AN OVERVIEW OF PRIVACY SOLUTIONS FOR THE INTERNET OF THINGS

Koyya Doondy Sai Vyshnavi

III[rd] Year, B. Tech in Computer Science & Engineering with Specialization in IoT, Vellore Institute of Technology, Vellore, Tamil Nadu, India.

ABSTRACT

The IoT connects several devices with varying operating systems, processing speeds, and other features. Increased need for security and privacy protection are brought about by the heterogeneity of networks and the pervasiveness of IoT devices. Thus, cryptographic techniques must be robust enough to satisfy these expanded needs while also being efficient enough to be deployed on resource-limited devices. In this study, we provide a comprehensive analysis of how well-known cryptographic methods fare on the kinds of resource-limited gadgets that populate IoT networks. We test various microcontrollers, smart-cards, and mobile devices to see how well they execute symmetric primitives like block ciphers, hash functions, random number generators, and asymmetric primitives like digital signature methods and privacy-enhancing techniques.

Keywords: *Internet of Thing (IoT); Security threat; Privacy Solutions; Humans' Quality of Life (QoL)*

INTRODUCTION

The concept of privacy has been around for a very long time, having been brought up in ancient Greek philosophic conversations. The philosophical, political, sociological, and anthropological discussions around the topic of privacy have all expanded greatly alongside the advent of modern technologies (DeCew, 2018). Constitutions in many nations guarantee citizens the right to privacy, and regulations are in place to limit the release of sensitive data. The European Union's Data Protection Directive (1995) and Data Protection Regulation (2012) are only two examples of the proposed rules, industrial conventions, and privacy agreements that have been discussed and debated on a global scale. Despite attempts at generalization, the concept of privacy remains nuanced when one considers the slight distinctions in personality and cultural background. With the rapid evolution of technology, it's natural to feel uneasy about safeguarding your personal information.

There are two main types of privacy: personal space and confidential data. Data security, the safeguarding of information against tampering while in transit or storage, somewhat overlaps with information privacy, the protection of personally identifiable details. Data may contain private information, so protecting its privacy is equally important to protecting the data itself (Parent, 1983). The proliferation of Internet-connected devices has only heightened these privacy worries (IoT). Exposure of sensitive data can occur in a variety of ways. Information leaks of any kind, especially those involving sensitive data, location, and identity, pose serious risks to privacy in the

forms of monitoring, localization, and personalisation (Porambage et al., 2016). Several types of data mining make advantage of content analysis for committing indirect data violations. When operators of Internet of Things systems (or providers of IoT services) act irresponsibly, it can lead to serious intrusions into users' private lives.

The IoT provides several applications across many industries, such as healthcare, logistics, transportation, and autonomous vehicles, thanks to the convergence of multiple technologies such as cloud computing, artificial intelligence, fifth-generation (5G) networks, and software-defined networks. It allows for a wide variety of permutations and possibilities to capitalize on the connectivity to create coherent and highly functional IoT applications and networks, which in turn generates, collects, distributes, and analyzes a massive amount of personally identifiable information. By 2025, the International Data Corporation predicts that there will be 41.6 billion IoT-enabled devices in use, creating a data volume of 79.4 zettabytes.

This research takes a look at the present state of privacy and security in IoT systems by analyzing data acquisition, storage, transmission, and dissemination. Security mechanisms with confidentiality, integrity, availability, authenticity, and accountability are currently utilized on IoT devices and via IoT networks.

MODELS WITH PROTOCOL LAYERS

The perception, network, and application layers are the usual building blocks of an Internet of Things design. The detecting, transmitting, and processing of data for various IoT gadgets and software is handled by these levels. To gather information about the world around them, devices have sensors that measure things like temperature and humidity. The network layer links the gadgets together so that the collected data can be sent to a central hub, where it can be processed and used to provide useful services to the end users (Zeadally et al., 2019).

