Blockchain-Based Man-in-the-Middle (MITM) Attack Detection for Photovoltaic Systems

Jinchun Choi
Dept. of Electrical Engineering and Computer Science
Texas A&M University-Kingsville
Kingsville, TX, 78363 USA
jinchun.choi@tamuk.edu

Bohyun Ahn
Dept. Electrical Engineering and Computer Science
Texas A&M University-Kingsville
Kingsville, TX, 78363 USA
bohyun.ahn@students.tamuk.edu

Gomanth Bere
Dept. Electrical Engineering and Computer Science
Texas A&M University-Kingsville
Kingsville, TX, 78363 USA
gomanth.bere@tamuk.edu

Seerin Ahmad
Dept. Electrical Engineering and Computer Science
Texas A&M University-Kingsville
Kingsville, TX, 78363 USA
seerin.ahmad@students.tamuk.edu

Homer Alan Mantooth
Dept. Electrical Engineering
University of Arkansas
Fayetteville, AR, 72701 USA
mantooth@uark.edu

Taesic Kim*
Dept. Electrical Engineering and Computer Science
Texas A&M University-Kingsville
Kingsville, TX 78363 USA
taesic.kim@tamuk.edu

2021 IEEE Design Methodologies Conference (DMC) | 978-1-6654-0301-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/DMC51747.2021.9529949

***Abstract*—Cybersecurity of photovoltaic (PV) systems entails a much larger scope than just encryption and firewall of communications. For instance, integrity of data in transit between inverters and a cloud server can be compromised by authorized third-party, devices, and internal network within security perimeter (i.e., man-in-the-middle (MITM) attack). To address this challenge, this paper proposes a blockchain-based MITM attack detection method for a PV system. A breakthrough method includes screening network data, network intrusion detection, and hash comparison of in-transit data using distributed ledgers. The proposed method is implemented in Internet-of-Thing (IoT) security modules as clients of a blockchain network and validated by experiments.**

Keywords—blockchain, cybersecurity, man-in-the-middle attack, photovoltaic system, security module

I. Introduction

Cybersecurity of photovoltaic systems still relies on network-based security postures such as authentication of users, firewall rules, and the encryption of communication based on transport layer security (TLS) [1], [2]. However, security entails a much larger scope than current network-based security methods [3]. Human risks always exist, which threatens users' passwords. Firewall rules cannot screen spoofed messages/malware and still contain software vulnerabilities that may allow attackers to bypass their protection [4]. Moreover, encryption just ensures that the encrypted data cannot be understood. Man-in-the-middle (MITM) attacks enable to modify in-transit data between a cloud server and solar inverters if any middle systems/devices such as third-party services (e.g., virtual private network (VPN)), network devices (gateway, switches), other non-end equipment in a PV system is compromised [5]. The in-transit data flows through the malicious middle node will be read and modified before it is arrived at the destination.

Recently, Case studies on demonstrating MITM attacks targeting a PV system have been studied. In [6], A MITM attack on smart meter data (e.g., grid active and reactive power) for ancillary services such as reactive power compensation and reverse power flow mitigation using a commercial PV inverter is demonstrated in a lab experimental environment. MITM Attacks on the IEC 61850 manufacturing message specification used in a PV system communication is investigated in [7], which shows how a custom tool to perform MITM attacks, manipulate sensor and control command data, and affect the PV physical system. Such MITM attacks are developed by accessing the local network of the PV system and fully utilize the vulnerability of the network protocols such as Modbus.

Real-time intrusion detection methods have been widely studied to detect forged data by MITM attacks (e.g., sensor data and PQ set-points used for PV inverter controllers) using signature/rule-based network intrusion detection (e.g., detecting irregular network packet format, reply, and message authentication) [8], behavior-based detection using artificial intelligence [9], model-based methods [10] (e.g., detecting anomaly behaviors of inverters and PV system or network traffic patterns) and tamper detection using signal process methods (e.g., water marking) [11]. However, intrusion detection for the advanced MITM attacks such as authorized device MITM and a third party MITM have been less studied.

Blockchain technology can provide a secure distributed system framework currently available in information and communication technology (ICT) applications utilizing the latest cryptography, public key infrastructure, consensus, and access control mechanisms. By incorporating smart contracts, the recent emergence of blockchain technology has been mostly studied in power sector such as electricity trading platform [12], demand-side energy management [13]. Recently, the authors have proposed a blockchain-based network for an IoT-enabled micro solar inverter [14] and smart contract-defined control and co-simulation for smart inverters in a PV system [15]. However, the practical investigation of MITM attack detection for PV systems against the advanced MITM cases such as malware injection and firmware modification using blockchain technology has not been studied to the authors' best knowledge.

