Lightweight Authentication for IOT Edge Devices

Imane Zerraza
ICOSI Laboratory, Abbes Laghrour University, 40004, Khenchela, Algeria
E-mail: zerraza.imane@ univ-khenchela.dz

Keywords: authentication protocol, edge computing, internet of things (IOT), security.

Received: 12 April 2024

The advent of Internet of Things (IoT) technology has brought considerable advantages to both personal and professional realms. However, the integration of IoT devices into diverse systems has underscored the pressing issue of security. Safeguarding data confidentiality in IoT systems necessitates the implementation of robust security measures, including encryption, authentication, and access control mechanisms. When effectively employed, these measures pave the way for the development of an efficient and secure IoT system, offering substantial benefits to end-users. This paper introduces a lightweight authentication solution tailored for IoT edge devices. Specifically designed for the edge network's large-scale nodes, our proposed solution optimally transmits information under limited bandwidth using lightweight symmetric cryptography, leveraging the chacha20 algorithm for session key establishment. Rigorous protocol correctness analysis using the Scyther tool confirms the superiority of our proposed protocol over alternative approaches, particularly in terms of communication and time costs.

Povzetek: Predstavljena je avtentikacijska rešitev za IoT robne naprave, ki uporablja simetrično kriptografijo s Chacha20 algoritmom. Preizkusi kažejo izboljšano varnost in učinkovitost pri zaščiti podatkov v IoT omrežjih.

1 Introduction

The increasing use of internet of things devices in our life touche different areas like healthcare, smart home, smart grid and others applications [1]. To guaranty the proper management and security of this data requires the use of decentralization. This approach entails the deployment of multiple local computing devices or cloudlets situated in proximity to the data source, thereby mitigating latency and enhancing data security [2]. The data shared among these IoT devices could encapsulate sensitive details that must be safeguarded against malicious entities. Traditional security mechanisms such as authentication and others cannot be adapted to IoT devices due to their constraints in computing performance and available memory [3]. This entails the execution of lightweight and efficient schemes to align with the constraints of IoT devices and safeguard sensitive data against various attacks that may target multiple layers [4]. Given that certain types of attacks such as side channel attacks can have a significant impact on these IoT objects and with a negligible possibility of detection [5], the only viable defense mechanism is to minimize leakage or introduce noise [6]. The contribution of this paper is listed as follows: 1) We propose a lightweight authentication protocol for IOT devices in an edge environment. Our design uses a symmetric cryptography to establish session key using chacha20. 2) We enhance the security against different types of attacks, such as side channel attacks. 3) We evalute the performance of our protocol. The rest of this paper is organized into several sections. Section 2 of the paper will discuss various security and privacy measures for authentication protocols. Section 3 of the paper will discuss an authentication for the Internet of Things.An evaluation of the protocol will discuss in section 4. Lastly, section 5 will conclude the paper.

Table 1: Summary of relevant authentication protocols.

Protocol	Authentication scheme	Adventages	Limitations
[7]	Smart home-based authentication	Suitable for smart home. Resist to DOS attacks.	High computation cost. Cannot resist side channel attack.
[8]	Authentication for wearable devices	Use hash function and Xor operation.	Use centralized server. Cannot resist side channel attack
[9]	decentralized authentication for iot devices	secure communication. Use blockchain technology.	Cannot resist side channel attack.
[10]	Smart home-based authentication	Provide mutual authentication.	Cannot resist side channel attack
[11]	smart city-based authentication.	Apply to device-to-device	Storage weakness keys.

		communication.	High time cost
[12]	Distributed Key Management Authentication	Use hash function and Xor operation.	Cannot resist side channel attack
[13]	IBchain methodology	Use blockchain technology.	-
[14]	Blockchain based IOT network.	Secure communication. Use blockchain technology.	-

2 Related work

In this section will discuss several proposed works aimed at bolstering IoT security through the implementation of authentication protocols represented in table 1.

