







An Efficient Lightweight Cryptographic Algorithm for IoT Security

Muyideen Abdulraheem , Joseph Bamidele Awotunde^(✉) ,
Rasheed Gbenga Jimoh , and Idowu Dauda Oladipo 

Department of Computer Science, University of Ilorin, Ilorin, Nigeria
{muyideen, awotunde.jb, jimoh_rasheed, odidowu}@unilorin.edu.ng

Abstract. To perform various tasks, different devices are interconnected and interact in several emerging areas. The emergence of the Internet of Things (IoT) and its applications makes it possible in the network for many constrained and low-resource devices to communicate, compute processes, and making a decision within themselves. But IoT has many challenges and problems, such as system power consumption, limited battery capacity, memory space, performance cost, resource constraints due to their small size and protection of the communication network. The traditional algorithms have been slow for data protection point of view, and cannot be used for data encryption on an IoT platform given the resource constraints. Therefore, this paper proposes lightweight cryptography based on the Tiny Encryption Algorithm (TEA) for an IoT driven setup to enhance speed benefit from software perspective rather than hardware implementation. The proposed algorithm was used to reduce the time for encryption in the IoT platform and to preserve the trade-off between security and efficiency. In terms of memory use, execution time, and precision, the proposed work is compared with recent works on lightweight start-ups. Results show that in an IoT driven setup, the algorithm is more secure and efficient, and more suitable for data securing.

Keywords: Tiny encryption algorithm · Cryptography · Communication network · Internet of things · Lightweight cryptographic algorithm · IoT security

1 Introduction

In remote media communication, the Internet of Things (IoT) has been a new technology that is cutting-edge and it is progressing globally [1, 2]. IoT creates network connections that are efficient and very important in uniting data, processes, and people more ever before [2, 3]. Without communicating with the devices, computer hardware, and software, and interrelated artifacts can transfer data over the network, such artifacts are digital machines, sensors, cell phones, RFID tags, and people. IoT consists of millions of embedded devices capable of data identification, analysis, and transmission [1, 2, 4–7].

By the year 2020, it was estimated that almost 50 billion devices will be connected via the Internet [8, 9] and the number will continue to rise exponentially. Considering this volume of devices, privacy, and security of the devices as well as security the data

transmitted is a source of concern to researchers. The operations of IoT are susceptible to insecurity which if compromised threat the security and privacy of its users. IoT security and protection of data need to provide standard and basic principles of security requirements of integrity, confidentiality, authorization, authentication, and availability of users' data. Large volumes of data are transferred every second between those devices. The information transmitted remained a major concern in IoT applications due it network vulnerability [10, 11].

IoT provides different services for different users which can be accessed through users' small devices such as a smartphone. Controlling access is paramount. [12] Proposed the use of Lightweight Biometric Access Control for Remote Authentication Users and Key Access Agreement Scheme for IoT Services. IoT deserves an efficient and effective authentication and key agreement to secure its environment because of its resource constraint. [13] propose lightweight authentication and key agreement protocol to secure WiFi Sensor Network (WSN) of IoT. In their effort to secure IoT devices, [14] proposed Elliptic Curve Cryptography based Authentication Protocol to secure IoT. But to make the mission very difficult, the number of hackers and intruders have been increased recently.

To ensure data transmitted are very secure and keep away from intruders and hackers, different cryptographic algorithms have been employed. The techniques used handling of cryptographic algorithms will determine its effectiveness over handling, defining, and transmitting the secret keys. The algorithm may be technically and functionally working well, but poor handling and maintenance of secret keys make the cryptographic algorithm useless [15, 16]. In the IoT network, two prominent cryptographic algorithms are used namely symmetric and asymmetric algorithms. The same key is used for both encryption and decryption in the symmetric algorithm, while the asymmetric algorithm makes use of two distinct keys called private and public keys. The exchanged of keys within sender and receivers determines the efficiency of any algorithm. The asymmetric algorithm has a better advantage over the symmetric algorithms due to different keys used for the sender and receiver since the private key is protected and not transmitted through the network.