In addition, a five-layer design was mentioned as an alternative. The network layer is split up into transport and processing layers in this design. In the transport layer, data travels through various networks, while at the processing layer, it is stored and processed. The processing layer is where features like databases, cloud computing, and big data processing are put into action (Sethi and Sarangi, 2017). Data-driven strategies for achieving organizational objectives are developed at the business layer. Adding this on top of the application layer results in a five-layer design (see Figure 1).

Similar to the two preceding architectures, other IoT architectures organize their layers from physical to application in the same way as the traditional (OSI or TCP/IP) protocol stacks. Some of the current Internet protocols may not be practical for IoT implementation due to the limited power, memory, and high-end microcontrollers found in most IoT devices. Protocols like HTTP and TCP, which are intended to ensure consistency, nevertheless burden networks with unnecessary complexity and inefficient ways of communicating.

Hence, since 2003, many working groups within the Internet Engineering Task Force (IETF) have designed a lightweight communication protocol stack for the limited IoT systems. Figure 1 shows the protocol stack, which consists of IPv6 over Low-Power Wireless Personal Area Network

(6LoWPAN: RFC 6282), IPv6 Routing Protocol for Low power and Lossy Networks (RPL: RFC 6550), and Constrained Application Protocol (CoAP: RFC 7252).

The IEEE 802.15.4 standard is utilized at the physical and MAC layers in the IETF IoT stack; it specifies the behavior of these layers in low-bandwidth, low-cost, low-speed, and low-energy scenarios. The Internet Engineering Task Force (IETF) created the lightweight 6LoWPAN protocol to facilitate the transmission of IPv6 packets over IEEE 802.15.4 networks. To coordinate data flows between nodes on a local network, RPL employs a directed acyclic graph topology with edges directed to their respective destinations and a distance measure between each node and the network's root node as a ranking metric (Palattella et al., 2013). In order to save power, UDP is used for transport rather than TCP. CoAP was created by the IETF's limited RESTful environments working group to meet the needs of constrained environments while also being easily translatable to HTTP for integration with the web. These needs include multicast support, low overhead, and simplicity.

Security and privacy are not top priorities when first building this protocol stack. However, certain issues were brought up: the CoAP specification defines four separate security options (NoSec, PreSharedKey, RawPublicKey and Certificate), IPsec can be implemented over IPv6, and the RPL ranking mechanism can prevent spoofing nodes from becoming parent nodes (Lin and Bergmann, 2016). Subsequently, IETF working groups in the security domain released solutions in the context of a constrained environment, including: (a) the DTLS in constrained environment (DICE) (RFC 7925) provides guidelines for using TLS and DTLS in the IoT system; (b) the access in constrained environments (RFC 7744) defines authentication and authorization mechanisms for the entire life cycle of constrained devices; and (c) the IoT Security Architecture (IoTSA (c). Signatures, a message authentication code, and a way to describe cryptographic keys are only a few of the security methods for the CBOR data format that are defined in RFC 8125, "CBOR Object Signing and Encryption" (COSE) (Morabito and Jimenez, 2020).

Wi-Fi (IEEE 802.11), LoRaWAN (long-range wide-area network), Bluetooth Low Energy (BLE), near-field communication (NFC), radio-frequency identification (RFID), and cellular/mobile connections are all examples of wireless air interfaces used in the Internet of Things (3GPP LTE. NR). These wireless protocols provide pervasive wireless communications that can support a wide range of IoT use cases. Yet, the broadcast nature of wireless networks makes transmission vulnerable to passive and active security assaults, and the characteristics of portable and mobile devices increase this risk.

