International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 10-No.4 (2024) pp. 1475-1485
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Modelling the Hybrid AES-HMAC-DHKE Approach for Secure TFTP in M2M System: A Proof of Concept

Nur Nabila MOHAMED[1]*, Yulianta SIREGAR[2], Nur Arzilawati MD YUNUS[3], Fazlina MOHD ALI[4]

[1] School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, Selangor, 40450 Malaysia
* **Corresponding Author Email:** nurnabilamohamed@uitm.edu.my - **ORCID:** 0000-0001-7446-5247

[2]Department of Electrical Engineering, Faculty of Engineering, Universitas Sumatera Utara, Medan, Indonesia
Email: julianta_srg@usu.ac.id - **ORCID:** 0000-0002-6867-519X

[3]Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, University Putra Malaysia, Selangor, 43400 Malaysia
Email: nurarzilawati@upm.edu.my - **ORCID:** 0000-0002-4957-8247

[4]Center for Software Technology and Management (SOFTAM), Faculty of Information Science and Technology, University Kebangsaan Malaysia, Selangor, 43600 Malaysia
Email: fazlina.mohdali@ukm.edu.my - **ORCID:** 0000-0001-7520-1307

Article Info:

DOI: 10.22399/ijcesen.344
Received : 14 June 2024
Accepted : 19 December 2024

Keywords :

Machine to Machine,
Trivial File Transfer Protocol,
Hybrid Security Approach,
Advanced Encryption Standard,
Diffie Hellman Key Exchange.

Abstract:

The importance of secure data transfer for operational efficiency is undeniable. This study explores a novel approach to securing communication protocols by combining the established Diffie-Hellman key exchange method with the popular AES encryption algorithm. A practical demonstration (Proof of Concept) showcases the effectiveness of this method in safeguarding data against unauthorized access and impersonation during transmission, particularly within the resource-efficient Trivial File Transfer Protocol (TFTP) used by devices with limited capabilities. This research proposes an innovative secure file transfer protocol that utilizes multiple cryptographic techniques to strengthen data transmission in Machine-to-Machine (M2M) communication. This enhanced security has the potential to make TFTP a reliable option for managing and upgrading low-cost embedded systems, particularly in environments susceptible to security threats.

1. Introduction

The Trivial File Transfer Protocol (TFTP) plays a pivotal role in Machine to Machine (M2M) technology by facilitating tasks such as data collection between machines, remote firmware upgrades for routers, and remote device booting. However, TFTP is inherently insecure, lacking access control and authentication mechanisms, making it vulnerable to cyber-attacks. Malicious actors can exploit TFTP's weaknesses to impersonate legitimate nodes or launch Man-in-the-Middle attacks [1]. As reported in [2], during data transmission over TFTP, impersonation attacks can occur due to the absence of security measures, allowing intruders to masquerade as legitimate agents and inject malicious code, thereby compromising the entire communication system. Similarly, MITM attacks can occur during firmware upgrades in Access Points (APs), where an attacker intercepts and alters the installation process, potentially embedding backdoors in the bootloader, kernel, or applications, thus compromising connected devices [3].

Despite its vulnerabilities, TFTP offers benefits like reducing communication overhead and costs due to its lightweight, fast, and straightforward design. However, its lack of security features undermines its reliability, particularly as the deployment of M2M embedded systems and IoT-Cloud TFTP servers expands, heightening future threat scenarios. Consequently, adopting a compatible security solution is crucial to mitigate these threats. As the utilization of M2M embedded systems increases [4] and IoT-Cloud TFTP server implementations proliferate, the associated risks are expected to escalate. Thus, implementing a robust security measure is imperative to preempt potential TFTP

mishaps and safeguard against the aforementioned vulnerabilities. The motivation of this work arises from he urgent need to enhance TFTP's security and fortify it against cyber threats. This research uniquely contributes to the field by proposing an advanced key agreement technique that integrates the conventional DHKE technique with AES symmetric encryption scheme. This novel approach aims to mitigate MITM and impersonation attacks in TFTP communications. It is believed that this is the first study to present a Proof of Concept (PoC) for a key agreement technique in TFTP to demonstrate its effectiveness against these vulnerabilities. The security enhancement is expected to improve TFTP's functionality, making it more secure and viable for commercial systems, particularly in cost-sensitive M2M embedded infrastructures. This paper begins by reviewing related works on hybrid techniques and algorithms for enhancing security in M2M services. It then details the development of the proposed hybrid approach, followed by the PoC study and subsequent discussion. The conclusion section underscores the importance of securing TFTP in the evolving landscape of M2M technology.