The network is used to forward the public key that will be used to encrypts sender plaintext by the receiver, thus sends ciphertext results to the recipient through the network. Since the hidden key cannot be received by the hacker, even if the public key is received, he/she will not be able to read the message. The receiver will use their private key to decrypt ciphertext. The symmetric is very easy to use and deploy when compare with asymmetric algorithms [17–19]. Therefore, symmetric algorithms are always employed in IoT network implementation to protect information transmission. The algorithm is very easy to implement, reliable as long as the key remains secret and useless overhead resources. This paper, therefore, proposes a tiny efficiency encryption algorithm for IoT-based devices for more effective use. Similar to several other encryption algorithms, the proposed algorithm requires less time to encrypt and decrypt, resulting in increased security and efficiency for all modern IoT network applications. Figure 1 revealed various Internet of Things devices.

The remaining sections of the paper are arranged as follows: Sect. 2 details the design of IoT. Section 3 looks at the framework for Cryptography Tiny Algorithms. Section 4



Fig. 1. Internet of Things with different devices

deals with IoT protection and privacy, while Sect. 5 and Sect. 6 deals with experimental results and discussion, and conclusion.

2 Security and Privacy of Internet of Things

Different attacks are the target of this layer with diverse techniques employ by attackers. One of the attacks of this layer is node capture in which an attacker gets access to control important node and obtain communication details of the sender and receiver nodes. The information stored in the memory could be altered or changed completely. An attacker can intercept secret communication in real-time in an unauthorized manner and steal information transmitted over the network. Such an attack is an eavesdropping attack common in the perception layer. In the replay attack, an attacker can playback communication that occurred in a network to access authentication information on the network. The attacker spy to obtain authentication details from the sender and later use it to communicate with other authentic users in the network. Since the authentication information is genuine, other users will accept the message as authentic.

The Internet of Things (IoT) is an emerging global Internet-based information architecture that facilitates the exchange of goods and services in global supply chain networks. For example, the lack of certain goods would be reported to the provider automatically, which in turn causes electronic or physical delivery immediately. The architecture

is based, from a technical point of view, on data communication devices, mainly RFID-tagged items (Radio-Frequency Identification) [20]. The IoT aims to provide a stable and reliable IT-infrastructure to enable the exchange of “things” [21].

The identified technological architecture of the IoT affects the protection and privacy of the concerned stakeholders. Privacy involves the concealment of sensitive information and the right to monitor what happens to it [22]. The right to privacy can be treated either as a human right, fundamental and inalienable or as a personal right of possession [23];

Consumers may not be aware of the attribution of tags to objects, and there may not be an auditory or visual cue to draw the user’s attention. Individuals can also be tracked without even thinking about it, leaving their data or at least signs of it in cyberspace [24, 25]. Further aggravating the problem, it is no longer only the State that is interested in collecting the data, but also private actors like marketing companies [26].

Peer-to-Peer (P2P) networks are another tool for improving security and privacy, which usually demonstrate strong scalability and efficiency in apps. These P2P systems may be based on DHT (Distributed Hash Tables). However access control must be executed at the actual EPCIS itself, not on the data stored in the DHT, as none of these two prototypes provide encryption [27, 28]. Insofar, it is fair to believe that encryption of the EPCIS link and customer authentication could be enforced without significant difficulties, using the security mechanisms for the Web and the web service [24]. In particular, customer authentication can be done through the issuance of shared secrets or the use of public-key cryptography [29, 30].

Man-in-the-middle (MiTM) and denial of service (DoS) attacks are major assaults in the network layer. In MiTM attack, an attacker intercepts, modifies or removes a communication between authentic sender and receiver and then forwards it to the receiver as sources. However, in DoS attack, authentic sender and receiver are denied access to the network and network resources by persistently making an unnecessary request and therefore keeping the network constantly busy. This would make it impossible for authentic users to access the network.