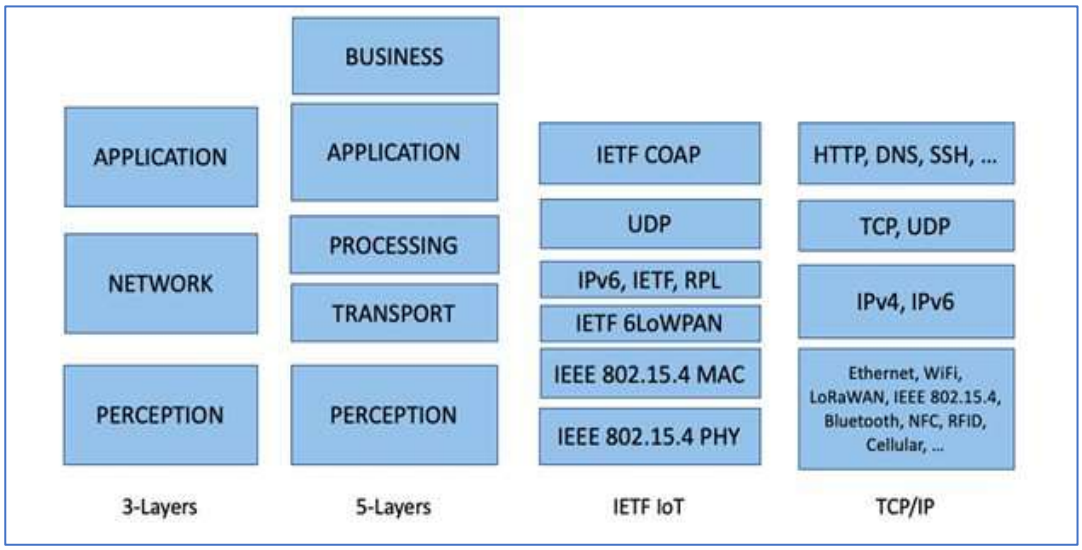

Figure 1: Infrastructures for The Internet of Things and Their Corresponding Protocol Stacks

Measures taken to ensure the safety of devices at the IoT's lower layers fall mostly into two categories: computational security and information-theoretic security (Zou et al., 2016). Communication systems that use cryptographic algorithms often rely on the principal method of protection, computational security. One can choose between symmetric and asymmetric cryptographic schemes (public-key cryptography [PKC]). It is the key that determines how the bits are substituted and transformed in symmetric encryption techniques. The security of algorithms for asymmetric encryption relies on the computational difficulty of solving mathematical issues based on functions like discrete logarithms. When it comes to symmetric encryption and decryption, session keys and PKC are the usual workhorses.

For Internet of Things (IoT) devices that need to keep costs down, power consumption down, and compute load light, this may not be the best option. Additionally, quantum attacks can easily exploit asymmetric techniques. PKC methods are vulnerable to decryption by powerful quantum computers unless key sizes grow to impractically large values (Zhang et al., 2017). As an alternate security approach for IoT devices, physical layer security (PLS) has been examined; PLS was first introduced by Shannon (1949) and relies on information-theoretic proofs of complete secrecy. Channel coding is where the information theory is most often put into practice to protect the integrity of digital communication. By using channel coding, the receiver can identify and fix mistakes generated by transmission impairments including noise, interference, and fading. The 4G and 5G New Radio (NR) turbo codes are a good illustration of this principle; they use polar codes for the control channels and LDPC for the data channels. No matter how sophisticated an adversary's computational attack is, a PLS system should be impregnable from a theoretical standpoint, and enacting one couldn't be simpler.

GENERATION AND DISTRIBUTION OF KEYS

Zigbee

In a trusted Zigbee network, the coordinator (trust center) is in charge of vetting new devices to ensure they are legitimate. To encrypt and decode the general protocol maintenance data and some user data, the trust center delivers a network key to authenticated devices, which is shared by all the nodes in the same network. If two nodes want to encrypt and decrypt their communications at the application layer, the trust center will give a link key for them to use. Each ZigBee device uses a certificate-based key establishment to generate a private key pair (and other security features) that serve as its identification, even if they all run the same ZigBee application. ZigBee 3.0 enables a distributed security network for decentralized security management in addition to the centralized security network.

BLE

Unless actively exchanging data, BLE devices will remain in a sleep state (BLE, 2017). The first step is for two Bluetooth low energy (BLE) devices to verify each other's identities. During the pairing process, the two devices exchange permanent encryption keys. An encrypted link is established between the two devices after authentication, and the necessary keys are shared. The gadgets are considered to be confined if the keys are stored for later reconnection. A connection can be run in a predefined security mode with varying degrees of protection, as specified by BLE.