This paper introduces a potential vulnerability of in-transit data modification by MITM attacks in a PV system and proposes a blockchain-based MITM attack detection method. The proposed method utilizes security modules attached to operational technology (OT) devices such as inverters and a gateway/aggregator device (e.g., site data manager) as blockchain client nodes in a PV system. The proposed

This work was supported in part by the Department of Energy (DOE) under award No. DE-EE0009026. Funding was provided in part by U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Solar Energy Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government.

978-1-6654-0301-6/21/$31.00 ©2021 IEEE

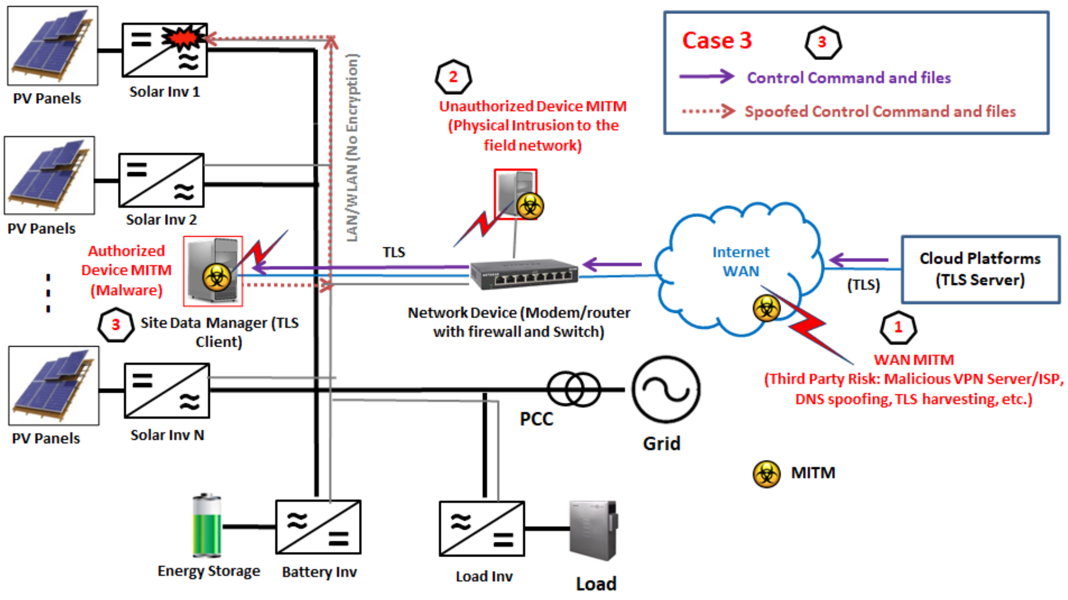

PV Panels
Solar Inv 1
PV Panels
Solar Inv 2
Authorized Device MITM (Malware)
3
Site Data Manager (TLS Client)
PV Panels
Solar Inv N
LAN/WLAN (No Encryption)
2
Unauthorized Device MITM (Physical Intrusion to the field network)
TLS
Network Device (Modem/router with firewall and Switch)
Case 3
3
Control Command and files
Spoofed Control Command and files
Internet WAN
(TLS)
Cloud Platforms (TLS Server)
1
WAN MITM (Third Party Risk: Malicious VPN Server/ISP, DNS spoofing, TLS harvesting, etc.)
PCC
Grid
MITM
Energy Storage
Battery Inv
Load Inv
Load

Fig. 1. Cases of MITM attack in a PV system that modify in-transit data from cloud platforms.

mechanism is used to check the authentication, integrity and authorization of the in-transit data by using smart contracts which continuously executes hash comparisons by the security modules and the blockchain network. Experimental results validate the proposed MITM attack detection framework.

