



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Safe And Cost-Efficient Blockchain- Enabled Framework For Secure IOT Software Updates.

M.Prathibha¹, Mr. K. Niranjana²

¹PG student, Vemu Institute of Technology, P. kothakota.

²Assistant Professor, Vemu Institute of Technology, P. kothakota.

ABSTRACT

This project introduces a groundbreaking solution to the security vulnerabilities encountered by resource-limited Internet of Things (IoT) devices during software updates. Traditional methods are fraught with security risks and inefficiencies due to multiple data transfers. The proposed blockchain-based framework revolutionizes this process by leveraging Ciphertext-Policy Attribute-Based Encryption (CP-ABE) for cryptographic tasks, custom authorization policies for enhanced security, and smart contracts to ensure secure delivery and payment. By storing encrypted software update blocks across multiple IPFS nodes and recording their addresses on the blockchain, the system mitigates the risk of a single point of failure. This innovative approach not only guarantees secure, efficient, and auditable software updates but also significantly bolsters IoT device security compared to conventional methods.

Keywords: Blockchain, Ciphertext

INTRODUCTION:

Internet of Things (IoT) landscape is rapidly expanding, with billions of devices connecting daily to facilitate a myriad of tasks across various sectors,

from healthcare and agriculture to smart homes and industrial automation. With global spending on IoT devices skyrocketing from \$749 billion in 2020 to an anticipated \$1,100 billion by 2023, the IoT ecosystem's growth trajectory is undeniable. However, this rapid proliferation comes with its set

of challenges, predominantly concerning cybersecurity vulnerabilities. Notably, the infamous Mirai botnet attack in 2016 underscored the susceptibility of IoT devices to compromise, highlighting the dire need for robust security measures. A pivotal aspect of safeguarding IoT ecosystems hinges on timely and secure software updates. Yet, the existing software update mechanisms are fraught with vulnerabilities, including potential attacks on the update process itself, such as manipulated updates or unauthorized access. Furthermore, ensuring the confidentiality, integrity, and availability of software updates remains a daunting challenge, necessitating innovative solutions to address these multifaceted security concerns effectively.

LITERATURE SURVEY:

C. Zhang and R. Greenet al

Author proposes addressing the security challenges inherent in the Internet of Things (IoT) by developing a lightweight defensive algorithm tailored to combat Distributed Denial of Service (DDoS) attacks within IoT environments. Recognizing IoT's pervasive integration into various critical sectors, the author emphasizes the necessity for robust self-management and self-security capabilities within IoT networks. DDoS attacks, characterized by overwhelming servers with a deluge of requests from a network of compromised computers, pose a significant threat to IoT networks, causing network congestion and impairing normal network functionalities. Consequently, the proposed algorithm aims to safeguard IoT networks by analyzing and mitigating

the interactive communication dynamics among diverse network nodes, ensuring uninterrupted and secure IoT operations.

E. Alsaadi and A. Tubaishat et al

Author delves into the intricacies of the Internet of Things (IoT), envisioning a future where everyday physical objects seamlessly connect to the internet, transcending the conventional realm of smartphones and portables. While this transformative vision promises enhanced connectivity and intelligence integration with our environment, it also introduces a myriad of challenges and vulnerabilities. The paper meticulously investigates the emergent features, challenges, and weaknesses stemming from the widespread adoption of IoT, focusing on issues such as denial of service attacks, eavesdropping, node capture, and sensor physical security. Through a qualitative exploratory research design and literature review, the study uncovers vulnerabilities inherent in the distributed architecture of IoT, including the potential hijacking of unsecured network devices for malicious purposes, exploitation of communication channels, and susceptibility to node capture attacks and privacy leaks.