IEEE 802.11

As part of the initial 802.11 specifications, the wired equivalent privacy (WEP) algorithm was developed by the IEEE 802.11 wireless working group to protect sensitive information (1997). To address these vulnerabilities in WEP, the Wi-Fi Alliance introduced Wi-Fi Protected Access (WPA) in 2003. Wi-Fi Protected Access II (WPA2), often known as robust security networks, is an implementation of IEEE 802.11i, a revision of the original IEEE 802.11 that was adopted in 2004. (RSNs). The authentication, access control, privacy, and message integrity services are defined by the IEEE 802.11i RSN security specification's five phases of operations, which also include the generation and distribution of two types of keys: pairwise and group keys (Stallings, 2017). With the release of WPA3 in 2018, enhanced security features such as a more robust handshake mechanism for establishing connections, coverage for open hotspots, and a larger key size were available.

CONFIDENTIALITY OF DATA

Zigbee and BLE both use 128-bit Advanced Encryption Standard (AES) based encryption for network communication. Two data confidentiality and integrity protocols are defined by the IEEE 802.11i amendment: the counter mode with cipher block chaining message authentication code protocol and the temporal key integrity protocol. In the latter, the cipher block chaining message authentication code is used to ensure the integrity of messages, while in the latter, AES is encrypted using the CTR block cipher mode.

Microcontrollers are often huge and complicated, yet AES can be implemented quickly on them. Moreover, there is no guarantee that the block size is always ideal. An RFID authentication mechanism, for instance, might simply require 64-bit numbers to be encoded (Beaulieu et al., 2015). Several efforts have been made to modify the AES into a lightweight solution suitable for Internet of Things (IoT) applications. ISO/IEC 29192-(1-5) is a set of lightweight cryptography standards that Hogan and Piccarreta (2018) note are NIST-approved. These standards address block ciphers, stream ciphers, asymmetric approaches, and hash functions.

PLS

Researching the channel's physical characteristics for the purpose of implementing more secure encoding and signal processing is the primary goal of PLS. Using Shannon's theory, we can calculate the maximum attainable secrecy rate as a function of the channel characteristics and the encoders' block length. So, the characteristics of the channels via which information travels are related to its safety.

The unpredictability of channels (the characteristic features) are taken into account during key generation on the physical layer through techniques including channel probing, quantization, information reconciliation, and privacy amplification. The procedure is resource-efficient, does not require a third party, and is relatively simple. Hence, in many circumstances, especially with IoT devices, physical key generation can be utilized instead of PKC. Using the communication radio channel and the hardware as individual entropy sources, Shakiba-Herfeh et al. (2020) explored the potential of relocating security core functions (node authentication, message integrity, and message secrecy) down to the physical layer. To secure the IoT network, Zhang et al. (2017) implemented a hybrid strategy that combines physical layer key generation with physical layer encryption, the latter of which carries out encryption operations during the modulation phases of the physical layer.

CONNECTIVITY SOLUTIONS

The majority of IoT networks are cloud-centric solutions, with user data stored and security services centralized on cloud servers. Network problems including high latency, bandwidth bottlenecks, and scalability energy within this paradigm have evolved as the volume of data and the number of connected IoT devices continue to grow at an exponential rate. To address this issue, the concepts of fog computing and edge computing were subsequently introduced into IoT networks, with processing and data storage located close to end-devices. This resulted in the relocation of cloud-based security services like authentication and identification to the fog and edge layers.

Threats to users' privacy posed by the Internet of Things include traffic analysis, eavesdropping, and assaults such as Man-in-the-Middle, Denial of Service, and Distributed Denial of Service. Authentication of devices and key management techniques to avoid compromise of communication channels are crucial security considerations for IoT networks at present. Due to the diverse and scattered nature of edge servers, as well as the limited compute and storage resources available, lightweight security services are required. Moreover, programs often deal with

user identification and authentication and data confidentiality in their own unique ways at the application layer.