The authors [7] enforced a sturdy authentication scheme for assimilating IOT applications within smart home surrounding. The initial step, describe how to assign a unique identifier to smart devices for authentication in an IOT network. Secondly, the unauthorized accesses are avoided through the use of the session key.

Wu Fan et al [8] proposed an authentication scheme that consists of three phases: initialization, pairing, and authentication. The scheme requires initiating the communication link between the smartphone and wearable device,subsequently, generating the session key as part of the authentication phase.The proposed scheme suffers from single point of failure attacks because all keys are stored in a centralized server.[9]

Santoso and Vun [10] presented a secure authentication protocol employing ECC for IOT based smart homes. The suggested system utilised a gateway centric AllJoyn framework, offering an improved authentication interface for Android devices.

Li et al [11] introduced an innovative lightweight mutual authentication approach in smart city applications leveraging public key encryption to achieve a balance between communication overhead and efficiency while maintaining robust security.

Rachini and Khatoun [12] addressed the security concerns associated with RFID systems and proposes a scheme that leverages encryption techniques and additional security measures to enhance RFID tag authentication and protect against unauthorized access.

Tanweer Alam [13] presented 'IBchain,' a system combining IoT and blockchain ensuring secure communications across a smart city network.

Rajesh[14]introduced a decentralized IoT network,allowing untrusting devices to interact independently through blockchain technology, ensuring verifiability.

3 Authentication scheme

An architecture and protocol consisting of three phases to secure IoT networks is proposed. The registration phase is used to register edge nodes with a trusted server, the authentication and session key establishing phase is used to verify the identity of nodes, and the establish session key between nodes. This proposed protocol could be beneficial for ensuring a secure, reliable connectivity between IoT devices.

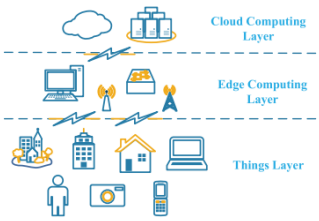

Figure 1: Edge computing architecture.

3.1 Architecture

In this study, we present a novel approach to bolstering the security of IoT networks through the implementation of an infrastructure grounded in edge computing principles, as depicted in Figure 1 [15]. At the core of our framework lies the utilization of edge devices, specifically IoT devices [16], tasked with the responsibility of data collection within the network. These edge devices seamlessly interface with an edge server, serving as the nexus linking disparate edge devices together. The processor, which calle PARAM [17] use as microprocessor. Central to our proposed protocol is the employment of the symmetric Chacha20 algorithm for data encryption purposes. By leveraging Chacha20, renowned for its robust security features and efficiency, we aim to furnish an effective and resilient solution for safeguarding sensitive data traversing IoT networks. The selection of Chacha20 as our encryption algorithm underscores our commitment to striking a balance between security and performance, crucial considerations in the context of resource-constrained IoT environments. Through the integration of edge computing infrastructure and Chacha20 encryption, we envisage fortifying the security posture of IoT networks, thereby mitigating potential vulnerabilities and safeguarding against malicious threats.

3.2 Authentication protocol

Within the realm of cryptography, Chacha20[18] stands out for its capacity to offer robust security measures while being particularly well-suited for resource-constrained devices. In light of these advantages, leveraging Chacha20 for mutual authentication presents a promising avenue for establishing secure secret keys within IoT networks. By implementing this method, a secure connection between

devices can be established, paving the way for the safe and confidential exchange of data. Through the utilization of Chacha20-based mutual authentication, IoT ecosystems can benefit from heightened security measures without compromising on efficiency or performance, thus ensuring that sensitive information remains protected and communication channels remain secure.

3.2.1 Edge node registration phase

• Each device (node or user) can get its ID using blockchain technology. This allows devices to securely exchange data and also creates a secure environment for executing smart contracts. Additionally, blockchain technology helps ensure that the devices remain anonymous while still utilizing the full benefits of distributed ledger technology.