Instead, the framework layer isn’t spare. One of the attacks on the application layer is cross-site scripting, in which the attacker inserts malicious script into a message from the sender to the receiver. By so doing attackers can alter the message or replace the message to the receiver and later use the original message for an illegal action. Malicious code is another attack in the application layer where self-activation code is inserted and it automatically requests the user for some information. Such information is subsequently forwarded to the attacker.

[31] proposed a Machine Learning approach in securing the IoT network and discussed a comprehensive account of IoT architecture, as well as different approaches of an attack on the IoT network. [32] proposes the use of elliptic curve cryptography ECC to provide security of IoT devices at the Network layer of the IoT architecture. The conventional security algorithms in existence cannot cater for the security of IoT devices as a result of their small size, low battery capacity, and speed. [33] propose a modified lightweight cipher to address the security issue in IoT devices.

3 Methodology

Wheeler and Needham, in 1994, developed the Tiny Encryption Algorithm (TEA) as a lightweight block cipher solution for securing wireless communication. TEA is based on a Feistel structure. It encrypts 64 bits of data using a 128-bit key schedule. To provide non-linearity, it employs XOR, ADD and SHIFT operations for secured communication rather than using P-boxes and S-boxes to achieve diffusion and confusion respectively. It uses the mixed algebraic technique used for IDEA but in a simple way making it faster to implement and take less memory space. It is considered resistant to differential cryptanalysis and in about six rounds, it achieves complete diffusion.

The basic process of TEA is extremely simple and easy to understand. The main inputs of TEA are essentially a plaintext block P and a transfer key K . The plaintext P is split into two halves: $left[0]$ and $right[0]$ while $left[64]$ is ciphertext C . Every half of plaintext P is used to encrypt the other half over 64 processing rounds, and combined to create ciphertext block.

The TEA structure criteria involve splitting a 128-bit key into four 32-bit keywords and splitting the block size of each encryption into two 32-bit terms as well. In encryption rounds, TEA uses a Feistel scheme known as F , where two Feistel operations and many variations, bitwise XOR and SHIFT, are used in one round of Tea.

3.1 TEA Encryption Process

In the encryption method, the 64-bit plaintext P is divided into two input halves of 32-bit each LP and RP , respectively as left plaintext and right plaintext. The 128-bit key to the encryption process is divided into four $K0$, $K1$, $K2$, and $K3$ subkey parts. Every subkey in each round is used as an input to TEA encryption.

The original 32-bit half RP is left-shifted 4-bit, and then the output is added to the first subkey $K0$. From memory, the result is stored as $RP1$. The original 32-bit half RP is again added to a decimal value of the Golden Ratio constant 2654435769, and the output is stored as $RP2$ in the memory. The 32-bit half R is the next 5-bit right-shifted, adding the result to the $K1$ subkey and saving the result as $RP3$.

XOR operation is then performed on $RP1$, $RP2$, and $RP3$ and the final result is then recorded as $RP4$. Furthermore, the value stored in $RP4$ is added to the initial 32-bit half LP and the result is hence stored as $RP5$. This process completes the first round of the half-cycle of TEA.

The second round of half-cycle of TEA starts with the initial stored result $RP5$ being 4-bit left-shifted and the result is then added up with the first subkey $K2$. The results are stored as an $LP1$ in the memory. The stored result $R5$ is again applied in decimal value to a constant of the Golden Ratio 2654435769 and the result is stored as $LP2$ in the memory. The stored result $R5$ is the next 5-bit right-shifted and the result added to $K3$ and saved as $LP3$.

XOR operation is then performed on $LP1$, $LP2$, and $LP3$ and the final result is then recorded as $LP4$. Furthermore, the value stored in $RP4$ is added to the initial 32-bit half RP and the result is hence stored as $LP5$. This process completes the second round of half-cycle of TEA.