S. Huh, S. Cho, and S. Kim et al

Author proposes leveraging blockchain technology to address the synchronization challenges inherent in large-scale Internet of Things (IoT) deployments. While blockchain initially gained prominence as the underlying technology for Bitcoin, its versatile applications have since expanded to diverse sectors, including finance, security, and IoT. Recognizing the limitations of traditional server-client models in

managing vast IoT device networks, the author advocates for a blockchain-based approach to facilitate seamless communication and configuration among IoT devices. Specifically, the proposed system utilizes Ethereum's smart contract capabilities to implement a robust key management system, where public keys are stored on the Ethereum platform, and private keys reside on individual devices. This decentralized approach not only enhances IoT device configuration but also offers a more granular level of system management compared to conventional blockchain platforms, paving the way for scalable and secure IoT ecosystems.

PROBLEM STATEMENT:

Existing Techniques contains many other attacks listed below

IOT manufacturer: malicious IOT manufacturer can send wrong or old updates to get money from IOT owner

IOT owner: this are the IOT owners who purchase IOT from manufacturer and then can generate fake payment to receive updates from the manufacturer as all existing payment are based on traditional HTTP REST API based processing which can be easily hack

Malicious Attacker: In this attack, a malicious attacker aims to interrupt the software notification sent by the manufacturer to IOT owners, causing delays/failures of software updates.

Confidentiality attack: In this attack, a malicious attacker attempts to retrieve the contents of a software update during its delivery process with the

aim of receiving a software update it should not have while also avoiding payment.

Invalid update attack: In this attack, a malicious attacker attempts to send an invalid/damaged software update to IOT devices aiming to damage the functionality of the IOT device.

Roll-back attack: In this attack, a malicious attacker aims to send a valid old software with a known pre-existing vulnerability that the attacker can exploit to damage the functionality of the IOT device

PROPOSED METHOD:

To overcome from above attacks author employing Blockchain based IOT software's updates which consists of following technologies

Blockchain: Blockchain has inbuilt support for data security and verification and store all data as Block or transaction and associate each block with unique hash code. Blockchain verify hash code of each block storage and if hash code does not match then it will detect as data alteration and this verification make Blockchain as tamper proof and impossible for data change. Blockchain can be access by using smart contract which contains function to store or retrieve software updates

CPABE: cipher policy attribute based encryption is applied to encrypt software updates. Each encryption technology is based on key generation for encryption and decryption. This key generation required heavy computation so generating keys for each IOT require more space and computation and this computation can be reduced by applying CPABE algorithm. CPABE

uses list of allowed users or IOT as input and for all users generate single keys and the users or IOT who are in list are allowed for data decryption. So manufacturer add list of paid IOT and then generate keys and only those IOT can receive and decrypt software updates.

Elliptic Curve Digital Signature Algorithm

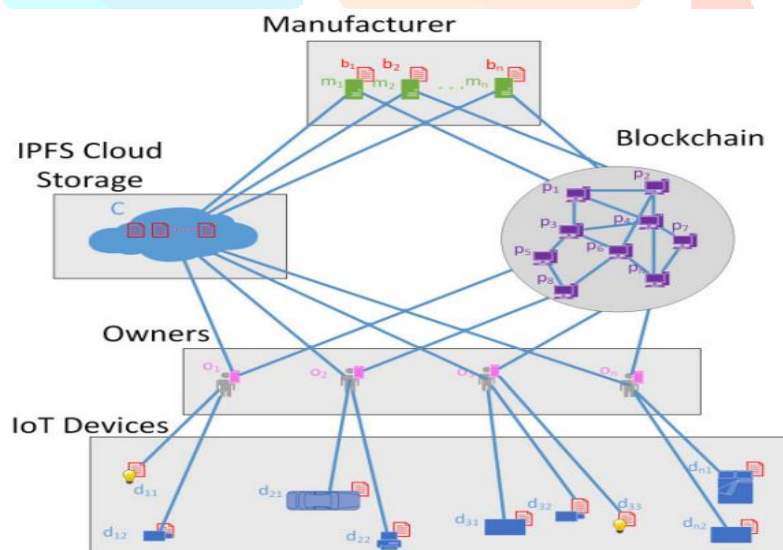
(ECDSA): software updates sent by manufacturer must be checked for attacks so ECDSA will generate hash code on software update and sent to IOT and then IOT will also generate hash code on received updates and if generated and received updates match then IOT consider received software is attack free. If received data alter by attacker then different

hash code will be generated and attack will be detected.