• After receiving the smart contract, the edge server can store it in a secure database. This ensures that the data stored is safe from any malicious activity or unauthorized access. Additionally, the edge server can use various security measures such as encryption, authentication, and authorization to protect the data stored in the database.

3.2.2 Authentication phase

The user first selects a Node and then requests its information. Our proposed authentication scheme works as follows:

1) The user starts the session by requesting a random predefined offset time stamp (off_{TS}) from the server.

2) The user creates a local timestamp variable named T_U, then transmits T_U, off_{TS}, and $h(T_U,K)$ to the Node.

3) When the Node receives the User's variables, it generates its local time stamp T_N.

4) Node compute T_{Noff} ($T_{Noff} = T_N \oplus off_{TS}$), this number can be computed as a random number and it is simple to generate and does not cost the Node a lot of operation as well as the generation of random number.

5) The node calculates the value of E1, which corresponds to its local ID and the value of E2 as specified below:

$E1=h(ID_{node}||T_U||T_{N\,off})$.
$E2=h(K, T_N)$.

Subsequently, it returns to the user the values of E1, E2 and T_{Noff}.

6) Following the data reception from the previous step, the user proceeds by computing T'_N as $T_{N\,off} \oplus off_{TS}$ to derive T_N. Then, it computes E3 where $E3=h(T'_N,k)$.

7) Then it verifies if the difference, $\Delta T = T_U - T'_N$, between its local time and the expected local time of the Node. If $\Delta T > 0$ and $E3 \neq E2$ then it aborts the protocol and it finds that (maybe) there is an attack.

8) The User computes $T_{Uof\,f}$ ($T_{Uof\,f} = T_U \oplus off_{TS}$) in order to hide its local time.

9) Then the receiver computes the value of E4, where $E4=h(ID_{User}||T_{Uoff})$ and sends the values of E1,E4, T_{Uoff},T_{Noff} and $h(T_{Uoff},k)$ to the database server.

10) At this phase, the server generates its local time stamp T_S.

11) The server checks if $\Delta' T = T_S - T'_U$ ($=T_{Uoff} \oplus off_{TS}$) is positive, if yes, then it aborts the protocol and it concludes that there is an attack and the server it not authenticated neither the Node. Otherwise, it continues with the following steps:

a) It Tries to find in its local records (Data) any User ID (ID'_{User}) that verifies the expression: $E5 = h(ID'_{User}||T_{Uoff})$. If $E4 \neq E5$, then it concludes the User is not registered in the local database or there is an attack, in both cases the protocol is aborted. Otherwise, the User is authenticated.

b) Then the server compute $E6 = h(ID'_{node} || T_{Uoff} \oplus off_{TS}|| T_{Noff})$, and it starts the search in its local database to find a Node (ID'_{node}) which verifies the E6 by checking if E1 = E6, if it is not the case, then it concludes that the Node is not registered in the local database or there is an attack, in this case the protocol is aborted. Otherwise, the Node is authenticated successfully and it continues to the next steps.

c) if E1 = E6, then the server computes the following expressions:

$$E7 = h(ID'_{User}||T_s) \quad (1)$$
$$E8 = h(ID'_{node}|| T_s) \quad (2)$$

then, it sends the encryption message ($Enc_k(Ts, E7, E8)$) ; its local time T_S and the values of E7 and E8 to the User.

12) The User receives the message ($Enc_k(Ts, E7, E8)$) from the server. After that, it computes the value of E9 , where $E9 = h(ID_{User} || T_s)$ in order to verify if it is equal to the pre-computed one which received from the server (E7).

13) If E9 = E7, then the User verifies that it is authenticated by the database server, then it sends a message ($Enc_k(Ts, E8)$) to the node. Otherwise, it aborts the protocol.

14) After the user is authenticated, the Node receives the time stamp of the server and the value of E8 from the server. At this phase, the node calculates the value of E10, where $E10 = h(ID_{node} ||Ts)$.