II. Case Studies of In-Transit Data Modification through MITM Attacks

A. *Cyber-Physical PV System Configuration*

Fig. 1 shows a diagram of a PV system consisting of multiple smart inverters connected to PV panels to supply local loads, common loads, a battery energy storage system (BESS) and power to the power grid. The cloud platforms such as an operator's remote-control center and vendor server are connected to the PV system gateway through wide area network (WAN) which relies on security properties of TLS (i.e.., client-server authentication, data encryption and integrity check). TLS establishes an encrypted connection between an authenticated server and a client. IEEE 1547-2018 standard protocols for distributed energy resources (DER) such as DNP3 and SEP2.0 utilize TLS as a key security posture. Although individual smart inverters as TLS clients enable to directly communicate with the cloud server, a PV system typically utilizes a site data manager that efficiently aggregates and manages multiple smart inverters in the PV system. The multiple smart inverters are typically connected to the site data manger using the insecure LAN or wireless LAN (WLAN) with a Modbus TCP/RTU protocol. Moreover, utility meters/phasor measurement units (PMUs) and other systems such as BESS can exchange data with the site data manager. This paper considers in-transit data such as firmware update files and control commands from the cloud server are critical data requiring high security level since the data can significantly change operational conditions of the PV system.

B. *Three Cases of In-Transit-Data through MITM Attacks*

Fig. 1 shows three potential MITM attack cases that can change in-transit control command or files in a PV system: 1) wide area network (WAN) MITM, 2) unauthorized device MITM; and 3) authorized device MITM. WAN MITM attack could be caused by third-party such as a VPN provider, a domain name server (DNS), and internet service provider (ISP)). Since the security of the third-party is out of security perimeter of the PV system, it is hard to validate data passed by the malicious third-party or breached third-party by hackers. Although people consider the TLS is currently secure, advanced attacks such as TLS harvesting may break the TLS (e.g., stealing session key logs).

Second, an unauthorized MITM device will be physically located and connected to the local area network (LAN). This can make an unauthorized MITM attack. Field network protocols without strong authentication and encryption are vulnerable to this MITM attack. It is noted that most MITM attack detection in PV systems applied this attack scenario.

Due to the malware injection attacks, the authorized devices can be a MITM attack devices. As shown in Fig. 1, the site data manager is an aggregator and a gateway in a PV system acting as a major middleman between the inverters and the cloud. The encrypted TLS data are decrypted and converted to the local network protocols such as Modbus TPC in the site data manager. Therefore, malicious site data manager can easily make MITM attacks although this device is authenticated and authorized in the current PV system security perimeter such as firewall rules and encryption-based security controls. Although, the site data manager will be a critical target device in attacker's perspective, compromised inverters or OT network devices also can create the MITM attacks.

III. Proposed Blockchain-Based MITM Detection

A. *Cooperative Security Ecosystem using Blockchain*

Fig. 2 illustrates the overall concept of blockchain based zero-trust ecosystem for a PV system. A private blockchain network can build a collaborative security ecosystem where multiparty (e.g., utility, operator, vendors, and security service provider) can seamlessly handle the user- or vendor-identified incidents through effective notification, coordination, disclosure, and validation mechanism, while considering privacy of the PV system using smart contract and multichannel blockchain. The multichannel platform is a