IPFS storage: Interplanetary File System is used to store all encrypted software updates sent by Manufacturer. As Blockchain contains small amount of memory and its storage is also very costly so author using IPFS for software update storage. Blockchain will encrypt software update and then send to IPFS for storage and this IPFS will send back address of store data which will be used by Blockchain to locate that store data in future.

So by using above technologies we can avoid all attacks in the IOT software updates network.

ARCHITECTURE:



METHODOLOGY:

User Management: The foundation of the system lies in its user management capabilities. Upon accessing the platform, users can register either as Manufacturers or Owners. All user data, including usernames, passwords, contact details, and user types, is securely stored on the blockchain. This

decentralized approach ensures data immutability, reducing the risk of unauthorized access and data tampering.

Software Update Upload: Manufacturers play a pivotal role in the system by uploading software updates through a user-friendly web interface. To enhance security, the software files undergo

encryption using CP-ABE (Ciphertext-Policy Attribute-Based Encryption) before being uploaded to the InterPlanetary File System (IPFS). This process involves splitting the uploaded files into multiple blocks based on size, with each block encrypted individually. Subsequently, the hashes of these encrypted blocks, along with relevant metadata, are stored on the blockchain, facilitating seamless tracking and verification of software updates.

Payment Processing: The system also incorporates a transparent and secure payment processing mechanism. Owners can purchase software updates directly from registered Manufacturers. All payment-related information, such as owner and manufacturer names, payment amount, and transaction date, is recorded on the blockchain. This ensures full transparency, auditability, and trustworthiness of the payment transactions, mitigating the risks associated with traditional payment gateways.

Viewing and Verification: Both Manufacturers and Owners can access their respective interfaces to view comprehensive details about software updates, payments, and individual blocks. To maintain data integrity and authenticity, the system utilizes verification hashes, ensuring that any modifications to the software updates can be easily detected and addressed.

Blockchain Transactions: The system's core functionalities heavily rely on blockchain technology. Software update information, including encrypted block details, is meticulously saved on the blockchain using the 'saveDataBlockChain' function. To enhance clarity and streamline data

management, different types of data, such as user details, software updates, and payment records, are stored separately on the blockchain.

Encryption and Decryption: The system prioritizes data security by implementing robust encryption and decryption mechanisms. CP-ABE encryption and decryption functions are employed to safeguard software update files. Furthermore, the system ensures secure storage of public and private keys, further fortifying the encryption process.

Smart Contract Development: Smart contracts, written in Solidity, form the backbone of the system. These contracts encapsulate the business logic governing user management, software updates, and payment processing. Utilizing tools like Remix or Truffle, these smart contracts are compiled and deployed on the Ethereum blockchain, ensuring seamless execution of predefined functionalities.

System Architecture: The system is architecturally structured around three main components: a Django-based web application serving as the backend, the Ethereum blockchain for smart contract deployment and data storage, and IPFS for decentralized file storage. This decentralized architecture ensures high availability, fault tolerance, and scalability, making it ideal for large-scale software update deployments.

Technologies Used: The system leverages a blend of cutting-edge technologies to deliver its functionalities:

Django: A Python-based web framework powering the backend development.

Web3.py: A Python library facilitating interactions with the Ethereum blockchain.

IPFS: A decentralized file storage system ensuring secure and efficient file storage.

ECIES: Elliptic Curve Integrated Encryption Scheme employed for encryption and decryption.

Solidity: The smart contract language used for Ethereum blockchain development.

Frontend Development: On the frontend, user interfaces are meticulously designed using HTML, CSS, and JavaScript. Django's robust templating engine is utilized to render dynamic web pages,

seamlessly integrating user interactions with backend functionalities.