That process completes the entire TEA cycle. Thirty-two complete process of TEA encryption is repeated to satisfy the requirement of a full TEA.

$$RP1 = RP \ll 4 + K0$$

$$RP2 = RP + \text{delta}$$

$$RP3 = RP \gg 5 + K1$$

$$RP4 = RP1 \oplus RP2 \oplus RP3$$

$$RP5 = RP4 + LP$$

Equations of first-round encryption

$$LP1 = RP5 \ll 4 + K2$$

$$LP2 = RP5 + \text{delta}$$

$$LP3 = RP5 \gg 5 + K3$$

$$LP4 = LP1 \oplus LP2 \oplus LP3$$

$$LP5 = LP4 + RP$$

Equations of second-round encryption.

3.2 TEA Decryption Process

TEA being a Feistel structure, TEA decryption is nearly the same as TEA encryption with reversed function operation. The encrypted text of RP5 and LP5 starts the decryption process as the input that is now known as right ciphertext RC and left ciphertext LC respectively.

The original 32-bit half RC is left-shifted 4-bit, and the output is then added to the third K2 subkey. The result is stored as RC1 in your memory. The original 32-bit half RC is again applied to a decimal value of 2654435769 that represents the Golden Ratio constant, thus output is stored in the memory as RC2. The 32-bit half RC is the next 5-bit right-shifted, applying the result to the fourth K3 subkey and saving the result as RC3.

XOR operation is then performed on RC1, RC2, and RC3 and the final result is then recorded as RC4. Furthermore, the value stored in RC4 is added to the initial 32-bit half LC and the result is hence stored as RC5. This process completes the first round of the half-cycle of TEA.

TEA's second half-loop round begins with the initial stored result RC5 being left-shifted 4-bit, and the result is then added with the first subkey K0. The result shall be stored as LC1 in the memory. The stored result RC5 is again added in decimal value to 2654435769 as constant of Golden Ratio and stored the result in a memory called LC2. The stored value RC5 is right-shifted next 5-bit and the value is added to K1 and saved as LC3.

XOR operation is then performed on LC1, LC2, and LC3 and the final result is then recorded as LC4. Furthermore, the value stored in LC4 is added to the initial 32-bit half RC and the result is hence stored as LC5. This process ends TEA's second half cycle round.

This process completes the full cycle of TEA. Thirty-two complete process of TEA encryption is repeated to satisfy the requirement of a full TEA. The decrypted ciphertext is then compared with the encryption process's input plaintext to prove both are the same value or message.

$$RC1 = RC \ll 4 + K2$$

$$RC2 = RC + \text{delta}$$

$$RC3 = RC \gg 5 + K3$$

$$RC4 = RC1 \oplus RC2 \oplus RC3$$

$$RC5 = RC4 LC$$

Equations of the first-round decryption

$$LC1 = RC5 \ll 4 + K0$$

$$LC2 = RC5 + \text{delta}$$

$$LC3 = RC5 \gg 5 + K1$$

$$LC4 = LC1 \oplus LC2 \oplus LC3$$

$$LC5 = LC4 + RC$$

Equations of second-round decryption.

Figure 2 displayed the encryption and decryption of the TEA algorithm framework.

The source data can be any form data including structure, semi-structured and structure data and the data can contain text, video, or audio, and both. To access the cipher code data can be sent straight to the TEA encryption algorithm. This data are converted to binaries apart from text data and streams of the ones and the zeros. To access the ciphertext the binary stream is sent to the TEA encryption algorithm using the secret key