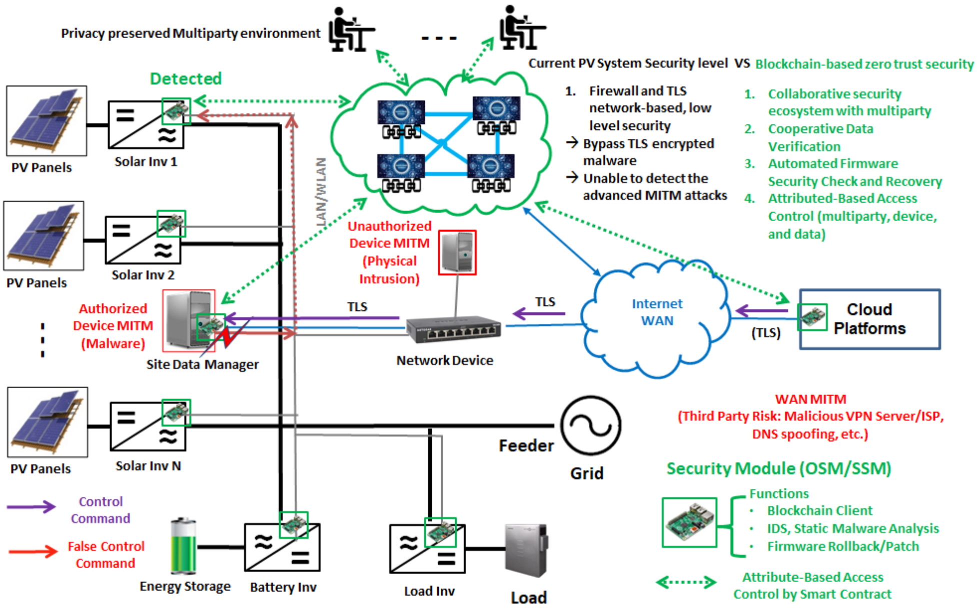

Privacy preserved Multiparty environment
Current PV System Security level VS Blockchain-based zero trust security
1. Firewall and TLS network-based, low level security
→ Bypass TLS encrypted malware
→ Unable to detect the advanced MITM attacks
1. Collaborative security ecosystem with multiparty
2. Cooperative Data Verification
3. Automated Firmware Security Check and Recovery
4. Attributed-Based Access Control (multiparty, device, and data)
Detected
PV Panels
Solar Inv 1
PV Panels
Solar Inv 2
LAN/WLAN
Unauthorized Device MITM (Physical Intrusion)
Authorized Device MITM (Malware)
Site Data Manager
TLS
TLS
Network Device
Internet WAN
(TLS)
Cloud Platforms
WAN MITM
(Third Party Risk: Malicious VPN Server/ISP, DNS spoofing, etc.)
PV Panels
Solar Inv N
Feeder
Grid
Control Command
False Control Command
Energy Storage
Battery Inv
Load Inv
Load
Security Module (OSM/SSM)
Functions
Blockchain Client
IDS, Static Malware Analysis
Firmware Rollback/Patch
Attribute-Based Access Control by Smart Contract

Fig. 2. A concept of block-based zero trust security for a PV system.

modular architecture where peer nodes will be in a cloud system or party's server and uses parallelism by assigning a separate ordering and fast peer server per channel. The proposed blockchain platform is designed for zero-trust security (i.e., no network-related entity including sensitive data, devices, applications, systems can be trusted). The consistent and continuous process of verifying identity, validating activity, and limiting access and privilege will increase trustworthiness of the system security services to ensure integrity and authenticity of critical assets, thus providing a viable way to manage the evolving cyber risks on the PV systems.

Security modules are attached/installed in the critical devices such as cloud, site data manager, and smart inverters. The security module will be a software module or an on-board hardware module defending on the resources of the critical devices. The security module mainly consists of blockchain client program, intrusion detection system (IDS), static malware analysis, and firmware rollback/patch. The blockchain client enable to submit transactions, access ledgers, and public key infrastructure (PKI, as part of membership service) such events can be controlled by smart contracts. In this paper, we focus on the MITM detection scheme using the blockchain-based cooperative in-transit data verification process. Fig. 3 show the block diagram of the proposed blockchain platform for an in-transit control command integrity validation scenario where the control command is considered as a critical asset. Therefore, the authentication, integrity, and authorization of the critical in-transit data are keeping verified, and the results are stored in the ledger as security logs.

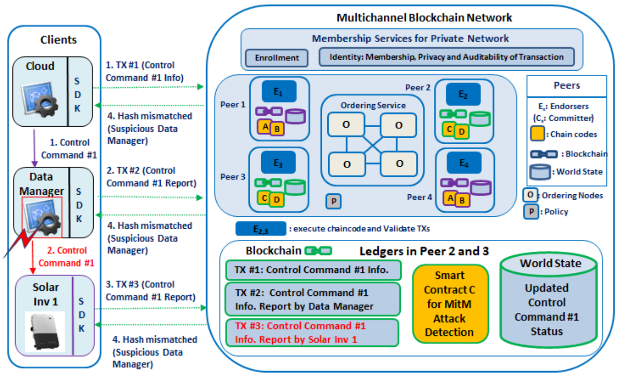
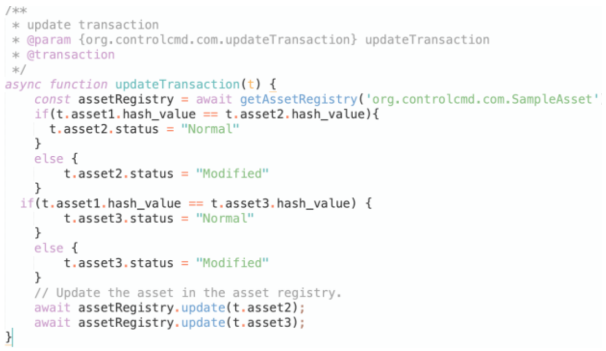