For IoT networks in general, we advocate the use of a software-defined network (SDN) because of the flexibility it provides in terms of network setup. Because the control and data planes are separated, SDN controllers can see the entire network, which aids in the detection of dangerous traffic patterns and the creation of SDN applications to perform those jobs. This allows for more efficient use of network resources and increased resistance to outside threats. We classify the Internet of Things (IoT) networks as indicated in Figure 2 as local networks, edge servers, a core network, and cloud servers. Using their IP addresses at the local network layer, a LAN can link together disparate Internet of Things (IoT) gadgets. Each device has a centrally located edge server that it communicates with. To coordinate their SDN domain, which consists of smaller networks managed by the same SDN controller, the nodes at the core network layer act as primary SDN controllers. Access control, authentication, and domain name system (DNS) domains are just a few examples of security and privacy regulations that can be effectively managed by SDN domains. Furthermore, data-chain architectures, such as blockchain, can be implemented either internally to a domain or externally to other domains. A simplified diagram of a software-defined network (SDN) is shown in Figure 2, where SDN domain 1 represents a university campus network consisting of multiple buildings in different locations, all of which are connected to a single SDN controller in the core of the network. In contrast, SDN domain 2 consists of a smart home solution interconnected with yet another SDN controller in the backbone of the network.

RECORD KEEPING AND FILE SHARING

Information gathered or produced locally by IoT devices is often uploaded to a remote cloud storage service (however some information, such as metadata, may be maintained locally at the edge or fog layer). Relational databases, NoSQL databases, and data lakes are all viable options for storing data, but the best option will depend on the data's form and intended use. Users can store their own data on the cloud, and businesses can utilize it to share information and collaborate. Given the current state of affairs, hostile attacks on private data and service provider malfeasance are both very real possibilities.