2. Methods

Cryptography [5,6] is widely recognized as the best technique for ensuring security in many M2M services. It involves the study of diverse security techniques using mathematical and logical principles to protect information. Referring to Singh [7], both the encryption algorithm and the encryption key are crucial for guaranteeing the privacy and completeness of encrypted data. To achieve this, it must be extremely difficult to derive either the encryption or the decryption key using computational power. This difficulty ensures that unauthorized individuals cannot decrypt the information. Different encryption schemes have their own strengths and limitations. Symmetric encryption schemes [8], which use a single key, offer high performance in terms of speed but pose concerns regarding secret key distribution. Meanwhile, asymmetric schemes [9] address the key distribution issue but demand more computational resources due to slower processing speeds compared to symmetric encryption. Cryptographic hash functions [10] generate unique messages to verify data integrity but are inherently irreversible functions. A growing trend in cryptographic security involves the strategic combination of several techniques to achieve a more robust security architecture. This hybrid approach integrates various cryptographic schemes such as hashing functions, symmetric and asymmetric encryption, to fulfill multiple security objectives such as data authentication, integrity, non-repudiation and confidentiality. This layered approach effectively addresses the limitations inherent in single algorithms, providing enhanced security. A review of existing literature explored various hybrid approaches that combine these cryptographic primitives, with a focus on studies that aimed to achieve efficient key agreement and data confidentiality in communication systems [11-13]. A literature review was conducted to identify existing hybrid approaches that combine cryptographic hash functions, symmetric, and asymmetric encryption schemes. Several studies aim to provide efficient key agreement and data confidentiality in communication systems.

This section explores various approaches proposed in existing literature to secure communication systems. A key challenge lies in balancing different security objectives such as data confidentiality, integrity, and efficient key agreement. Studies by Gajra et al. [14] and You et al. [15] prioritize data confidentiality through a hybridization of symmetric and asymmetric encryptions (AES, Blowfish, DHKE and ECC). They achieve efficient key agreement but lack performance analysis or data integrity measures. Meanwhile, Sharma et al. [16] focus on performance by utilizing AES for encryption and ECDH for key agreement. They achieve good avalanche and correlation metrics, but their approach might be vulnerable to certain attacks as mentioned in later works [17]. Dhanalakshmi et al. [18] address the pre-distributed key issue with their Instant Key Generation Mechanism (IKGM) while using RSA and DSA for encryption. This approach offers high malicious behavior detection but lacks details on overall network performance impact. Li et al. [19] propose a solution for database security using a combination of AES and ECC. While achieving faster encryption/decryption, they neglect data integrity by not implementing a hash function. Ravikant [20] focuses on reducing decryption time by comparing hybrid algorithms. Their analysis suggests DHKE-AES (D-AES) and ElGamal-AES (E-AES) as efficient options for both speed and security. Apart from that, several studies have explored methods to enhance overall system security by addressing key agreement, data confidentiality, and data integrity. In a study conducted by Xin [21], the author proposed mixed encryption using the AES scheme to encrypt data and ECC scheme for key agreement. MD5 was also integrated with the methods to form the hybrid security. From the result obtained, the improved ECDH approach reportedly achieved a three times faster execution speed. Rewagad et al. [22]

Table 1. *Summary of Various Hybrid Approaches*

Study	Cryptographic Schemes			Security Properties			
	Symmetric encryption	Asymmetric Key Agreement	Hash function, digital signature	General Key Agreement	Authenticated Key Agreement	Confiden-tiality	Integrity
Gajra [14]	AES and Blowfish	ECC and DHKE		√		√	
You [15]	Hill Cipher	DHKE		√		√	
Sharma [16]	AES	DHKE, ECDH		√		√	
Dhanalakhshmi [18]		RSA	DSA	√			√
Li [19]	AES	ECC		√		√	
Ravikant [20]	DES and AES	RSA, DHKE, El Gamal		√		√	
Xin [21]	AES	ECC, DHKE	MD5	√		√	√
Rewagad [22]	AES	DHKE	DSA	√		√	
Abdelgader [23]	AES, IDEA	RSA	MD5	√		√	√
Proposed work	AES	Authenticated HMAC-DHKE	HMAC		√	√	√

combined three schemes; digital signature, DHKE and AES schemes to serve authentication and confidentiality properties over data stored in the cloud. his approach creates a trusted environment for user data. Abdelgader et al. [23] present a hybrid system with AES, IDEA, and RSA, focusing on data confidentiality and efficient key agreement/distribution through RSA public key cryptography. Table 1 summarizes the hybrid approaches and the security properties they provide. From the review studies, it can be seen that the selection of various cryptographic schemes depends on the security attributes that each system wants to achieve. Majority of the studies wanted to achieve data confidentiality and integrity properties in the IoT and M2M systems as proposed in [18,21,22,23]. In addition, AES emerges as the most popular choice for its efficiency across platforms [14,16,19,20,21,22,23]. The widespread adoption of AES is due to its well-established efficiency in both software and hardware, ensuring its suitability for a broad spectrum of platforms [24]. While several studies demonstrate the efficiency of DHKE-AES combinations for speed and encryption/decryption [14,16,20,21,22], there is a lack of research on combining authenticated key agreement with AES for a more robust solution [17,25]. This approach strengthens data confidentiality and builds an efficient cryptosystem for users. Notably, none of the reviewed studies have combined authenticated key agreement with AES symmetric encryption in a hybrid approach. Some studies concentrate on data confidentiality and general key agreement using asymmetric encryption, but conventional protocols like DHKE are still susceptible to unauthorized access and impersonation [26].