15) If E10 = E8 then the Node is authenticated

successfully and verifies that it is authenticated by the Database server. Otherwise, it aborts the protocol.

3.2.3 Establish session key phase

Using the FTKD for Identity-based Threshold Symmetric Encryption, compute the message specific whole key wk_i to send the session key $(k||Ts||off_{Ts})$. $wk_i = e(H1(id_{Node}), H2(com((k||Ts||off_{Ts}) , v_i)))^s$,all the users will be able to connect to the same node by using the node's identity. Then, the node will access to any message encrypted under its identity[19].

3.3 Security proof with scyther

Scyther [20] emerges as a potent and efficient tool for the thorough examination and identification of potential security breaches and vulnerabilities within security protocols. Its automated analysis capabilities enable comprehensive scrutiny of protocol behavior, effectively assessing its resilience against a spectrum of potential attacks. One of Scythe's standout features, Niagree, offers assurance to communicating parties regarding the secure transmission and correct sequencing of messages, bolstering confidence in the integrity of data exchanges. Furthermore, the inclusion of the Alive feature serves to validate protocol steps, ensuring proper authorization by involved parties and mitigating the risk of unauthorized access. Additionally, Scythe's Weakagree feature acts as a crucial defense mechanism against impersonation attacks, further fortifying the security posture of IoT networks. Through the integration of these advanced features, Scythe facilitates the establishment of a secure and reliable environment for data transmission within IoT networks, instilling trust and confidence in the integrity of communication channels. As depicted in Figure 2. , the findings from Scyther's analysis have confirmed the effectiveness of the security measures integrated into our proposed system. These outcomes indicate a convincing affirmation that our system manifests resilience against well-documented security threats. To summarize, Scyther has demonstrated its invaluable contribution to our research, providing us with the ability to carefully assess the security demands of our authentication protocol, detect possible weaknesses, and improve the overall security stance of our system.

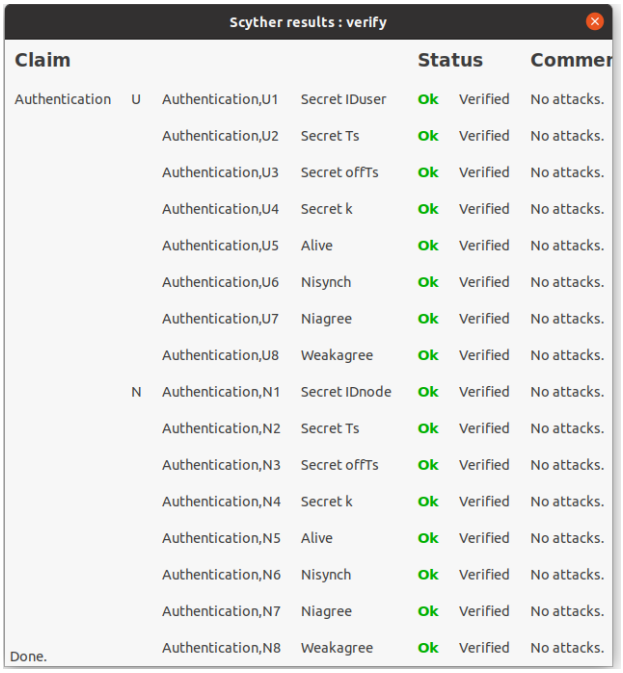

Figure 2: Scyther simulation outcames of our protocol.

3.4 Security analysis of the proposed protocol

We examine the security characteristics in this section. Table 2 presents the results. In the table, we use a checkmark (✓) to indicate that the scheme possesses this security property. Otherwise, a cross (✗) is used.
The comparative analysis demonstrates that the proposed scheme offers superior security compared to the existing authentication scheme in the IoT network as illustrated in Table 2.