Testing and Deployment: Prior to deployment, rigorous testing is conducted to ensure the system's reliability, security, and performance. Unit tests are performed for backend functions and smart contracts, while integration tests validate the seamless interaction between system components. Initially, the application is deployed on a test network for preliminary testing. Upon successful validation, the finalized application is deployed on a production network, ready for real-world usage.

RESULTS:

```

C:\Windows\system32\cmd.exe
Accounts:
(0) 0xee551ce9aaee8679048b7adf357720fe4f598f
(1) 0x92f95bfa32d466e9fbc70456f00b9eef3a18fb55
(2) 0x5e6ae5f142fbbc59b2fcd60e156a63d95d5dc84b
(3) 0xaaad1c25917d9ef0dca524d052dbb5053b2c04088
(4) 0x96bcc492e746eaa9312fe184e79cab3e418323d
(5) 0x7ef739bd1d85e32762e024a9b90886cb7a05bd9
(6) 0xe434eee435e74b69e226066d4a1cbb589a9da902
(7) 0xdba49b5968d63974e342832630298beac69fd565
(8) 0x6a9c1fbfc895b452e54c929eb25eb440c7197309e
(9) 0xba18598e2eff8d3a0261d268a898c15fd50a303d

Private Keys:
(0) 445fbfe2f2c9aa5d89ca6a7d25ea0a4fb6991fab7c1b7a04ece361f1a76bb790
(1) 3346236b1a101aacd1c8248022685cd1e401e9e51092e29202cb1f642d2144e
(2) e3644f2948432915345c73b53de415df8684de85a258726cb4e76884624f4ca
(3) ca6cb49eb54976fd5561cf03809179ac811f3e52258805f293eae772672cc811
(4) 22324840e7d771faaa5f7b893f0609e031dfa8405ba8d1ab1ff522bd0face079
(5) c356080625617981fb651ec3d36d7263a6a9267b97d5830eb28e55ec4d22f30e
(6) 05608492734e2942adc73d9d164094659c1e5105c6be85b91a7205c83c59a821
(7) d1d5dc33030316b4fcdde1e12d66d183bd6e1aa2fda928d4354bf0956e0ad191
(8) 901d2bf92f9ed31799eabc410b4f7879fb41b3a16cf22960f49c8114e65f6db
(9) 5863a7ba6948b171ada3a7a9be2d00d4101a83c5cd9a476e1fdbeac5b1dcf259

Mnemonic: donor cactus say swing bike snap member pulse hockey pyramid school clinic
?? Important ?? : This mnemonic was created for you by Truffle. It is not secure.
Ensure you do not use it on production blockchains, or else you risk losing funds.

truffle(develop)> migrate_

```

In above Blockchain generated default account and private keys

```

Select C:\Windows\system32\cmd.exe
Starting migrations...
> Network name: 'develop'
> Network id: 5777
> Block gas limit: 6721975 (0x6691b7)


2_deploy_contracts.js
-----
Replacing 'SoftwareUpdate'
> transaction hash: 0xf24bfff6588ef92a279d6a4708ade914c4be58ff74816f5da377bc49eea264757
> Blocks: 0 Seconds: 0
> contract address: 0xF1071ad271410500F85BF057968aE484bDBcB1C9
> Block number: 1
> block timestamp: 1693222748
> account: 0xee551ce9aaee8679048b7adf357720fe4f598f
> balance: 99.99889554
> gas used: 552230 (0x86d26)
> gas price: 2 gwei
> value sent: 0 ETH
> total cost: 0.00110446 ETH

> Saving artifacts
> Total cost: 0.00110446 ETH

Summary
-----
> Total deployments: 1
> Final cost: 0.00110446 ETH

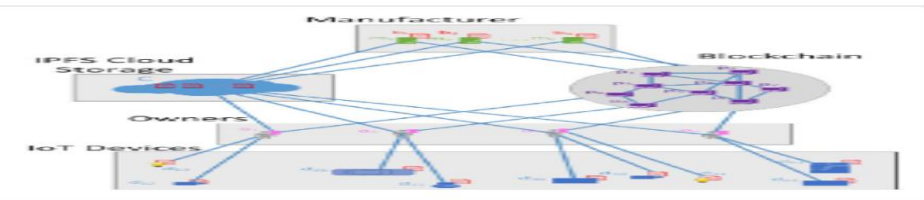
```