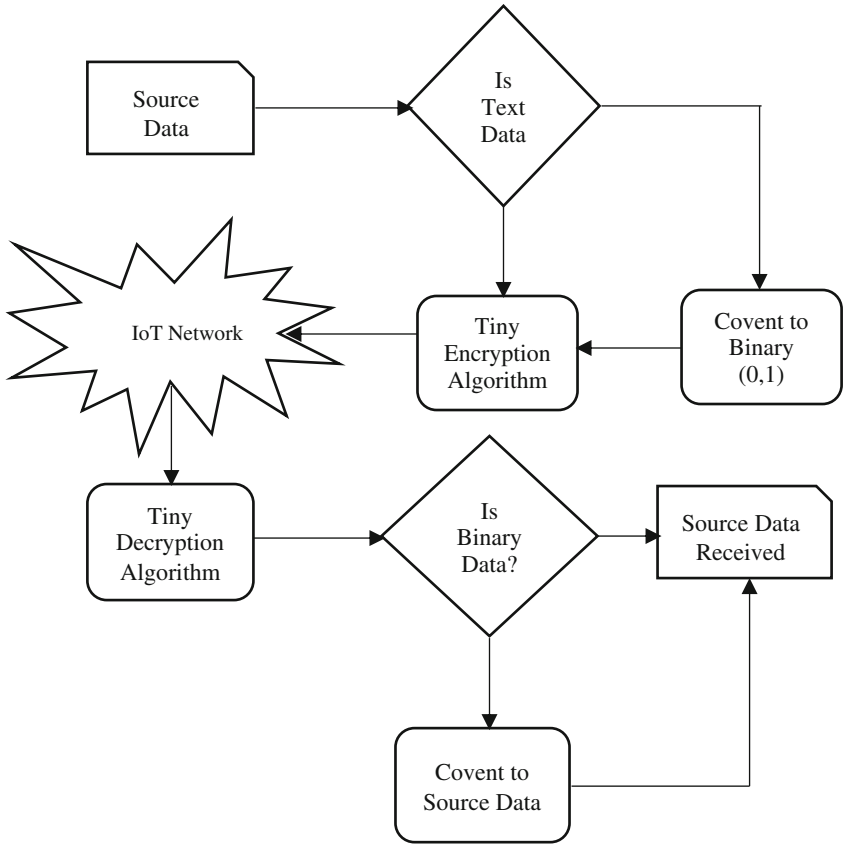


Fig. 2. Framework for TEA symmetric encryption algorithm encryption and decryption

certain by the sender and the receiver. The TEA transforms the ciphertext into plaintext after receiving ciphertext from the sender. The file must be translated to the source file after converted file to zeros and other streams (binary), otherwise, the plaintext will be provided by the sender. Both encryption and decryption of the TEA algorithm are presented in Algorithm 1 and Algorithm 2.

Algorithm 1 TEA Encryption Algorithm

Encrypt (plaintext p, key k):

- 1: Start
- 2: Assign delta = 9E3779B9
- 3: Compute k0, k1, k2, k3 from k
- 4: slip p into 32-bit bock rp and lp
- 5: Assign cycle = 0
- 6: compute rp1 as rp LSHIFT 4 AND k0
- 7: compute rp2 as rp AND delta
- 8: compute rp3 as rp RSHIFT 5 AND k1
- 9: compute rp4 as rp1 XOR rp2 XOR rp3
- 10: assign rp4 AND lp to rp5
- 11: compute lp1 as rp5 LSHIFT 4 AND k2
- 12: compute lp2 as rp5 AND delta
- 13: compute lp3 as rp5 RSHIFT 5 AND k3
- 14: compute lp1 XOR lp2 XOR lp3
- 15: assign lp4 AND rp to lp5
- 16: Increment cycle by 1
- 17: Repeat step 6 through step 16 until cycle = 32

Algorithm 2. TEA Decryption Algorithm

Decrypt (cipher c, key k):

- 1: Start
- 2: Assign delta = 9E3779B9
- 3: Compute k0, k1, k2, k3 from k
- 4: slip c into 32-bit bock rc and lc
- 5: Assign cycle = 0
- 6: compute rc1 as rc LSHIFT 4 AND k2
- 7: compute rc2 as rc AND delta
- 8: compute rc3 as rc RSHIFT 5 AND k3
- 9: compute rc4 as rc1 XOR rc2 XOR rc3
- 10: assign rc4 AND lc to rc5
- 11: compute lc1 as rc5 LSHIFT 4 AND k0
- 12: compute lc2 as rc5 AND delta
- 13: compute lc3 as rc5 RSHIFT 5 AND k1
- 14: compute lc1 XOR lc2 XOR lc3
- 15: assign lc4 AND rc to lc5
- 16: Increment cycle by 1
- 17: Repeat step 6 through step 16 until cycle = 32