Table 2: Security properties analysis

	[7]	[8]	[9]	[11]	[12]	[21]	[22]	Ours
Mutual authentication	✓	✓	✓	✓	✓	✓	✓	✓
Replay attack	✓	✓	✓	✓	✓	×	✓	✓
Perfect forward secrecy	✓	✓	×	×	×	×	✓	✓
MITM	✓	×	×	✓	✓	✓	✓	✓
Side channel attack	×	×	×	×	×	×	×	✓

• Mutual authentication is a way to ensure that a communication party is exchanging messages with an intended party. Since K is a long-term shared value used for mutual authentication, it can be assumed that the two entities have already exchanged the key. Also, the proposed protocol uses double check authentication; help to prevent the raise of this type of attack.
• Secure against the reply attack: the use of commitment and local time in the proposed protocol helps to protect against reply attacks, as the attacker would not be able to send duplicate messages. The commitment ensures that every step is verified, while each new connection needs to use local time in node and an offset time stame generated by the server, user and server. Also, an offset number is used in order to attain maximum security. By

implementing these measures, the system can help to ensure secure data transmission and protect against malicious actors.

• Side channel attacks are a type of attack that can be used to examine cryptographic algorithms and exploit information related to power consumption, execution timing, and electromagnetic fields. The use of the microprocessor with built-in security for power side-channel attacks, call PARAM[15], this makes it difficult for attackers to obtain any useful information from the encryption devices, thus helping to ensure secure data transmission in IoT networks.

•Perfect Forward Secrecy ensures that even if a long-term secret key is compromised, past communications cannot be decrypted. In other words, if someone were to gain access to the private encryption keys used in past communications, they would still not be able to decrypt those communications. In our protocol, even if an attacker gains access to the long-term secret key (pre-shared key k) or other components used to derive the session key, they cannot decrypt past session keys. Hence the perfect forward secrecy can be achieved in this proposed protocol.

• Man-in-the-middle attack: This is a type of cyberattack where the attacker secretly intercepts and relays messages between two parties who believe they are communicating directly with each other. Using a pre-shared key between two entities can help prevent man in the middel attack.This is because the pre-shared key allows both entities to authenticate each other securely before communication begins. Additionally, it is difficult for an attacker to obtain useful information (T_N, ID_{node}) from messages sent to user. On the other hand, the attacker also cannot change the values of variables sent between the user and the Server. As well as the forwarded messages between the server to the user to the node.

4 Performance evaluation

In this section, we conduct a comprehensive examination and comparative analysis of various edge computing protocols, juxtaposed against our proposed protocol, to highlight its superior performance characteristics.

The computation times of these protocols, including our proposed one, are meticulously scrutinized and presented in Figure 3 for clarity and comparison.

Upon scrutinizing the bar chart analysis, a clear trend emerges, showcasing the remarkable efficiency of our proposed protocol. With a computation time of merely 11.5 milliseconds, our protocol significantly outperforms its counterparts presented in references [21] and [22], which exhibit computation times of approximately 13 milliseconds and 41 milliseconds, respectively.

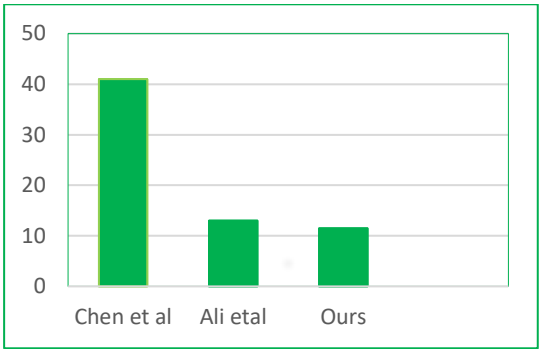

Figure 3: Time cost comparison.

In the comparison of communication costs, as depicted in Figure 4, our proposed protocol demonstrates superior efficiency when contrasted with other related protocols. Notably, the communication cost for our scheme is a merely 1472 bits, comprising 256 bits for Chacha20 encryption/decryption operations, 160 bits for the cryptographic hash function.