In above screen in white colour text we can see 'SoftwareUpdates' contract deployed and we got contract address



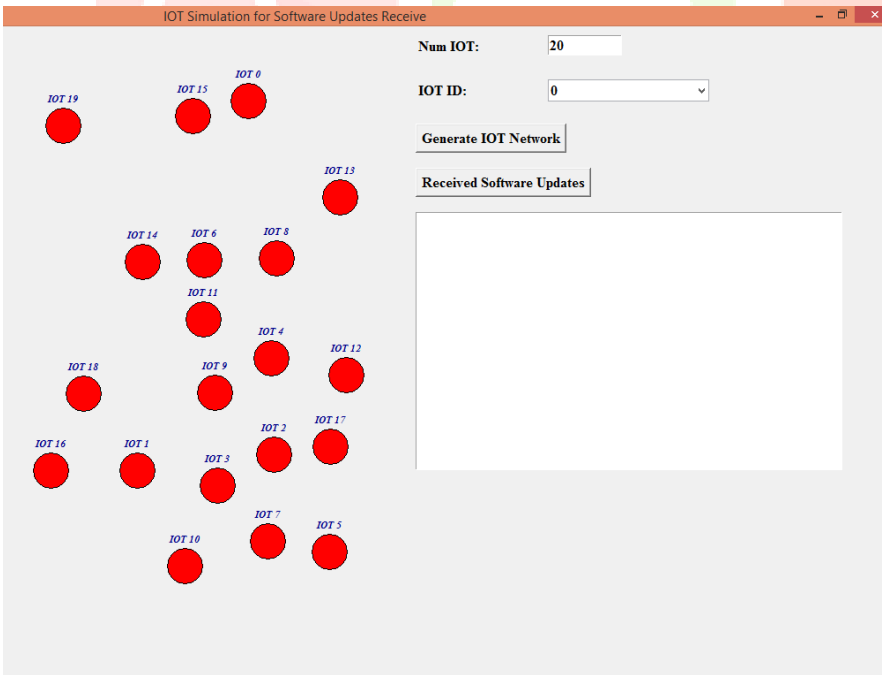
Manufacturer Name	Software Filename	Uploading Date	Software Block Name	Encrypted Block Data	Verif
aaa	googlereview.txt	2023-08-28	googlereview.txt_block_0	b'\x04\x13\x15j<\x1d\xe1\x0f\x90\xa4>\x05j\xbd\xe6:g\x1f\x0e\xce'	QmRjFSnhEQH7VUAwfl
aaa	googlereview.txt	2023-08-28	googlereview.txt_block_1	b'\x04a@*K\xaae\xa2I \xe8F\xde\xea\x9f5	QmdvuiU6hDa9PU2qapF
aaa	googlereview.txt	2023-08-28	googlereview.txt_block_2	b'\x04\xa94\x8c\xae\xbdp\xe2\x04\xaf\x08\x84\x11\x05\x0e'\xc7'\na'\xd8'\x12'	Qmcq9mK5g6UihcmZ4
aaa	googlereview.txt	2023-08-28	googlereview.txt_block_4	b'\x04\xec\xa5\xfbQ\xfb\x19\x9e\x08\xe7+\xe2Q\x05Y\x81D\x01\xa7'\xc2'	QmeyKTdx7ovteAJW1Ax

In above screen from same file multiple blocks are generated and we can see name of file, name of blocks, encrypted content and verification code