Clients
Cloud
SDK
1. Control Command #1
Data Manager
SDK
2. Control Command #1
Solar Inv 1
SDK
1. TX #1 (Control Command #1 Info)
4. Hash mismatched (Suspicious Data Manager)
2. TX #2 (Control Command #1 Report)
4. Hash mismatched (Suspicious Data Manager)
3. TX #3 (Control Command #1 Report)
4. Hash mismatched (Suspicious Data Manager)
Multichannel Blockchain Network
Membership Services for Private Network
Enrollment
Identity: Membership, Privacy and Auditability of Transaction
Peer 1
E_1
A B
Peer 2
E_2
C D
Ordering Service
O O O O
Peer 3
E_3
C D
Peer 4
E_4
A B
P
Peers
E_x: Endorsers
(C_x: Committer)
: Chain codes
: Blockchain
: World State
O : Ordering Nodes
P : Policy
$E_{2,3}$: execute chaincode and Validate TXs
Blockchain
Ledgers in Peer 2 and 3
TX #1: Control Command #1 Info.
TX #2: Control Command #1 Info. Report by Data Manager
TX #3: Control Command #1 Info. Report by Solar Inv 1
Smart Contract C for MitM Attack Detection
World State
Updated Control Command #1 Status

Fig. 3. Cooperative in-transit data validation using blockchain.

B. *MITM Attack Deteciton*

In the private blockchain network, authorized parties such as PV system vendors, an operator, utility, and security modules will be the clients that are authorized persons/devices proving data as a form of transactions to the blockchain network and can access/share their data stored in their blockchain ledgers. Uploading and accessing data in the ledgers are mutually agreed upon and programmed by smart contracts. Only authorized parties using the blockchain client program can create transactions that include hash values of control commands/files, and the blockchain network considers the control command or file update as an authorized event. Therefore, the blockchain network can provide increased visibility into the methods, applications, and services to easily ensure the integrity and authenticity of the control command/file assets. After the provide the hash values to the blockchain ledger, the smart contract is running for the integrity check without trusting existing security perimeters such as TLS and firewall whitelist.

Fig. 4 shows an example of smart contract for the hash comparison between blockchain ledgers and in-transit data received by the security modules. Algorithm 1 describes the process of the integrity checking algorithm used for the hashed value of the control command. When the cloud sends a control command to the PV system, the Cloud security module initializes the integrity checking process. The Cloud security module calculates hash value of the control command using SHA-256, updates the information to the blockchain network. Next, the Data Manager security module receives the control command from the cloud and calculates the hash of the received control command. After calculation, the Data Manger security module builds JSON data type with



```
/**
 * update transaction
 * @param {org.controlcmd.com.updateTransaction} updateTransaction
 * @transaction
 */
async function updateTransaction(t) {
    const assetRegistry = await getAssetRegistry('org.controlcmd.com.SampleAsset'
    if(t.asset1.hash_value == t.asset2.hash_value){
      t.asset2.status = "Normal"
    }
    else {
        t.asset2.status = "Modified"
    }
  if(t.asset1.hash_value == t.asset3.hash_value) {
        t.asset3.status = "Normal"
    }
    else {
        t.asset3.status = "Modified"
    }
    // Update the asset in the asset registry.
    await assetRegistry.update(t.asset2);
    await assetRegistry.update(t.asset3);
}
```

Fig. 4. An example of a smart contract that compares a hash value for integrity check and update status.

calculated hash value of the received control command and issue a transaction with the hash value. A smart contract compares the two hash values in the blockchain ledger. If they are matched, update the status as "Normal", otherwise the status will be "Modified". Inverter security modules that received the control command will report the hash value of the control command as well. After the comparison process, the asset security status will be updated in World State. For example, as shown in Fig. 3, the first control command is provided by the cloud security module in the cloud and issue the status of the control command is *Normal*. Data Manager security module received control command and sends its hash value to the blockchain network with *Unchecked* status. After running a smart contract comparing hash values of the control command, the status is updated as *Normal* if the comparing results are true. If the results are not matched, the status will be updated as *Modified*. Therefore, the blockchain will automatically detect a location/device that change the integrity of the control command.