In stark contrast, the counterpart protocols presented by Chen et al. [22] and Ali et al. [21] incur significantly higher communication costs, tallying at 3904 bits and 1954 bits, respectively. This substantial variance in communication costs underscores the streamlined and resource-efficient nature of our proposed scheme, positioning it as a frontrunner in edge computing protocol design.

Furthermore, the discrepancy in communication costs serves as a testament to the comprehensive security features embedded within our solution. By minimizing communication overhead while maintaining robust security measures, our protocol not only optimizes resource utilization but also ensures resilience against a wide spectrum of security attributes and potential threats.

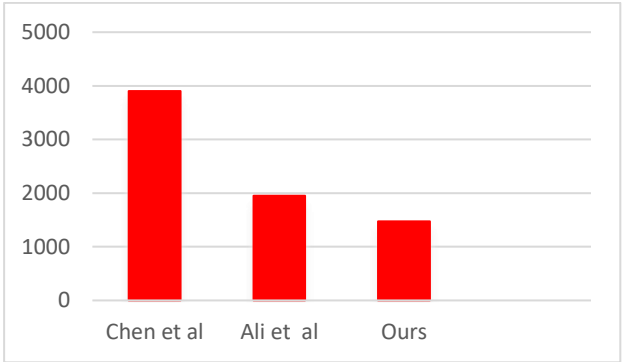

Figure 4: Communication cost comparison.

5 Conclusion

In this study, our proposed authentication scheme has been demonstrated to be highly effective in bolstering the security of IoT systems. The adoption of an edge architecture has played a pivotal role in mitigating network latency and optimizing response times, thereby significantly enhancing the overall performance of the system. Through the utilization of a symmetric algorithm for generating session keys, we have ensured that the

system is fortified with robust security mechanisms, thus fortifying its defenses against potential cyber threats.

The successful implementation of our proposed scheme in the IoT ecosystem underscores the critical importance of developing efficient and secure authentication protocols tailored to address the unique security challenges inherent in IoT networks. To validate the security efficacy of our proposed protocol, we subjected it to rigorous analysis using the Scyther tool, confirming its resilience against various types of attacks.

Furthermore, we conducted a comprehensive comparative analysis between our proposed protocol and other authentication protocols commonly employed in edge computing infrastructure. Our findings reveal that our protocol outperforms alternatives in terms of both computation time and communication cost. Notably, our protocol excels in security measures while maintaining a lightweight design, positioning it as a highly favorable choice within the landscape of edge computing.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, (2013), "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, doi: https://doi.org/10.1016/j.future.2013.01.010.

[2] E. Fazeldehkordi and T.-M. Grønli, (2022), "A Survey of Security Architectures for Edge Computing-Based IoT," IoT, vol. 3, no. 3, pp. 332–365, doi: https://doi.org/10.3390/iot3030019.

[3] M. binti Mohamad Noor and W. H. Hassan, (2019), "Current research on Internet of Things (IoT) security: A survey," Computer Networks, vol. 148, pp. 283–294, doi: https://doi.org/10.1016/j.comnet.2018.11.025.

[4] H. A. Khattak, M. A. Shah, S. Khan, I. Ali, and M. Imran, (2019), "Perception layer security in Internet of Things," Future Generation Computer Systems, vol. 100, pp. 144–164, Nov. 2019, doi: https://doi.org/10.1016/j.future.2019.04.038.

[5] Deogirikar, J., Vidhate, A, (2017),"Security attacks in IoT: A survey". In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), (pp. 32-37). IEEE. http://dx.doi.org/10.1109/I-SMAC.2017.8058363

[6] A. Mosenia and N. K. Jha, (2017), "A Comprehensive Study of Security of Internet-of-Things," IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 4, pp. 586–602, doi: https://doi.org/10.1109/tetc.2016.2606384.

[7] P. Kumar and L. Chouhan, (2020), "A secure authentication scheme for IoT application in smart home," Peer-to-Peer Networking and Applications, vol. 14, no. 1, pp. 420–438, doi: https://doi.org/10.1007/s12083-020-00973-8.