Manufacturer Name	Software Filename	Uploading Date	Software Block Names	Verification Hash
aaa	googlereview.txt	2023-08-28	googlereview.txt_block_0 googlereview.txt_block_1 googlereview.txt_block_2 googlereview.txt_block_4	QmRjFSnhEQH7VUAwflKnLLeQ4quHrPGExVtkd5vwU5NWX1p QmdvuiU6hDa9PU2qapH1dYEJ47nKKm9ZohX5KcF5ZetAfd Qmcq9mK5g6UihcmZ4oXTQxQNFerNeYgayMadpyceyXyiG QmeyKTdx7ovteAJW1Ax9PQjJedzoYHPTTruwDj2QfEmnh

In above we can see all files and their blocks

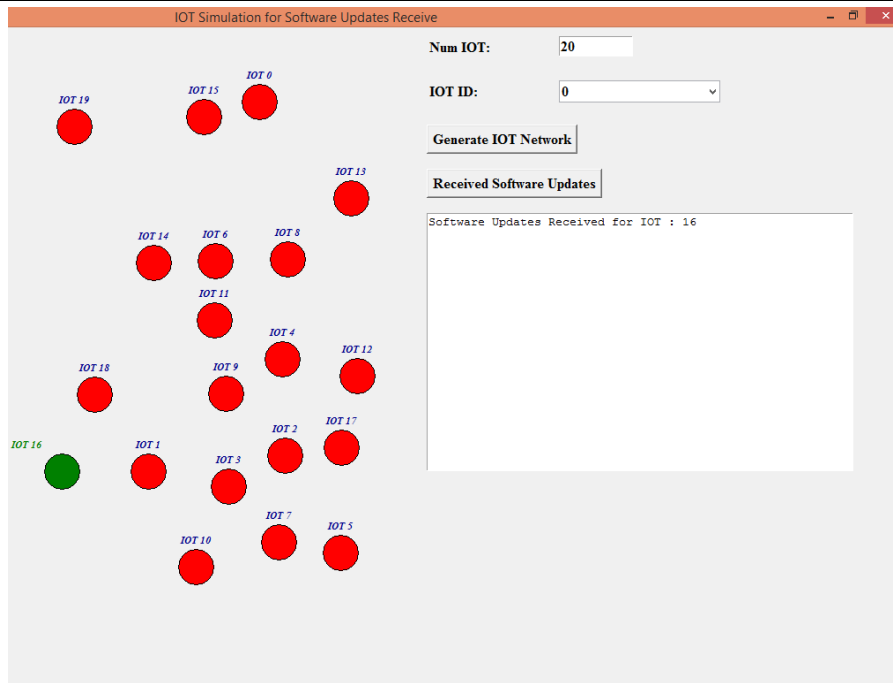


IOT Simulation for Software Updates Receive

Num IOT:

IOT ID:

In above screen all red colour circles are consider as IOT nodes and are placed at different location and now click on 'Received Software Updates' button to download and update software only for those IOT's purchased by owner



In above we can see the IOT for which IOT owner purchase updates will receive and changed its colour to green to indicate as its receiving updates.

CONCLUSION

The Internet of Things (IoT) revolutionizes various sectors, from healthcare to agriculture, by collecting and transmitting data for centralized processing. However, IoT devices are vulnerable to attacks, especially when updating software, which can compromise data integrity and security.

Traditional update mechanisms face numerous challenges, including attacks from malicious manufacturers, owners, and external attackers aiming to disrupt or manipulate the update process. These attacks can lead to significant consequences, such as compromised data integrity, system malfunction, and financial losses.

To address these challenges, the project proposes a comprehensive solution leveraging blockchain technology, CPABE encryption, ECDSA, and IPFS storage. By decentralizing software updates and implementing robust encryption and verification

mechanisms, the system enhances data security, integrity, and availability.

Extension concept further strengthens the system by distributing software updates across multiple IPFS nodes, mitigating the risk of single-point failures and unauthorized access. By adopting these technologies, the project offers a robust and scalable solution for secure and efficient IoT software updates.