3. Existing Works on TFTP Security

The Trivial File Transfer Protocol (TFTP) is widely used for transferring files in embedded systems due to its simplicity and lightweight nature. However, its lack of inherent security mechanisms poses significant risks, particularly in environments where sensitive data is transmitted. To enhance the security of TFTP, various hybrid security approaches have been proposed, integrating multiple cryptographic techniques and protocols to mitigate vulnerabilities. One effective strategy involves the incorporation of both symmetric and asymmetric encryption methods. Pauzi et al. [27] propose incorporating lightweight symmetric encryption for data encryption alongside asymmetric encryption for secure key exchange, which can be implemented in smart embedded devices. This dual approach not only secures the data being transferred but also ensures that the keys used for encryption are exchanged securely. Similarly, Horvat et al. propose a secure TFTP (STFTP) protocol that extends the existing TFTP by integrating additional security mechanisms, thereby addressing the fundamental security shortcomings of the original protocol [28]. Moreover, the use of hybrid encryption algorithms, such as combining Elliptic Curve Cryptography (ECC) with Advanced Encryption Standard (AES), has been shown to enhance data confidentiality and integrity. Zahid et al. highlight the effectiveness of

this hybrid approach in securing data management processes, which can be adapted to improve TFTP security as well [29]. The integration of ECC for key management and AES for data encryption can provide robust security without imposing significant computational overhead, making it ideal for embedded systems. In the context of Internet of Things (IoT) applications, where TFTP is often employed for network booting, the FLEX-IoT framework demonstrates how to secure TFTP communications effectively. Park et al. emphasize the importance of verifying the identity of IoT devices through a hash chain mechanism, which significantly enhances the attack resistance of TFTP while maintaining low latency during file transfers [30]. This approach illustrates the potential of combining access control measures with encryption to create a more secure file transfer environment. Furthermore, the implementation of anomaly-based detection systems can complement encryption efforts by monitoring encrypted traffic for suspicious activities. Chen and Fan propose a system that analyzes deviations from normal usage patterns to detect potential security breaches, which can be particularly beneficial when applied to TFTP communications [31]. This proactive security measure, when combined with encryption, can create a more comprehensive security framework for TFTP. In conclusion, enhancing the security of TFTP through hybrid approaches that integrate various encryption techniques, access control mechanisms, and anomaly detection systems presents a promising avenue for safeguarding sensitive data in embedded systems. The combination of these strategies not only addresses the inherent vulnerabilities of TFTP but also aligns with the growing need for robust security solutions in increasingly interconnected environments.

4. The Proposed Hybrid Security Approach

This study introduces a robust hybrid security approach that combines several cryptographic schemes to enhance TFTP Client-Server communication. The integration of multiple methods leverages their individual strengths, providing a comprehensive solution that addresses the limitations of relying on a single cryptographic technique. The AES ensures robust data encryption, while Diffie-Hellman Key Exchange (DHKE) improves the key agreement process. Additionally, incorporating HMAC-based authentication within DHKE ensures the integrity of the key exchange. This approach combines the strong encryption of AES with the secure key exchange of HMAC-DHKE, offering both high security and good performance. A significant finding from [27] reveals that the DHKE technique is compatible with the request-response-acknowledge sequence of TFTP, closely mirroring the TFTP's inherent lock-step approac. The following parameters were used throughout this work:

p	Prime number
g	Primitive root modulo p
xa	Client's previous session secret value
X_A	Client's previous session public value
a	Client's secret value
A	Client's public value
xb	Server's previous session secret value,
X_B	Server's previous session public value
b	Server's secret value
B	Server's public value
$S1$	First shared secret
$S2$	Second shared secret
H	Cryptographic hash function SHA256
MAC_A	Client's MAC value (256-bit)
MAC_B	Server's MAC value (256-bit)
K_{AB}, K_{BA}	Final shared secret

Figure 1 illustrates overall TFTP RRQ operation, enhanced with the authenticated key agreement technique based on Yoon's scheme [32], and the data encryption process using the AES encryption scheme. An example of TFTP negotiating the AK2048 option is used to explain the development of the proposed approach. It is assumed that the communicating parties have verified each other's identities (MAC addresses), and the values of p and g are publicly shared. To ensure key security, this protocol relies on ephemeral random values exchanged during communication. Prior to initiating communication, both Client and Server independently generate their previous session private values (xa and xb) to compute their public values (X_A and X_B). These values are distributed offline, ensuring the connection remains secure from internet exposure. The AK2048 option involves exchanging secret parameters using authenticated DH key exchange with a 2048-bit parameter length. In the pre-deployment stage, the Client sends its MAC address (ID_A) and security parameter A (2048-bit) in the header. Upon receiving the request, the Server responds with an option acknowledgment containing its parameters (B and MAC_B). The Client acknowledges by sending an ACK packet. If the verification is successful, the Server encrypts the file using the AES scheme and transmits the encrypted file to the Client. Each cipher block is acknowledged by the Client until the final block is received.