[8] F. Wu, X. Li, L. Xu, S. Kumari, M. Karuppiah, and J. Shen,(2017), "A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server," Computers & Electrical Engineering, vol. 63, pp. 168–181, doi: https://doi.org/10.1016/j.compeleceng.2017.04.012.

[9] U. Khalid, M. Asim, T. Baker, P. C. K. Hung, M. A. Tariq, and L. Rafferty, (2020), "A decentralized lightweight blockchain-based authentication mechanism for IoT systems," Cluster Computing, doi: https://doi.org/10.1007/s10586-020-03058-6.

[10] Santoso, F. K., Vun, N. C. (2015), "Securing IoT for smart home system". In 2015 international symposium on consumer electronics (ISCE) 2015, June, (pp. 1-2). IEEE. http://dx.doi.org/10.1109/ISCE.2015.7177843

[11] N. Li, D. Liu, and S. Nepal, (2017), "Lightweight Mutual Authentication for IoT and Its Applications," IEEE Transactions on Sustainable Computing, vol. 2, no. 4, pp. 359–370, Oct. 2017, doi: https://doi.org/10.1109/TSUSC.2017.2716953.

[12] Rachini, A. S., & Khatoun, R. (2020, February). Distributed Key Management Authentication algorithm in Internet of Things (IOT). In 2020 Sixth International Conference on Mobile And Secure Services (MobiSecServ) (pp. 1-5). IEEE. http://dx.doi.org/10.1109/MobiSecServ48690.2020.9042958

[13] Alam, T. (2021). IBchain: Internet of things and blockchain integration approach for secure communication in smart cities. *Informatica*, *45*(3). Doi: https://doi.org/10.31449/inf.v45i3.3573

[14] Sharma, R. K., & Pippal, R. S. (2023). Blockchain based Efficient and Secure Peer-to-Peer Distributed IoT Network for Non-Trusting Device-to-Device Communication. *Informatica*, *47*(4). DOI: https://doi.org/10.31449/inf.v47i4.3494

[15] L. Kong et al., (2022), "Edge-Computing-Driven Internet of Things: A Survey," ACM Computing Surveys, doi: https://doi.org/10.1145/3555308.

[16] Shah, T., Venkatesan, S., (2018)," Authentication of IoT device and IoT server using secure vaults". In 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE) (pp. 819-824). IEEE.

[17] KF, M. A., Ganesan, V., Bodduna, R., & Rebeiro, C. (2020), PARAM: A microprocessor hardened for power side-channel attack resistance. In 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) , (pp. 23-34). IEEE

[18] Santis, F.D., Schauer, A., Sigl, G. (2017). ChaCha20- Poly1305 authenticated encryption for high-speed embedded IoT applications. In Design, Automation and Test in Europe Conference and Exhibition (DATE), 2017, 2017, Lausanne, Switzerland, pp. 692-697. https://doi.org/10.23919/DATE.2017.7927078

[19] Christodorescu, M., Gaddam, S., Mukherjee, P., & Sinha, R. (2021, November). Amortized threshold symmetric-key encryption. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (pp. 2758-2779). http://dx.doi.org/10.1145/3460120.3485256

[20] https://people.cispa.io/cas.cremers/scyther/ accessed on Mar 16,2024.

[21] A. Shahidinejad, M. Ghobaei-Arani, A. Souri, M. Shojafar, and S. Kumari, (2021), "Light-Edge: A Lightweight Authentication Protocol for IoT Devices in an Edge-Cloud Environment," IEEE Consumer Electronics Magazine, pp. 1–1, doi: https://doi.org/10.1109/mce.2021.3053543.

[22] C.-M. Chen, L. Chen, Y. Huang, S. Kumar, and J. M.-T. Wu, (2021), "Lightweight authentication protocol in edge-based smart grid environment," EURASIP Journal on Wireless Communications and Networking, doi: https://doi.org/10.1186/s13638-021-01930-6