Algorithm 1: Security Decision Algorithm for Report and Integrity Check using Blockchain

```
1: Initialization: Report information of control command (i.e.,
                   asset_cloud) including sender, receiver, hash
                   value, timestamp, status; to blockchain
                   network by the cloud server
2: Calculate: Store the hash of the received asset using SHA-
              256 by security module
3: Report: Build a JSON data type with information including
           hashed value and status "Uncheck" to send to the
           blockchain network
4: Comparison: Execute the smart contract from the
               blockchain client to check integrity of the
               asset (asset_client)
5: if asset_cloud == None
6: then update asset_client's status = "Modified"
7: if asset_cloud.hash_value == asset_client.hash_value
8: then update asset_client's status = "Normal"
9: else update asset_client's status = "Modified"
10: Update: Update the asset_client's value in the blockchain
            ledger
11: End
```

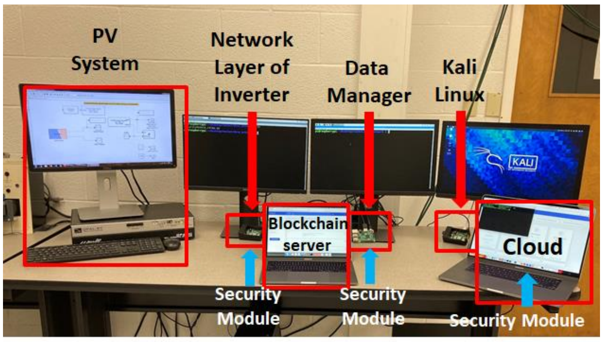
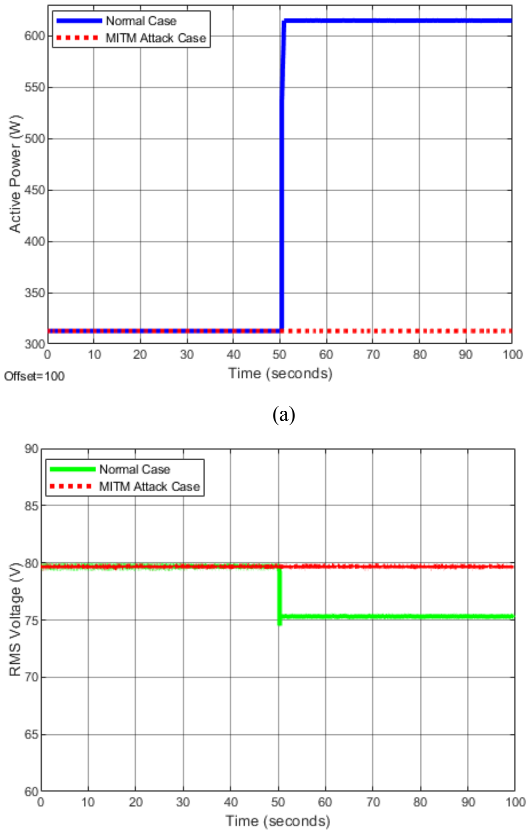


Fig. 5. The real-time HIL security testbed for a PV system.

IV. Validation

A. Experimental Setup

Fig. 5 illustrates the real-time HIL security testbed including 1) a real-time PV system simulator using an OPAL-RT's real-time simulator; 2) a cyber system testbed using real network system and servers including cloud and blockchain server (Hyperledger-Fabric [16]) installed in each laptop; and 3) Kali Linux penetration testing tool generating real cyber-attacks as real cyber events. In this paper, a MATLAB/Simulink model of the PV system [15] is converted to a RT-LAB model to interface with the OPAL-RT for the real-time PV system operation. A network layer of the smart



(a)

(b)

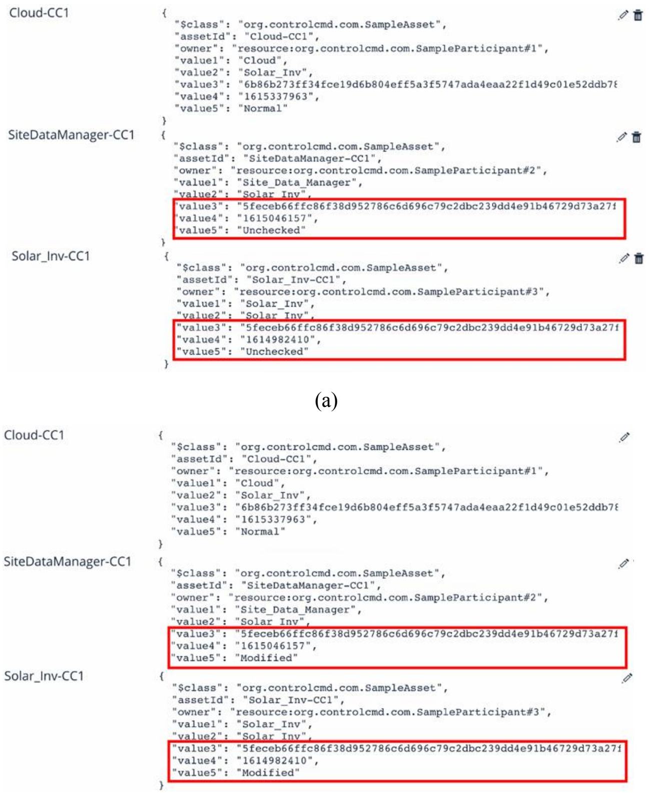
Fig. 6. Impact of the modified control command: (a) active power and (b) RMS voltage.