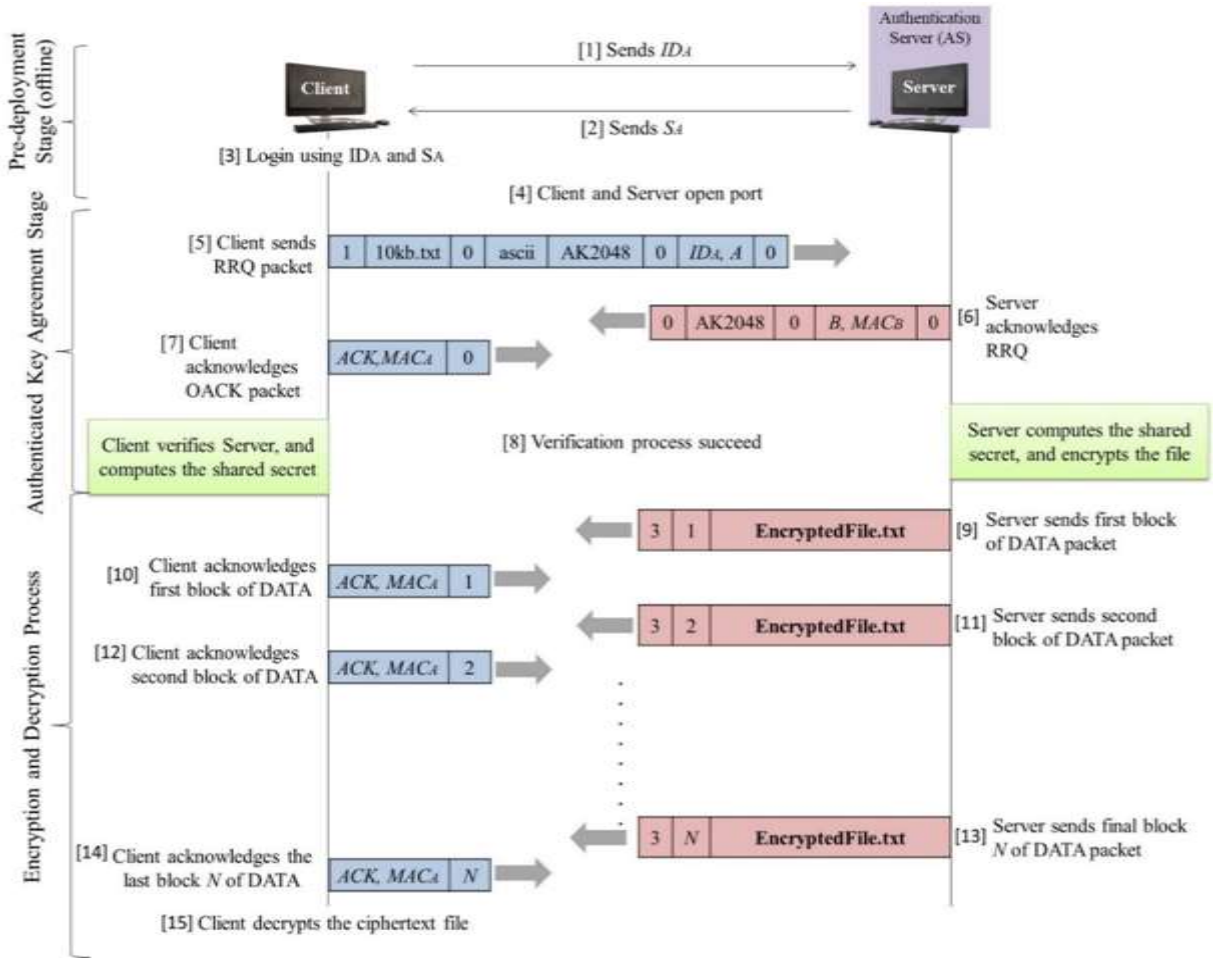

Figure 1. *Proposed key exchange and AES scheme on TFTP*

Client's knowledge	Public knowledge	Server's knowledge
Public: *p, g, A, ID_A, X_A, B, MAC_B, MAC_A* Private: *a, xa, S1, S2*	*p, g, A, B, ID_A, MAC_A, MAC_B*	Public: *p, g, ID_A, X_B, B, A, MAC_B, MAC_A* Private: *b, xb, S1, S2*

Table 2. *The Client, third party and Server's knowledge*

5. The Proof of Concept

In order to demonstrate that the proposed key agreement can mitigate the MITM and impersonation attacks, the PoC analysis was conducted. The parameters known by the Client, Server, and the third party (public knowledge) are presented in table 2. This analysis aimed to confirm that the secure TFTP protocol effectively prevents MITM and impersonation attacks. Table 2 shows that a malicious third party can obtain several parameters: *p, g, A, B, IDA, MAC_A* and *MAC_B*. In the analysis, it was assumed that the MITM or impersonator gained access to the system during the connection between the real Client and Server. The fake Client was able to steal the real Client's identity (ID_A) and used the parameter to attempt to obtain the shared secret. Figure 2 illustrates the possibility of an MITM attack during the key exchange process. MITM attack might occur when malicious entity intercepts the communication by substituting its own public value for those of the communicating parties. For example, when the Client transmits its public value to the Server, the MITM intercepts it by sending its own value *D* to the Server. Consequently, MITM and Server thus agree on one shared key, while the MITM and Client agree on another. After this exchange, the MITM can decrypt any messages sent by the Client or Server and possibly modify them before transmitting to the other party. Based on figure 2, the MITM generated its own value *d* and computed its public parameter *D*, then shared it with the Client and Server. Upon obtaining the Server's public value, it computed *S1'* and *S2'* to be shared with Server. The Server computed *MAC_B'* and sent it to the attacker, while the MITM also