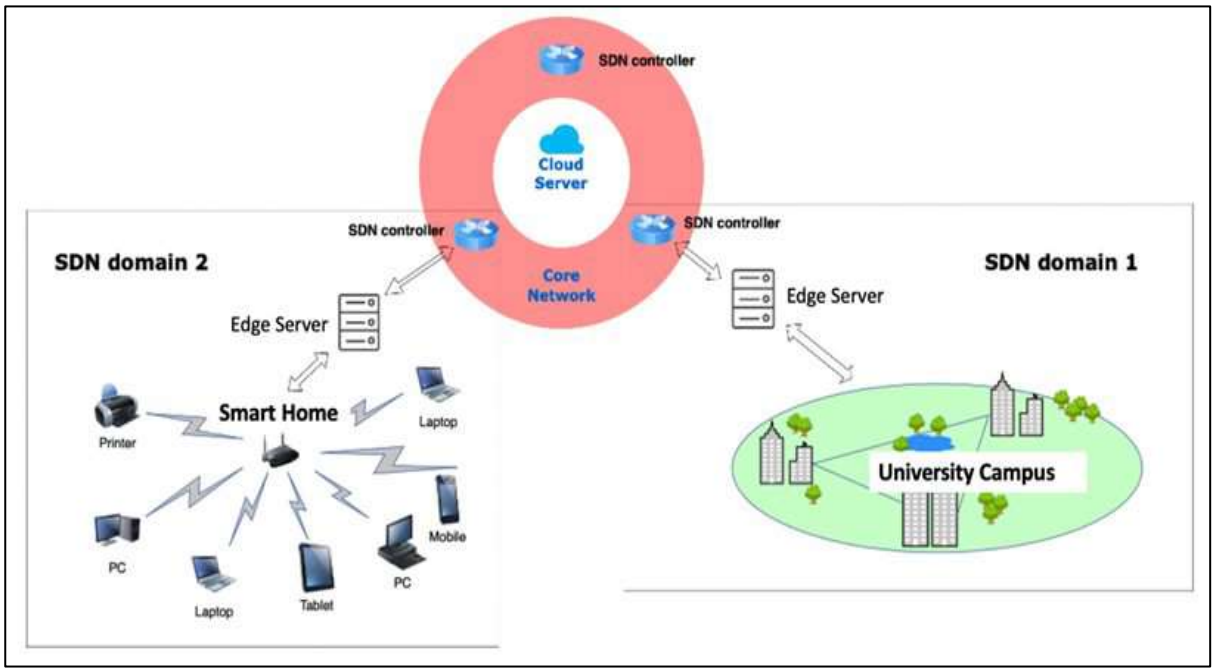

Figure 2: SDN Implemented Example of Internet of Things Network

Data mining methods that allow for information sharing without compromising user privacy, as well as models that can abstract global statistical information, have emerged as necessities of the IoT for making optimal use of the data. Chain structure, a novel data format, has attracted significant attention from both industrial and academic research for addressing additional security challenges related to documenting events/transactions that occur in the network.

The Concept of Differential Privacy

Differential privacy, as defined by Cynthia Dwork in Dwork (2006), is concerned with safeguarding the privacy of individuals within a database while coping with its existing statistical features. As a single random replacement has a negligible impact on the global statistical output in differential privacy thanks to the introduction of noise, privacy is maintained despite the fact that arbitrary inquiries cannot be used to trace any given individual.

An edge-based differential data collection strategy for sensor-cloud systems is demonstrated by Wang et al. (2019). When data is transferred from a local device to an edge server, it is first processed into metadata and residuals. The two pieces of information are encrypted by the edge server (using the AES- Reed-Solomon code). The cloud is where the encoded leftover data are kept. Users have the option of encoding metadata and storing it locally on their device, on the edge server, or on both. By taking this precaution, sensitive information stored in the cloud cannot be stolen or exposed in its raw form. In order to pick the right metadata and figure out the related residuals, two algorithms were used: (1) a small amount of discrepancy between the metadata and residuals is determined by calculating the root mean squared error; (2) using K-means clustering algorithms, a multidimensional data set is partitioned into subsets with similarity, cluster centers are determined as the metadata, and distances from the cluster center to the cluster are minimized as the residuals. Location privacy protection utilizing differential privacy was first developed by

Yin et al. (2018) for large datasets in the industrial internet of things. Using multilevel location data, the authors built a tree structure, identified access patterns, and then introduced noise based on the Laplace distribution. To safeguard users' anonymity while keeping their data accessible, we used a differential privacy protection methodology.

Secure Data Mining

Pattern discovery is the goal of data mining. Location pattern mining, identification mining, and sensitive text context mining are all instances of data mining that commonly violate privacy. Privacy-preserving data mining (PPDM) is an approach to data mining that safeguards personally identifiable information against accidental or intentional disclosure while still allowing for its intended usage (Xu et al., 2014). By making adjustments to the original data, PPDM ensures that data mining algorithms can do their jobs without jeopardizing the privacy of any individuals whose information is included. Perturbation, blocking, aggregation, and swapping are some of the changes. Moreover, many different privacy preservation techniques have been developed for data mining, including K-anonymity, classification, clustering, association rule, distributed privacy preservation, L-diversity, randomization, taxonomy tree, condensation, and cryptography (Sachan et al., 2013). To better manage data in a networked setting, DPPDM techniques can be used. Secure multiparty computation, perturbation, and restricted query are the current DPPDM methods that have been identified as distinct subsets of the larger field (Aldeen et al., 2015).

Linkages and Chains

Blockchain

Simply said, the blockchain is a shared digital record that is updated and maintained in real-time by a network of computers. A cryptographic hash function links the blocks in the ledger, which are timestamped records that also store the hash value of the prior block. Properties such as immutability and append-only are achieved via the cryptographic linking of blocks in the chain. The network's nodes (called "mining nodes") are responsible for creating and adding new blocks to the chain in accordance with a predetermined consensus procedure, and these nodes also construct a copy of the chain for distribution. By design, these characteristics protect the data's integrity and prevent tampering with financial transactions. Blockchains offer novel opportunities for network interoperability due to the decentralized nature with which all transactions are kept and shared by all nodes. Many Internets of Things (IoT) applications have therefore proposed blockchain-based designs to offer increased security and better system interoperability. The most widespread usage is seen in the E-healthcare system, in which authorized medical facilities may easily access and share patients' complete medical histories in chronological order.