4 Experimental Results and Discussion

TEA's output is evaluated and compared with the latest start-of-the-art methods. A network Low-power Wide Area Networks (LPWAN) was used to test the performance of the algorithm with Sigfox and IoT infrastructures. LPWAN was specifically designed for M2M and IoT devices to enable low power consumption and wireless connectivity over long distances. Sigfox offers a very advantageous battery life, power, and cost. In embedded devices, the TEA algorithm was implemented. The experimental setup used system architecture close to [1, 34, 35]. For the experiment, the text files contained in a cloud service database were used. The text files are then encrypted separately in four different situations using the TEA algorithm, and the algorithm's encryption and decryption times for various file sizes and key sizes were calculated (Table 1).

Table 1. Encryption time for a key size of 48 bits

FILE SIZE (Kilobytes)	Encryption time (MM)
0.82	0.121
1.65	0.216
12.32	0.893
36.50	2.014
50.2	3.142
100.7	5.461

Table 2 shown the results of encryption time in mm, the TEA achieved a lower time for various sizes used and the following results were obtained 0.121, 0.216, 0.893, 2.014, 3.142, and 5.461, the encryption time of the files is increasing as the size of the files keep on increasing and the is the normal time in IoT platform when compared with other algorithms.

Table 2. Decryption time for a key size of 48 bits

FILE SIZE (Kilobytes)	Encryption time (MM)
0.82	0.120
1.65	0.215
12.32	0.891
36.50	2.012
50.2	3.138
100.7	5.459

From the results obtained in Table 3, the TEA achieved a lower decryption time for various text file sizes of 0.120, 0.215, 0.891, 2.012, 3.138, and 5.459, the encryption

time of the files is increasing as the size of the files keep on increasing and the is the normal time in IoT platform when compared with other algorithms.

Table 3. Comparison of the proposed system with an existing method

Algorithm	Execution time		ROM	
	Clock cycles	Time gain (%)	(bytes)	ROM gain %
[3] SIMON	108901089	28%	1752	2.9%
[3] Optimized SIMON	86293887		1712	
TEA	62,546,057		1683	

For each encryption code, an MSP-cycle-watcher was used to count the CPU cycles, and serve as a quality-checking tool for measuring the CPU cycles. The encryption part is calculated by subtracting the cycle number within the code from the entire cycle code. The comparison of the encryption cycles by approximation is assumed that it will lead to a reasonable study of the power consumption. Also, adding a checkpoint to decide the start and endpoint of executing encryption cycles was achieved for counting the encryption cycles in this stage.

The results of the comparison of execution time and, ROM consumption are shown in Table 3. The TEA (48/72) absorbs by around 28 percent fewer clock cycles than SIMON (32/64) and SIMON (48/72). Also, the number of TEA execution cycles (48/72) is lower than SIMON (48/72) and SIMON (48/72) is optimized by 2.9%. The SIMON is versatile and appealing to IoT and multimedia applications. It provides a variety of block and key sizes and TEA also has almost the same characteristics as it offers a different set of key sizes and thus TEA algorithm was used for lightweight IoT security cryptographic.

5 Conclusion

In a lightweight block cipher, both the design part and implementation part go simultaneously which has revealed some significant limits and inherent conditions. Efficiency is a critical part of the Internet of Things but in an IoT architecture edge node devices also have resource constraints such as memory and power. This paper presented an IoT protection Implementation TEA algorithm. The development continued from interesting features found in the TEA algorithm and attractive results were obtained. The result indicates improved efficiency as compared to SIMON and modified SIMON with a minor reordering within the process based on execution time and memory usage. The TEA with 48/72 block models reveals an improvement percentage of 28% and 2.9% respectively in execution time and memory computation. Thus, TEA showed great efficiency both in terms of execution times for encryption and decryption. An improved TEA algorithm can also be used to boost IoT protection to increase the encryption of compressed files by further. Future work may also include and implement the data transfer algorithm in sensor, for and ad hoc networks.