inverter, a data manager, and the Kali Linux attack device are emulated by Raspberry Pi 4Bs. An inverter security module, a data manger security module and a cloud security are implemented in corresponding devices.

An WAN MITM attack and an unauthorized device MITM attack are executed by using Ettercap of Kali Linux [17] which can sniff a live connection in the network, filter and modify the network packets. For the WAN MITM attack, Kali Linux targets the network between the cloud and the data manger, assuming that Kali Linux is installed in a WAN third-party platform. The unauthorized MITM attack is created by sending a control command to a target device through the LAN. The authorized device MITM attack is conducted by using a customized malware code installed in the data manager emulator.

B. *MITM Attaack Deteciton*

Initial load status of an inverter was set as the half load where Load 1 is connected, and Load 2 is disconnected. Around 50 seconds, an operator want to connect Load 2 by sending the control command (Load 2 = on) from the cloud server due to the enough power capability of the PV panel connected to the inverter. The MITM attack change the in-transit control command to keep turning off Load 2 while the load condition needs to be changed to the full load after 50 seconds. Figs. 6(a) and (b) show the attack impact on the active power and root mean square (RMS) voltage supplied by the inverter, respectively. In the normal case, the control command provided by the cloud successfully increases the active power of the inverter around 50 seconds. However, the control command was interrupted by the malicious data manger.



```
Cloud-CC1              {
                         "$class": "org.controlcmd.com.SampleAsset",
                         "assetId": "Cloud-CC1",
                         "owner": "resource:org.controlcmd.com.SampleParticipant#1",
                         "value1": "Cloud",
                         "value2": "Solar_Inv",
                         "value3": "6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb78
                         "value4": "1615337963",
                         "value5": "Normal"
                       }
SiteDataManager-CC1    {
                         "$class": "org.controlcmd.com.SampleAsset",
                         "assetId": "SiteDataManager-CC1",
                         "owner": "resource:org.controlcmd.com.SampleParticipant#2",
                         "value1": "Site_Data_Manager",
                         "value2": "Solar_Inv",
                         "value3": "5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27f
                         "value4": "1615046157",
                         "value5": "Unchecked"
                       }
Solar_Inv-CC1          {
                         "$class": "org.controlcmd.com.SampleAsset",
                         "assetId": "Solar_Inv-CC1",
                         "owner": "resource:org.controlcmd.com.SampleParticipant#3",
                         "value1": "Solar_Inv",
                         "value2": "Solar_Inv",
                         "value3": "5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27f
                         "value4": "1614982410",
                         "value5": "Unchecked"
                       }
```

(a)

```
Cloud-CC1              {
                         "$class": "org.controlcmd.com.SampleAsset",
                         "assetId": "Cloud-CC1",
                         "owner": "resource:org.controlcmd.com.SampleParticipant#1",
                         "value1": "Cloud",
                         "value2": "Solar_Inv",
                         "value3": "6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb78
                         "value4": "1615337963",
                         "value5": "Normal"
                       }
SiteDataManager-CC1    {
                         "$class": "org.controlcmd.com.SampleAsset",
                         "assetId": "SiteDataManager-CC1",
                         "owner": "resource:org.controlcmd.com.SampleParticipant#2",
                         "value1": "Site_Data_Manager",
                         "value2": "Solar_Inv",
                         "value3": "5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27f
                         "value4": "1615046157",
                         "value5": "Modified"
                       }
Solar_Inv-CC1          {
                         "$class": "org.controlcmd.com.SampleAsset",
                         "assetId": "Solar_Inv-CC1",
                         "owner": "resource:org.controlcmd.com.SampleParticipant#3",
                         "value1": "Solar_Inv",
                         "value2": "Solar_Inv",
                         "value3": "5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27f
                         "value4": "1614982410",
                         "value5": "Modified"
                       }
```

(b)

Fig. 6. MITM attack detection results in blockchain ledger: (a) before detection and (b) after detection.