Using blockchain in IoT systems, however, still raises privacy concerns. Users doing transactions utilizing pseudonyms on blockchain networks maintain complete anonymity. When a user signs up for the Bitcoin network, they are issued a Bitcoin account, which is defined by an elliptic curve key pair consisting of a public key used to produce a Bitcoin address and a private key used to spend Bitcoin from that address (Herrera-Joancomarti, 2014). Users can maintain their anonymity on the Bitcoin network by opening as many separate Bitcoin wallets as they need for their individual transactions.

Bitcoins are recorded as having been transferred from one address to another in a transaction. While Bitcoin addresses are encrypted, the quantity of Bitcoins being transferred can be seen in the transaction's plaintext. Each transaction is announced to the network, and every user is responsible for validating the transaction to prevent double-sending attacks. And lastly, the blockchain is a permanent record of all transactions. Information like an account's usage history or the history of interactions between two accounts can be revealed. The fact that it may be able to link Bitcoin accounts to a real-world person is the worst part. Given that a user can have several accounts in the blockchain system, the attacker's goal is to pinpoint a group of addresses that all belong to the same user. With Bitcoin's public transaction history, Reid and Harrian (2011) built a transaction network and user network, analyzing both using tools like context discovery and flow analysis gleaned from the outside world. The findings showed that a large number of Bitcoin addresses may be linked together, using external identifiers, and that user activity can be tracked in great detail. In their study, Androulaki et al. (2013) modeled student-to-student Bitcoin transactions at a fictional institution. Based on the findings, it was possible to identify about 40% of the user population.

Holochain

In a holochain network, each node has a copy of all the transaction data, together with the relevant validation criteria and application source code. Like the blockchain, holochain uses a chain structure for data blocks and creates a hash value as the signature for each block to protect the integrity of transaction data and prevent tampering. The hash of the prior block is included in each new block. In contrast to blockchain, where each node in the network has its own copy of the ledger, holochain is replicated on every node and kept as a private source chain. All that a holochain agent keeps track of are the transactions that directly involve him or her. Each node in the source chain keeps track of its own transaction history and sends out transaction headers (and other public data) to a subset of peers at random for verification. Peers will keep the duplicate and distribute it to their neighbors if the transaction was successful. Each node in a holochain network stores a small piece of the DHT1, which is a key/value table containing all valid shard headers. The privacy settings of the hApp are ultimately within the control of the agent using them. When compared to data-centric methods, the personal data is better under control with this agent-centric method (Wahlstrom et al., 2020).

Janjua et al. (2020) offer a fog computing-based architecture for the secure and automated collecting of IoT environment logs. There are three distinct layers to the structure. The log preservation layer utilizes holochain to store the log in fog nodes, guaranteeing the log's integrity and offering resilience to forgery and other attacks. The logs are sent to the fog nodes from the edge nodes, where they are digested and kept locally on the chain, while the actual log files are archived in a separate layer (the cloud). The design in question also allows for logs to be admitted into trusts and for their ownership to be irrefutably shown. By retrieving the digests from the fog node holochains and DHT, the investigator can validate the cloud-stored logs.

CONCLUSION

In today's highly connected society, protecting personal information is more difficult than ever. Increasing numbers of people are worried about their personal data being shared publicly due to the Internet of Things' (IoT's) rapid expansion. Data are gathered from the perceptual layer, sent across the network, and stored in a data storage for all IoT applications. Each step in the process of protecting user privacy in the Internet of Things was examined in this study. Standardized data security methods provide confidentiality at the physical and data link layers of communication during the data gathering phase (such as encryption, authentication and key distribution). Particular encryption, authentication, or key distribution algorithms are offered when security protocols are either not applicable to the IoT devices in question or cannot meet the needs of the application at hand. Hence, safeguarding user anonymity and preventing indirect privacy leaks like the discovery of user behavior patterns are the primary considerations at this time. Users' personal information is protected at every stage of the data transfer, storage, and sharing process by following carefully crafted policies and guidelines.