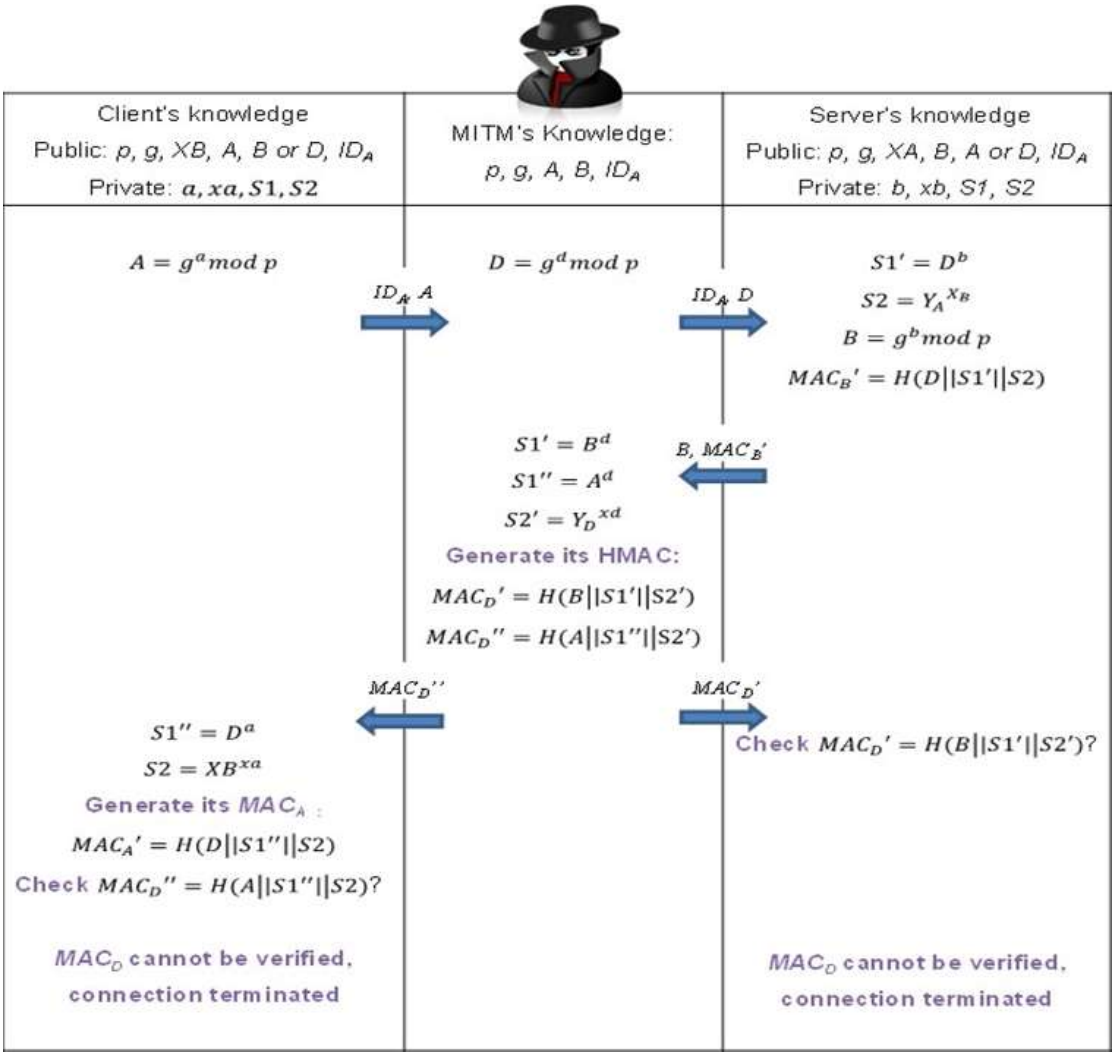

Figure 2. *MITM attack during key agreement process*

computed MAC_D' and shared it with Server. The attacker also sent MAC_D to the Client. The Client then computed *S1"* using the attacker's parameter. However, the computation failed while checking MAC_D' and MAC_D'' values, thus terminating the communication. From this analysis, it is evident that the MITM attack failed because the intruder, although able to replace public values, cannot compute *S2* due to the lack of knowledge of X_A and X_B. The second attack, an impersonation attack, involves an attempt to steal the identity of either the Client or Server, as depicted in figure 3. This attack allows the intruder to impersonate one or both of the legitimate parties and act in their place. In this work, the impersonator aims to obtain the private file from the Server by impersonating the Client and Server. Figure 3 shows that an impersonation attack might occur if the impersonator steals the Client's identity and public value. Without computing anything, the impersonator copies the obtained parameters and sends them to both the real Client and Server. However, the impersonator cannot compute the final shared secret because it lacks knowledge of X_A and X_B. In conclusion, the analysis indicates that MITM attacks are effectively mitigated by the proposed protocol, as the HMAC-DHKE scheme strengthens the key agreement process during initial communication. While an impersonation attack might initially affect the protocol, it cannot succeed in reading or obtaining the final shared secret due to the use of the message authentication code in the protocol.

