POSTER: Cracking the Graph Routes in WirelessHART Networks

Xia Cheng
Department of Computer Science
State University of New York at Binghamton
Binghamton, New York, USA
xcheng12@binghamton.edu

Junyang Shi
Department of Computer Science
State University of New York at Binghamton
Binghamton, New York, USA
jshi28@binghamton.edu

Mo Sha*
Department of Computer Science
State University of New York at Binghamton
Binghamton, New York, USA
msha@binghamton.edu

ABSTRACT

As a key response to the Fourth Industrial Revolution, IEEE 802.15.4-based wireless sensor-actuator network (WSAN) technology is gaining rapid adoption in process industries because of its advantage in lowering deployment and maintenance cost and effort in industrial facilities, such as steel mills, oil refineries, and chemical plants. Although most industrial applications operate at low data rates, they often require their underlying networks to provide real-time and reliable data deliveries in harsh industrial environments. IEEE 802.15.4-based WSANs are appealing for use in industrial networks, since they operate at low-power and can be manufactured inexpensively. To meet the stringent real-time and reliability requirements, WSANs, such as WirelessHART networks, make a set of unique design choices such as employing the Time Slotted Channel Hopping (TSCH) and graph routing that distinguish themselves from traditional wireless sensor networks designed for best effort services. However, the security aspects of this increasingly important class of wireless networks are insufficiently investigated in the literature. Our recent work shows that an attacker can reverse engineer the TSCH channel hopping sequences by silently observing the channel activities and put the network in danger of selective jamming attacks, where the attacker jams only the transmission of interest on its specific communication channel in its specific time slot, which makes the attacks energy-efficient and hardly detectable. A critical step for an attacker to launch selective jamming is to identify the routing paths. Our study shows that an attacker can crack the routes used by the graph routing in WirelessHART networks by silently observing the packet transmission activities. In this poster proposal, we present a vulnerability analysis and our case study performed on a 50-device physical testbed using a publicly accessible WirelessHART implementation.

KEYWORDS

Industry 4.0; Wireless Sensor-Actuator Networks; Graph Routing; WirelessHART

ACM Reference Format:
Xia Cheng, Junyang Shi, and Mo Sha. 2019. POSTER: Cracking the Graph Routes in WirelessHART Networks. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3321705.3331004

1 INTRODUCTION

As the extension of conventional Internet connections, the Internet of Things (IoT) brings broader connectivity into physical devices and everyday objects. Most works related to IoT nowadays focus on smart homes, mobile devices, and wearables. However, it is the industrial IoT that has the potential to provide one of the largest economic impacts in the future. Although most industrial applications operate at low data rates, they often require their underlying networks to provide real-time and reliable data deliveries in harsh industrial environments. Such challenging requirements have been traditionally met by wired solutions, e.g., the Highway Addressable Remote Transducer (HART) communication protocol [5], where a controller collects readings from sensors and sends commands to actuators through cables. However, the wired networks are usually costly to deploy and maintain in industrial environments and hard to update to meet new automation requirements.

As a key response to the Fourth Industrial Revolution (or Industry 4.0) [7], IEEE 802.15.4-based wireless sensor-actuator network (WSAN) technology is gaining rapid adoption in process industries because of its advantage in lowering deployment and maintenance cost and effort in industrial facilities, such as steel mills, oil refineries, and chemical plants [9]. IEEE 802.15.4-based WSANs operate at low-power and can be manufactured inexpensively. Battery-powered wireless modules easily and inexpensively retrofit existing sensors and actuators in industrial facilities without running cables for communication and power. To meet the stringent real-time and reliability requirements, the industrial WSAN standards, such as WirelessHART [6], make a set of specific design choices such as employing the Time Slotted Channel Hopping (TSCH) and reliable graph routing technologies that distinguish themselves from traditional wireless sensor networks (WSNs) designed for best effort services [9]. TSCH combines time-slotted media access control (MAC) access, multi-channel communication, and channel hopping to support real-time packet deliveries and combat narrowband interference and multi-path fading. Specifically, it divides time into slots of a fixed length that are grouped into a slotframe. In a time slot, only one transmission is scheduled on each channel across the whole network which can prevent channel contention to enhance network reliability. All devices in a network are time synchronized and hop channels to exploit frequency diversity. Graph routing

[*] Corresponding author.

AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6752-3/19/07…$15.00
https://doi.org/10.1145/3321705.3331004

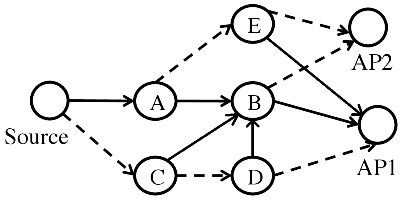

Figure 1: A graph routing example. The solid lines represent the primary paths and the dashed lines represent the backup paths).

is designed to enhance network reliability by taking advantage of route diversity. Under graph routing, each device maintains at least two neighbors to which they may send packets. As Figure 1 illustrates, the packet may take backup routes (through node C, D, or E) to reach the access points (AP1 and AP2) if the links on the primary path (through nodes A and B) fail to deliver a packet. Multiple redundant routes based on a routing graph can be used to combat external interference and enhance network reliability.

However, the security aspects of this increasingly important class of wireless networks are insufficiently investigated in the literature. Our recent work shows that the TSCH's function-based channel hopping simplifies the network operations at the cost of network security [1, 2]. An attacker can reverse engineer the TSCH channel hopping sequences by silently observing the channel activities and put the network in danger of selective jamming attacks, where the attacker jams only the transmission of interest on its specific communication channel in its specific time slot. Compared with the traditional jamming attacks, the selective jamming attacks post a much severer threat to WSANs, since those traditional jamming attacks can be easily detected and located by a wireless intrusion prevention system thanks to decades of research [11–13, 16, 17, 19]. Many countermeasures have been proposed in the WSN literature to combat the traditional jamming attacks, such as adjusting routing [3, 8, 18]. However, the existing approaches may fail to detect selective jamming, since the transmission failures caused by such attacks only happen occasionally and are buried in the normal fluctuations of low-power links. To launch selective jamming attacks to a TSCH-based WSAN, the attacker not only needs to know the channel hopping sequences, but also needs to crack the routing paths in the network. Our study shows that can crack the routes used by the graph routing in WirelessHART networks by silently observing the packet transmission activities. In this poster proposal, we present a vulnerability analysis and our case study performed on a 50-device physical testbed using a publicly accessible WirelessHART implementation.

The remainder of the poster proposal is organized as follows. Section 2 introduces our vulnerability analysis. Section 3 presents our case study. Section 4 concludes the poster proposal.

2 VULNERABILITY ANALYSIS ON GRAPH ROUTING

In this section, we present our vulnerability analysis on the graph routing in WirelessHART networks by demonstrating how an attacker cracks the graph routes by silently observing the packet transmission activities.

In WirelessHART networks, a Data-Link Protocol Data Unit (DLPDU) provides means for reliable communications at the Data-Link Layer (DLL). The WirelessHART DLPDU consists of Sequence Number, Network ID, destination address, source address, DLL payload, Message Integrity Code (MIC), and some other fields. And a Network Protocol Data Unit (NPDU) is stored in the DLL payload, which is composed of protocol specific control information and user data. WirelessHART does not require the network to encrypt the DLPDU and NPDU headers due to the communication overhead concern. The source and destination addresses of a communicating link are stored in the DLPDU header, while the address of the device which originally generated the packet and the final destination address of the packet are stored in the NPDU header. The attacker can use such information stored in the eavesdropped packets to derive the graph routes. Specifically, the attacker can execute the following steps to crack the routes used by the graph routing in WirelessHART networks:

(1) **Grouping the eavesdropped packets**: The attacker snoops the packet transmission activities and groups the eavesdropped packets based on their original source and final destination addresses stored in the unencrypted NPDU headers. All packets that share the same original source and final destination addresses in the NPDU headers belong to the same data flow.
(2) **Sorting the packets by capture time:** In each group of packets, the attacker sorts the packets by their time stamps at capture.
(3) **Identifying the primary routing path for each data flow:** In each group of sorted packets, the attacker identifies the primary routing path of each data flow by comparing the addresses storing in the DLPDU header with the ones in the NPDU header. As each DLPDU header includes the source and destination addresses for the communicating link, the attacker can identify all intermediate devices located on the primary routing path by checking the sorted packets one by one until when the link destination address in a DLPDU header is the same as the path destination address in the NPDU header carried by the DLPDU.
(4) **Identifying the backup routes:** As each device located on the primary routing path may transmit packets through its backup routes, the attacker can identify those backup routes by selectively jamming some links on the primary path.

With the cracked graph routes and TSCH channel hopping sequences, the attacker can launch the selective jamming attacks by jamming only the transmission of interest on its specific communication channel in its specific time slot.

3 CASE STUDY

We perform a case study using a publicly accessible WirelessHART implementation with graph routing [15] and run the experiments on our testbed with 50 TelosB motes deployed in the second floor of the Engineering Building in our university campus [14]. Figure 2 shows the testbed deployment. We configure the network to have two access points and 48 field devices operating on four channels. We set up six data flows with different sources, destinations, data periods, and priorities, as Table 1 lists. A maximum of three transmission

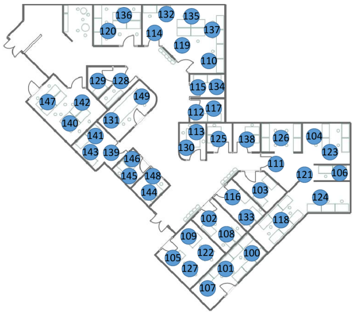

Figure 2: Testbed with 50 TelosB motes.

Table 1: Six data flows configured in WirelessHart network.

Flow	Sensor	Actuator	Period	Priority
1	147	126	320ms	1
2	144	143	640ms	2
3	105	104	1280ms	3
4	149	102	2560ms	4
5	136	135	5120ms	5
6	137	108	10240ms	6

Table 2: Cracking Result

Routing Path	Snooping Time
147-131-103-121-126	1920ms
144-108-121-101-105-143	1920ms
105-101-121-124-104	5120ms
149-113-103-121-101-102	7680ms
136-135-110-113-103-121-103-113-110-135	15360ms
137-110-113-103-121-108	40960ms

attempts are scheduled for each packet. The first two transmission attempts are scheduled through the primary route and the last attempt uses the backup route. Rate monotonic scheduling is used to generate packet transmission schedules.

The attacker is assumed to be a device which is capable of monitoring activities on all 16 channels in 2.4 GHz ISM band and has moderate computational capability (e.g., a Raspberry Pi 3 Model B [4] integrated with a Wi-Spy USB Spectrum Analyzer [10]). The attacker has neither prior knowledge on the routing nor the configuration of transmission and retransmission. The cracking goal is to derive the primary routing path and its corresponding time slot offset in a slotframe. We leave the cracking of the backup routes as future work.

Table 2 shows the primary routing path of each data flow cracked by our attacking program and the time consumed by the cracking process. For instance, the attacking program uses 1920ms to derive the primary routing path from node 147 to node 126. The time consumption is roughly six times of the data period of this data flow. From the results, we can see that the attacker can successfully identify the primary routing path of each data flow. The time consumption mainly depends on the time consumed to snoop the channels and eavesdrop enough on-air packets.

4 CONCLUSIONS

To meet the stringent real-time and reliability requirements posed by industrial IoT applications, WirelessHART networks make a set of unique design choices, such as employing TSCH and graph routing technologies. Our study shows that the unencrypted DLPDU and NPDU headers allow an attacker to identify the primary routing paths in the network. With the cracked routing and channel usage information, an attacker puts the network in danger of selective jamming attacks, where the attacker jams only the transmission of interest on its specific communication channel in its specific time slot, posting a realistic, severe threat to WSANs.