References

1. Rajesh, S., Paul, V., Menon, V.G., Khosravi, M.R.: A secure and efficient lightweight symmetric encryption scheme for the transfer of text files between embedded IoT devices. *Symmetry* **11**(2), 293 (2019)
2. Singh, S., Sharma, P.K., Moon, S.Y., Park, J.H.: Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *J. Ambient Intell. Human. Comput.* 1–18 (2017). <https://doi.org/10.1007/s12652-017-0494-4>
3. Alassaf, N., Gutub, A., Parah, S.A., Al Ghamdi, M.: Enhancing the speed of SIMON: A light-weight-cryptographic algorithm for IoT applications. *Multimedia Tools Appl.* **78**(23), 32633–32657 (2019)
4. Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., Taleb, T.: Survey on multi-access edge computing for the internet of things realization. *IEEE Commun. Surv. Tutorials* **20**(4), 2961–2991 (2018)
5. Ploennigs, J., Cohn, J., Stanford-Clark, A.: The future of IoT. *IEEE Internet Things Mag.* **1**(1), 28–33 (2018)
6. Philip, V., Suman, V.K., Menon, V.G., Dhanya, K.A.: A review on the latest internet of things based on healthcare applications. *Int. J. Comput. Sci. Inf. Secur.* **15**(1), 248 (2017)
7. Deshkar, S., Thanseeh, R.A., Menon, V.G.: A review of IoT based m-Health systems for diabetes. *Int. J. Comput. Sci. Telecommun.* **8**(1), 13–18 (2017)
8. Fink, G.A., Zarzhitsky, D.V., Carroll, T.E., Farquhar, E.D.: Security and privacy grand challenges for the Internet of Things. In: 2015 International Conference on Collaboration Technologies and Systems (CTS), pp. 27–34. IEEE, June 2015.
9. Mahdavinejad, M.S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., Sheth, A.P.: Machine learning for Internet of Things data analysis: a survey. *Digital Commun. Netw.* **4**(3), 161–175 (2018)
10. Bordel, B., Alcarria, R., De Andrés, D.M., You, I.: Securing Internet-of-Things systems through implicit and explicit reputation models. *IEEE Access* **6**, 47472–47488 (2018)
11. Frustaci, M., Pace, P., Aloï, G., Fortino, G.: Evaluating critical security issues of the IoT world: present and future challenges. *IEEE Internet Things J.* **5**(4), 2483–2495 (2017)
12. Dhillon, P.K., Kalra, S.: A lightweight biometrics-based remote user authentication scheme for IoT services. *J. Inf. Secur. Appl.* **34**, 255–270 (2017)
13. Ostad-Sharif, A., Arshad, H., Nikooghadam, M., Abbasinezhad-Mood, D.: Three party secure data transmission in IoT networks through the design of a lightweight authenticated key agreement scheme. *Future Generation Comput. Syst.* **100**, 882–892 (2019)
14. Rostampour, S., Safkhani, M., Bendavid, Y., Bagheri, N.: ECCbAP: A secure ECC-based authentication protocol for IoT edge devices. *Pervasive Mob. Comput.* **67**, 101194 (2020)
15. Wang, B., Zhan, Y., Zhang, Z.: Cryptanalysis of the asymmetric fully homomorphic encryption scheme. *IEEE Trans. Inf. Forensics Secur.* **13**(6), 1460–1467 (2018)
16. Jambhekar, N.D., Misra, S., Dhawale, C.A.: Cloud computing security with collaborating encryption. *Indian J. Sci. Technol* **9**(21), 1–7 (2016)
17. Yassein, M.B., Aljawarneh, S., Qawasmeh, E., Mardini, W., Khamayseh, Y.: A comprehensive study of symmetric key and asymmetric key encryption algorithms. In: 2017 International Conference on Engineering and Technology (ICET), pp. 1–7. IEEE, August 2017
18. Ahmad, S., Alam, K.M.R., Rahman, H., Tamura, S.: A comparison between symmetric and asymmetric key encryption algorithm based decryption mixnets. In: 2015 International Conference on Networking Systems and Security (NSysS), pp. 1–5. IEEE, January 2015.
19. Arogundade, O.T., Abayomi-Alli, A., Misra, S.: An ontology-based security risk management model for information systems. *Arabian J. Sci. Eng.* **45**(8), 6183–6198 (2020). <https://doi.org/10.1007/s13369-020-04524-4>