6. Result and Discussion

Table 3 presents the transfer times for six TFTP protocols on Raspberry Pi 3. The file transfer times captured using Wireshark exclude decryption time. The decryption time was measured by integrating code to compute the decryption process at the client's end. The total transfer time was derived by summing the decryption time with the transfer time recorded by Wireshark, reflecting the complete TFTP protocol operation. The results indicate that the proposed hybrid TFTP_AK3072 protocol has the

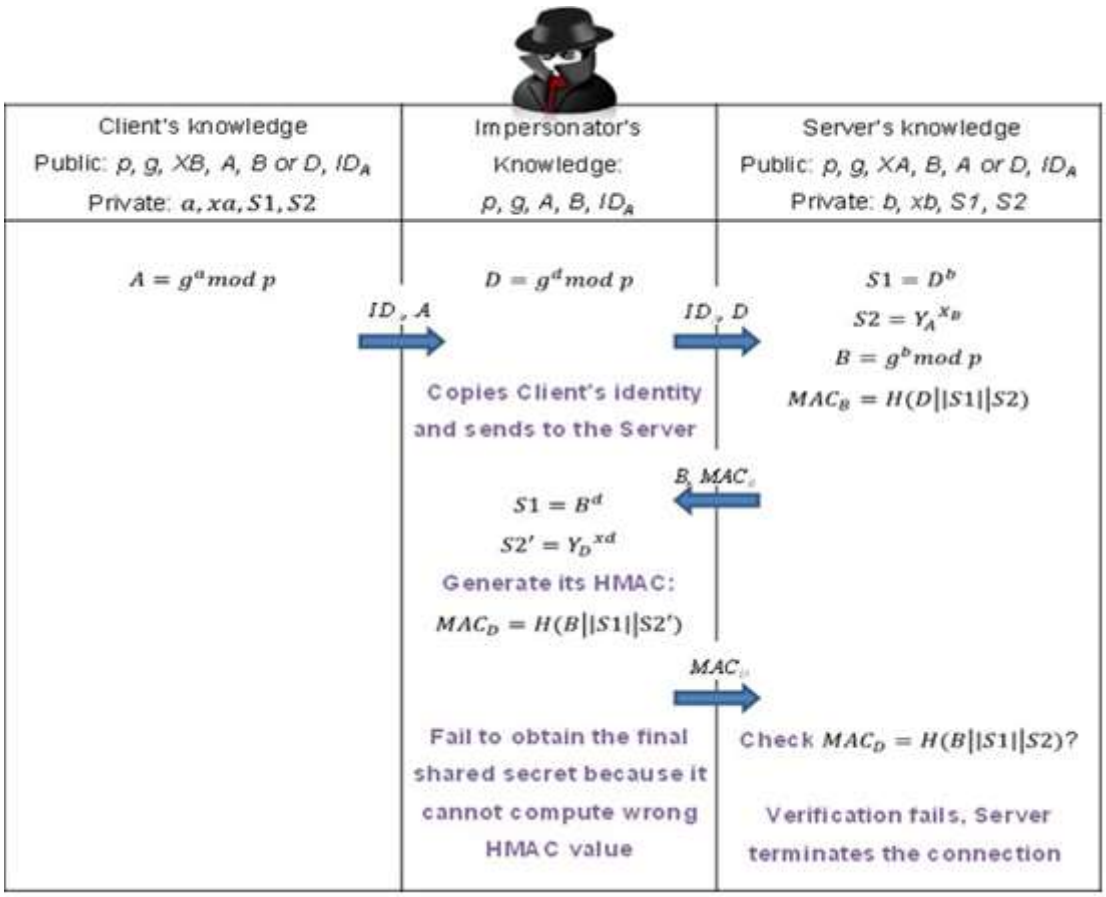

Figure 3. *Impersonation attack during key agreement*

Table 3. *Total File Transfer Time*

Protocols	NORMAL	AES256	DK2048	DK3072	AK2048	AK3072
File Transfer on Wireshark (s)	0.573319	0.788454	1.026571	1.041356	1.027933	1.048567
Decryption Time (s)	NA	0.025891	0.026032	0.026821	0.027631	0.028367
Total File Transfer Time (s)	0.573319	0.814345	1.052603	1.068177	1.055564	1.076934

Table 4. *Security Overhead Computation Among Secure Protocols*

	AES256	DK2048	DK3072	AK2048	AK3072
Initial Handshake (s)	0.170132	0.361999	0.364272	0.372638	0.379610
Enc./Dec. Time (s)	0.066307	0.068761	0.070175	0.069896	0.071315
Security Overhead (s)	0.236439	0.430760	0.434447	0.442534	0.450925

longest transmission time due to the added overhead of the authenticated key agreement and the encryption/decryption process. This demonstrates that extending security within the transmission protocol decreases file transmission performance. Specifically, the proposed secure TFTP protocols (TFTP_AK2048 and TFTP_AK3072) are nearly twice as slow as the standard protocol. Initial handshakes were also analyzed through Wireshark, as shown in figure 4. For the standard protocol, the initial handshake encompasses the time from sending the RRQ packet by the client to the first DATA packet from the server. For secure protocols, the initial handshake involves the negotiation of key agreements and computation of the shared secret, including the transmission of RRQ, OACK, and ACK packets. Figure 4 shows that TFTP_NORMAL has the fastest initial handshake at 0.022472 seconds. The delay in the proposed protocols (TFTP_AK2048 and TFTP_AK3072) is attributed to the computation

of the shared secret during the initial key agreement. Interestingly, TFTP_AES256 also shows a longer initial handshake despite not performing key exchange at the beginning, likely due to security option negotiation. From the result, it also indicates a minor difference of 0.015 seconds among secure protocols. The hybrid security approach (HMAC-DHKE-AES) focuses on efficiency and low latency, although the initial security negotiation introduces some delay. Future performance optimization may explore lightweight encryption schemes like CLEFIA and PRESENT.