ACKNOWLEDGMENT

This work was supported by the NSF through grant CRII-1657275 (NeTS).

REFERENCES

[1] Xia Cheng and Mo Sha. 2018. POSTER: Cracking the TSCH Channel Hopping in IEEE 802.15.4e. In *ACM SIGSAC Conference on Computer and Communications Security.*

[2] Xia Cheng, Junyang Shi, and Mo Sha. 2019. Cracking the Channel Hopping Sequences in IEEE 802.15.4e-Based Industrial TSCH Networks. In *Internet of Things Design and Implementation (IoTDI).*

[3] Jing Deng, Richard Han, and Shivakant Mishra. 2003. A Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks. In *ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN).*

[4] Raspberry Pi Foundation. 2018. Raspberry Pi. Retrieved September 17, 2018 from https://www.raspberrypi.org/

[5] FieldComm Group. 2007. HART Communication Protocol and Foundation (Now the FieldComm Group). Retrieved September 17, 2018 from https://fieldcommgroup.org/

[6] FieldComm Group. 2007. WirelessHART. Retrieved September 17, 2018 from https://fieldcommgroup.org/technologies/hart/hart-technology

[7] Henning Kagermann, Wolfgang Wahlster, and Johannes Helbig. April 2013. Recommendations for Implementing the Strategic Initiative Industrie 4.0. https://www.acatech.de/wp-content/uploads/2018/03/Final_report__Industrie_4.0_accessible.pdf

[8] Chris Karlof, Naveen Sastry, and David Wagner. 2004. TinySec: a Link Layer Security Architecture for Wireless Sensor Networks. In *ACM Conference on Embedded Networked Sensor Systems (SenSys).*

[9] Chenyang Lu, Abusayeed Saifullah, Bo Li, Mo Sha, Humberto Gonzalez, Dolvara Gunatilaka, Chengjie Wu, Lanshun Nie, and Yixin Chen. 2016. Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems. In *Proceedings of the IEEE, Special Issue on Industrial Cyber Physical Systems*, Vol. 104.

[10] METAGEEK. 2018. Wi-Spy USB Spectrum Analyzer. Retrieved September 17, 2018 from http://www.wi-spy.co.uk/index.php/products/wi-spy

[11] Aristides Mpitziopoulos, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. 2009. A Survey on Jamming Attacks and Countermeasures in WSNs. *IEEE Communications Surveys and Tutorials* 11, 4 (2009), 42–56.

[12] Konstantinos Pelechrinis, Marios Iliofotou, and Srikanth V. Krishnamurthy. 2011. Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses. *IEEE Communications Surveys and Tutorials* 13, 2 (2011), 245–257.

[13] David R. Raymond and Scott F. Midkiff. 2008. Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses. *IEEE Pervasive Computing* 7, 1 (2008), 1536–1268.

[14] Mo Sha. 2016. Binghamton University Wireless Embedded System Testbed. Retrieved September 17, 2018 from http://www.cs.binghamton.edu/~msha/testbed

[15] Wireless Cyber-Physical Simulator (WCPS). 2013. Wireless Cyber-Physical Simulator (WCPS). http://wsn.cse.wustl.edu/index.php/WCPS:_Wireless_Cyber-Physical_Simulator

[16] A.D. Wood, J.A. Stankovic, and S.H. Son. 2003. JAM: a Jammed-area Mapping Service for Sensor Networks. In *IEEE Real-Time Systems Symposium (RTSS).*

[17] Anthony D. Wood and John A. Stankovic. 2002. Denial of Service in Sensor Networks. *Computer* 35, 10 (2002), 54–62.

[18] Anthony D. Wood, John A. Stankovic, and Gang Zhou. 2007. DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON).*

[19] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. 2005. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc).*