20. Osho, O., Musa, F.A., Misra, S., Uduimoh, A.A., Adewunmi, A., Ahuja, R.: AbsoluteSecure: a tri-layered data security system. In: Damaševičius, R., Vasiljevičienė, G. (eds.) ICIST 2019. CCIS, vol. 1078, pp. 243–255. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30275-7_19
21. Tripathy, B.K., Anuradha, J. (Eds.) Internet of Things (IoT): Technologies, Applications, Challenges, and Solutions. CRC Press, Boca Raton (2017).
22. Gürses, S., Berendt, B., Santen, T.: Multilateral security requirements analysis for preserving privacy in ubiquitous environments. In: Proceedings of the UKDU Workshop, pp. 51–64 (2006).
23. Campbell, J.: The origins and development of the right to privacy. Edward Elgar Publishing, In Comparative Privacy and Defamation (2020)
24. Weber, R.H.: Internet of Things-New security and privacy challenges. Computer law security review **26**(1), 23–30 (2010)
25. Ben-Daya, M., Hassini, E., Bahrour, Z.: Internet of things and supply chain management: a literature review. Int. J. Prod. Res. **57**(15–16), 4719–4742 (2019)
26. Grubbauer, M.: Assisted self-help housing in mexico: advocacy, (micro) finance, and the making of markets. Int. J. Urban Reg. Res. **44**(6), 947–966 (2020)
27. Fabian, B., Gunther, O.: Distributed ONS and its privacy impact. In: 2007 IEEE International Conference on Communications, pp. 1223–1228. IEEE, June 2007
28. Čolaković, A., Hadžialić, M.: Internet of Things (IoT): a review of enabling technologies, challenges, and open research issues. Comput. Netw. **144**, 17–39 (2018)
29. Malik, M., Dutta, M., Granjal, J.: A survey of key bootstrapping protocols based on public-key cryptography in the Internet of Things. IEEE Access **7**, 27443–27464 (2019)
30. Jiang, W., Li, H., Xu, G., Wen, M., Dong, G., Lin, X.: PTAS: privacy-preserving thin-client authentication scheme in blockchain-based PKI. Future Generation Comput. Syst. **96**, 185–195 (2019)
31. Tahsien, S.M., Karimipour, H., Spachos, P.: Machine learning-based solutions for the security of the Internet of Things (IoT): a survey. J. Netw. Comput. Appl. **161**, 102630 (2020)
32. De Rango, F., Potrino, G., Tropea, M., Fazio, P.: Energy-aware dynamic Internet of a Things security system based on Elliptic Curve Cryptography and Message Queue Telemetry Transport protocol for mitigating Replay attacks. Pervasive Mob. Comput. **61**, 101105 (2020)
33. Chatterjee, R., Chakraborty, R.: A modified lightweight PRESENT cipher For IoT security. In: 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1–6. IEEE, March 2020
34. Wu, F., Wu, T., Yuce, M.R.: An internet-of-things (IoT) network system for connected efficiency and health monitoring applications. Sensors **19**(1), 21 (2019)
35. Odun-Ayo, I., Misra, S., Omoregbe, N.A., Onibere, E., Bulama, Y., Damasevicius, R.: Cloud-based security driven human resource management system. In: ICADIWT, pp. 96–106, March 2017