There are often multiple technologies involved in real-world applications or suggested research solutions, and each one has its own set of privacy problems. Use blockchain for anti-tampering, tacking logs, or enhancing system interoperability; create lightweight encryption/authentication methods for IoT devices/network; employ fog edge structure and cloud storage as network architecture.

The security and privacy solutions for the Internet of Things (IoT) are numerous and often tailored to a particular use case, such as the storing or exchanging of data. There has been a wide range of differential privacy and PPDM implementations across IoT applications. As such, future efforts will benefit from comparing the implementations and summarizing the basics.

While many solutions aim to minimize implementation overhead, doing so runs the risk of creating an IoT ecosystem that is both inefficient and overly complicated. If the rapid evolution of the relevant technology makes it impractical to propose a generalized paradigm, modularization may be an option. Even so, we see that the biggest problem, the disclosure of user information by IoT system owners (or service owners), has not been well addressed. Users have little control over their data and much less insight into how it is used because the infrastructure used to store and transmit it has been built and is the property of the system's owners. Users have little choice but to compromise or risk their privacy when the operation of an IoT service is more vital to the user, such as in some healthcare applications. So, more study in this area is necessary, especially with regard to protecting users' right to privacy when their data is stored and shared.

REFERENCES

1. Alwarafy, A., Al-Thelaya, K. A., Abdallah, M., Schneider, J., & Hamdi, M. (2020). A survey on security and privacy issues in edge-computing-assisted internet of things. *IEEE Internet of Things Journal*, *8*(6), 4004-4022.
2. Ding, W., Jing, X., Yan, Z., & Yang, L. T. (2019). A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion. *Information Fusion*, *51*, 129-144.

3. Li, C., & Palanisamy, B. (2018). Privacy in internet of things: From principles to technologies. *IEEE Internet of Things Journal*, *6*(1), 488-505.
4. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE internet of things journal*, *4*(5), 1125-1142.
5. Li, S., Tryfonas, T., & Li, H. (2016). The Internet of Things: a security point of view. *Internet Research*.
6. Malina, L., Hajny, J., Fujdiak, R., & Hosek, J. (2016). On perspective of security and privacy-preserving solutions in the internet of things. *Computer Networks*, *102*, 83-95.
7. Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *23*(3), 1622-1658.
8. Oleshchuk, V. (2009, May). Internet of things and privacy preserving technologies. In *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology* (pp. 336-340). IEEE.
9. Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Computer networks*, *57*(10), 2266-2279.
10. Ren, H., Li, H., Dai, Y., Yang, K., & Lin, X. (2018). Querying in internet of things with privacy preserving: Challenges, solutions and opportunities. *IEEE Network*, *32*(6), 144-151.
11. Singh, S., Hosen, A. S., & Yoon, B. (2021). Blockchain security attacks, challenges, and solutions for the future distributed IoT network. *IEEE Access*, 9, 13938-13959.
12. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer networks*, *76*, 146-164.
13. Samani, A., Ghenniwa, H. H., & Wahaishi, A. (2015). Privacy in Internet of Things: A model and protection framework. *Procedia Computer Science*, *52*, 606-613.
14. Tawalbeh, L. A., Muheidat, F., Tawalbeh, M., & Quwaider, M. (2020). IoT Privacy and security: Challenges and solutions. *Applied Sciences*, 10(12), 4102.
15. Ziegeldorf, J. H., Morchon, O. G., & Wehrle, K. (2014). Privacy in the Internet of Things: threats and challenges. *Security and Communication Networks*, *7*(12), 2728-2742.