REFERENCES:

- [1] S. Poslad, "Ubiquitous computing: Basics and vision," in *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Hoboken, NJ, USA: Wiley, 2011, pp. 1–40.
- [2] F. Wortmann and K. Fluchter, "Internet of Things," *Bus. Inf. Syst. Eng.*, vol. 57, no. 3, pp. 221–224, 2015.
- [3] (Jun. 2019). Global Internet of Things (IoT) Market Size and Forecast To 2026. [Online].

Available: <https://www.verifiedmarketresearch.com/product/global-internet-of-things-iot-market-size-and-forecast-to2026/>

[4] (Dec. 2020). Global Internet of Things (IoT) Market By Software Solution, Report ID 6403. [Online]. Available:

<https://www.verifiedmarketresearch.com/product/global-internet-of-things-iot-market-size-and-forecast-to2026/>

[5] (Jan. 2020). Internet of Things (IoT) in the US. [Online]. Available: <https://www-statista-com.libproxy.scu.edu/study/61733/internet-ofthings-iot-in-the-us/>

[6] (Jan. 2020). Size of the Internet of Things (IoT) in Retail Market in the United States From 2014 to 2025. [Online]. Available: <https://wwwstatista-com.libproxy.scu.edu/statistics/688756/iot-in-retail-market-inthe-us/>

[7] C. Koliass, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other Botnets,” *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[8] L. H. Newman. (2016). The Botnet That Broke the Internet Isn’t Going Away. [Online]. Available: <https://www.wired.com/2016/12/botnet-brokeinternet-isnt-going-away/>

[9] C. Zhang and R. Green, “Communication security in Internet of Thing: Preventive measure

and avoid DDoS attack over IoT network,” in *Proc. 18th Symp. Commun. Netw.*, 2015, pp. 8–15.

[10] Newman. (Oct. 2016). What We Know About Friday’s Massive East Coast Internet Outage. [Online]. Available: <https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/>

[11] (Mar. 2016). Inside the Cunning, Unprecedented Hack of Ukraine’s Power Grid. [Online]. Available:

<https://www.wired.com/2016/03/insidecunning-unprecedented-hack-ukraines-power-grid/>

[12] (Mar. 2016). Analysis of the Cyber Attack on the Ukrainian Power Grid: Defense Use Case. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2016/05/2008_1514/E-ISAC_SANS_Ukraine_DUC_5.pdf

[13] J. Cappos, J. Samuel, S. Baker, and J. H. Hartman, “A look in the mirror: Attacks on package managers,” in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2008, pp. 565–574.

[14] E. Alsaadi and A. Tubaishat, “Internet of Things: Features, challenges, and vulnerabilities,” *Int. J. Adv. Comput. Sci. Inf. Technol.*, vol. 4, no. 1, pp. 1–13, 2015.

[15] S. Huh, S. Cho, and S. Kim, “Managing IoT devices using blockchain platform,” in *Proc. 19th*

Int. Conf. Adv. Commun. Technol. (ICACT), 2017, pp. 464–467.

[16] M. Samaniego and R. Deters, “Blockchain as a service for IoT,” in Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData), 2016, pp. 433–436.

[17] M. Samaniego and R. Deters, “Using blockchain to push software-defined IoT components onto edge hosts,” in Proc. Int. Conf. Big Data Adv. Wireless Technol., Nov. 2016, pp. 110–119.

[18] D. Li, R. Du, Y. Fu, and M. H. Au, “Meta-key: A secure data-sharing protocol under blockchain-based decentralized storage architecture,” *IEEE Netw. Lett.*, vol. 1, no. 1, pp. 30–33, Mar. 2019.

[19] A. Dorri, S. S. Kanhere, and R. Jurdak, “Towards an optimized BlockChain for IoT,” in Proc. 2nd Int. Conf. Internet-Things Design Implement., Apr. 2017, pp. 173–178.

[20] T. Placho, C. Schmittner, A. Bonitz, and O. Wana, “Management of automotive software updates,” *Microprocessors Microsyst.*, vol. 78, Oct. 2020, Art. no. 103257.