Figs. 7(a) and (b) present the MITM attack detection results in the blockchain ledger corresponding to the control command attack by the authorized device MITM. The blockchain ledger clearly shows that the blockchain automatically detect the mismatched control command due to the compromised site data manager. Therefore, blockchain-based MITM attack detection method can automatically detect the advanced MITM attacks and provide security status of assets in the PV system.

V. Conclusion

This paper provides a potential vulnerability of in-transit data modification by advanced MITM attacks in PV systems. To address this threat, we propose a blockchain-based MITM attack detection. The proposed method utilizes security modules attached to OT devices in a PV system and distributed blockchain network with users or vendors involved to build a cooperative data integrity validation ecosystem. The proposed method can detect MITM attacks modifying in-transit data by keep tracing authentication, integrity, and authorization of the data as well as provide security logs of the critical assets, which can zero-trust system for PV systems.

References

[1] J. Johnson, "Roadmap for photovoltiac cyber security," Sandia National Laboratories, Dec. 2017.

[2] J. Johnson, I. Onunkwo, P. Cordeiro, B. J. Wright, N. Jacobs, C. Lai, "Assessing DER network cybersecurity defences in a power-communication co-simulation environment, *IET Cyber Physical System*, Mar. 2020.

[3] EPRI, Zero trust consideration for utility OT cyber security strategies, Technical Report, June 2020.

[4] Alert (AA20-352A), Jan. 07, 2021. [Online]. Available: https://us-cert.cisa.gov/ncas/alerts/aa20-352a

[5] IEC 62351-12, Resilience and security recommendations for power systsm with distributed energy resources (DER) cyber-physical systems, Techincal report, 2016.

[6] G. Tertytchny et al., "Demonstration of man in the middle attack on a commercial photovoltaic inverter providing ancillary services," in *Proc. 2020 IEEE CyberPELS (CyberPELS)*, Miami, FL, USA, 2020, pp. 1-7.

[7] B. Kang et al., "Investigating cyber-physical attacks against IEC 61850 photovoltaic inverter installations," in *Proc. 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, Luxembourg, Luxembourg, 2015, pp. 1-8.

[8] C. B. Jones, A. R. Chavez, R. Darbali-Zamora, and S. Hossain-McKenzie, "Implementation of intrusion detection methods for distributed photovoltaic inverters at the grid-edge," in *Proc. 2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washinton, DC, Feb. 17-20, pp. 1–5.

[9] Q. Li, F. Li, J. Zhang, J. Ye, W. Song, and A. Mantooth, "Data-driven cyberattack detection for photovoltaic (PV) systems through analyzing micro-PMU data," in *Proc. 2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, Detroit, MI, USA, 2020, pp. 431– 436.

[10] Z. Zhang, M. Easley, M. Hosseinzadehtaher, G. Amariucai, M. B. Shadmand and H. Abu-Rub, "An observer based intrusion detection framework for smart inverters at the grid-edge," in *Proc. 2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, Detroit, MI, USA, 2020, pp. 1957–1962.

[11] J. Ramos-Ruiz et al., "An active detection scheme for cyber attacks on grid-tied PV systems," in *Proc. 2020 IEEE CyberPELS*, Miami, FL, USA, 2020, pp. 1–6.

[12] Microgrid Media, "It's like the early days of the internet, Blockchain-based microgrid tests P2P energy trading in Brooklyn," Mar. 2016.

[13] Z. Li, S. Bahramirad, A. Passo, M. Yan, and M. Shahidehpour, "Blockchain for decentralized transactive energy management system in networked microgrids," *The Electricity Journal*, vol. 32, pp. 58-72, Apr. 2019.

[14] A. A. Hadi, U. Sinha, T. Faika, T. Kim, J. Zeng, and M-H. Ryu, "Internet of Things (IoT)-enabled solar microinverter using blockchain

technology," in *Proc. 2019 IEEE IAS Annual Meeting*, Baltimore, MD, Sept. 29-Oct. 3, 2019, pp. 1-5.

[15] A. A. Hadi, G. Bere, B. Ahn, and T. Kim, "Smart contract-defined control and co-simulation for smart inverters in a photovoltaic (PV) system and blockchain network," in *Proc. 2020 IEEE CyberPELS Workshop*, Oct. 13, 2020, pp.1-6.

[16] Hyberledger-Fabric, [Online] Available, https://www.ibm.com/blockchain/hyperledger.html.

[17] Ettercap landing page, [Online]. Available: https://www.ettercap-project.org/