, pp. 647–654, 1987.

[16] C. A. Greenhall, "A review of reduced Kalman filters for clock ensembles," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 59, no. 3, pp. 491–496, 2012.

[17] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.