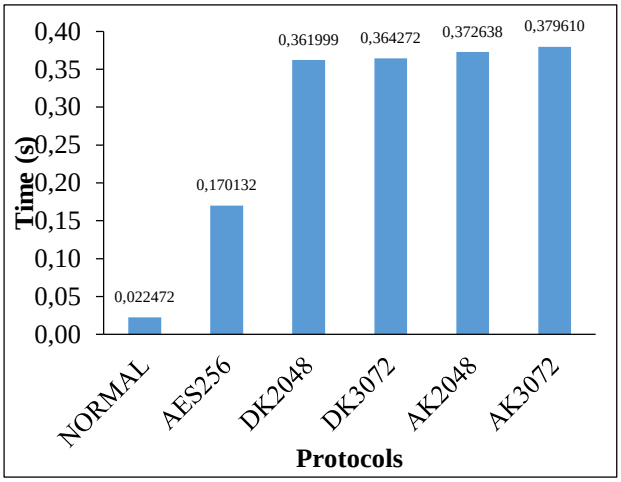

Figure 4. *Initial Handshake for Different Protocols*

The security overhead was evaluated as the time taken for initial key agreement handshake and encryption/decryption processes in TFTP communication. Table 4 shows the security negotiation times for the secure TFTP protocols when transferring 5KB files. The hybrid key agreement and encryption schemes required 0.442534 seconds for TFTP_AK2048 and 0.450925 seconds for TFTP_AK3072. TFTP_AES256, which does not perform key computation during the handshake, showed a faster security negotiation time of 0.236439 seconds, nearly twice as fast as the proposed protocols. Figure 5 highlights that the key agreement handshake constitutes about one-third of the overall operation (35%), while encryption and decryption contribute 7% due to block-level data processing. The higher time difference between the standard and proposed protocols is primarily due to the key computation process during the initial key agreement involving large parameter sizes.

While this study emphasizes empirical performance metrics, such as actual execution time, it also acknowledges the theoretical time complexity of the proposed hybrid security approach. The empirical metrics provide valuable insights into the real-world efficiency of the proposed approach. Despite the promising results, the study identifies limitations, such as computational overhead during security negotiation and dependency on cryptographic key size, which can impact performance. Additionally, while AI-based approaches rely on learning patterns [33], the proposed method uses proven cryptographic techniques, ensuring robust security without vulnerabilities associated with learning-based models. This method is particularly advantageous for resource-constrained IoT devices where reliability and security are crucial [34]

7. Conclusion

This work introduced an enhanced security solution for TFTP communication, combining AES, HMAC and DHKE to create a more secure data exchange framework. While these algorithms are commonly used together in major security protocols such as TLS, IPSec, and SSH, the unique

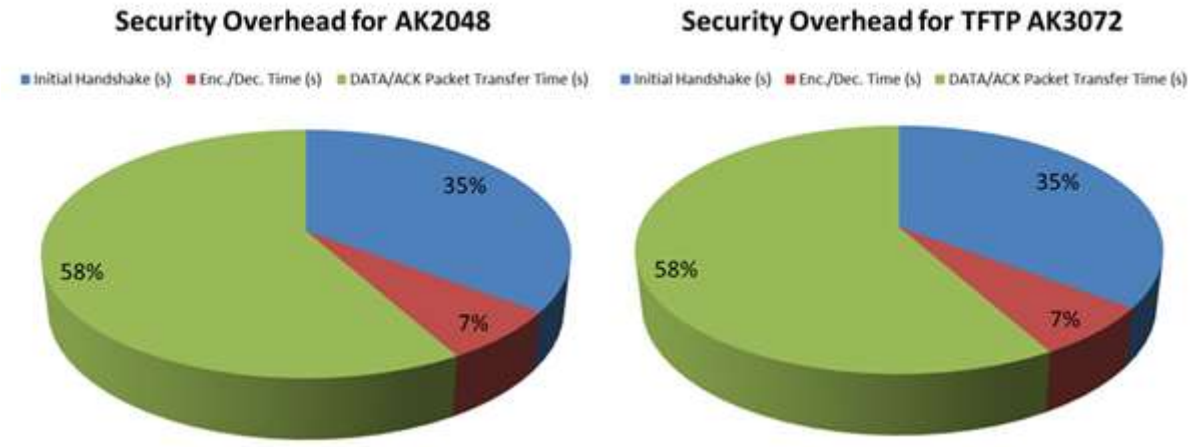

Figure 5. *Security Overhead for TFTP_AK2048 and TFTP_AK3072*

contribution of this study lies in its tailored application for the lightweight TFTP within M2M environments, particularly where resource constraints are critical. Unlike traditional implementations in resource-abundant systems, the proposed hybrid approach was specifically designed to address the performance and energy efficiency challenges of TFTP in IoT and M2M applications. The proof-of-concept demonstrated the viability of this solution, showing that it enhances security against common attacks such as MITM and impersonation, without significantly impacting the protocol's operational efficiency. By focusing on the unique demands of M2M systems, where low latency and minimal computational overhead are essential, this work contributes a novel implementation of established cryptographic techniques for TFTP. This positions the proposed approach as a valuable alternative for securing data in low-power, embedded IoT devices, laying the groundwork for further optimisations and deployment in real-world M2M applications.

The proposed hybrid approach achieves the following security objectives:

- **Authenticated Key Agreement**: Achieved through identity registration during initial communication, allowing the application domain to verify the data of the M2M device and confirm that the data was sent by the correct sender to the correct client. HMAC also provides authentication during the exchange.
- **Confidentiality**: Ensured through the implementation of AES encryption and public key exchange within the protocol.
- **Integrity**: Maintained by the HMAC scheme, which is transmitted alongside the encrypted file, preventing illegal data alteration such as delaying, replaying, and modifying the information.

Despite the existing time delay in the proposed approach, future improvements will focus on reducing both delay and execution time. This will be achieved by implementing alternative lightweight encryption schemes like CLEFIA and PRESENT and conducting a comparative performance analysis against AES symmetric encryption. Additionally, incorporating other extension options within the security header, such as block size or window size options, could enable larger packet sizes for each transmission. Investigating potential packet loss during data transmission is also advisable to ensure complete data reception by the client or server, ultimately improving delay times. This study primarily focuses on empirical performance metrics, acknowledging that the real-world applicability of the proposed approach may vary under different network conditions and deployment scenarios. Future work should explore the adaptability of the approach in diverse IoT environments, providing a comprehensive understanding of its effectiveness in various contexts [35-52].

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors would like to express gratitude to Universiti Teknologi MARA, Selangor, Malaysia for providing funding and invaluable research facilities, academic guidance, and a supportive environment throughout the completion of this study.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] A. Singh, B. Singh, and H. Joseph, (2008). Vulnerability Analysis for FTP and TFTP, in *Vulnerability Analysis and Defense For The Internet*, Springer, Boston, MA, pp. 71–77.

[2] "Hitachi Data Collection Agent," 2013. http://www.hitachi-solutions.co.jp/datacollection/.

[3] Mohammad Salloum C. Group, (2020). Augmenting NFS and TFTP protocols to an Intrusion Detection System. *Department of Electrical and Computer Engineering Queen's University Kingston, Canada*

[4] C. Rajani and S. Ramesh, (2018). A Comprehensive Survey on Exiting Solution Approaches Towards Security and Privacy Requirements of IoT, *Int. J. Electr. Comput. Eng.*, 8(4);2319–2326, doi: 10.11591/ijece.v8i4.pp2319-2326.

[5] G. C. Kessler, (1998). An overview of cryptography, in *Handbook on Local Area Network*, pp. 1–63.

[6] M. Althamir, A. Alabdulhay, and M. M. Yasin, (2023). A Systematic Literature Review on Symmetric and Asymmetric Encryption Comparison Key Size," *Proc. - 2023 3rd Int. Conf. Smart Data Intell. ICSMDI*, pp. 110–117, 2023, doi: 10.1109/ICSMDI57622.2023.00027.

[7] G. Singh and S. Supriya, (2013). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security, *Int. J. Comput.*

Appl., 67)19);33–38, doi: 10.5120/11507-7224.

[8] M. Zakir and H. Sarker, (2005). A Cost Effective Symmetric Key Cryptographic Algorithm for Small Amount of Data," in *IEEE International Multitopic Conference*, pp. 1–6.

[9] Q. Luo and Z. Zhang, (2023) The Secure Data Transmission Method of a Cellular Communication Network Based on the Asymmetric Encryption Algorithm, *J. Commun.*, 18(2);82–88, 2023, doi: 10.12720/jcm.18.2.82-88.

[10] P. Gallagher, (2012) "Secure Hash Standard (SHS)," *Fed. Inf. Process. Stand.* 180-4.

[11] A. A. R. El-Douh, S. F. Lu, A. Elkony, and A. S. Amein, (2022). A Systematic Literature Review: The Taxonomy of Hybrid Cryptography Models, vol. 439 LNNS, no. March. *Springer International Publishing*.

[12] A. A. Abdullatif, F. A. Abdullatif, and S. A. Naji, (2019). An enhanced hybrid image encryption algorithm using Rubik's cube and dynamic DNA encoding techniques, *Period. Eng. Nat. Sci.*, 7(4);1607–1617, doi: 10.21533/pen.v7i4.885.

[13] S. Zainal, R. C. M. Yusoff, H. Abas, S. Yaacub, and N. Megat Zainuddin, (2021)"Review of Design Thinking Approach in Learning IoT Programming," *Int. J. Adv. Res. Futur. Ready Learn. Educ.*, 24(1);28–38, https://akademiabaru.com/submit/index.php/frle/article/view/4204.

[14] N. Gajra, S. S. Khan, and P. Rane, (2014). Private Cloud Security : Secured user Authentication by using Enhanced Hybrid Algorithm," in *International Conference on Advances in Communication and Computing Technologies Private*, pp. 1–6.

[15] W. You, G. Shi, X. Chen, J. Qi, and C. Qing, (2017). Research on a Hybrid System With Perfect Forward Secrecy," in *IEEE InformationTechnology, Networking, Electronic and Automation Control Conference*, 1942;1783–1787.

[16] S. Sharma and V. Chopra, (2017). Data Encryption using Advanced Encryption Standard with Key Generation by Elliptic Curve Diffie-Hellman, *Int. J. Secur. Its Appl.*, 11(3)17–28.

[17] O. Pal and B. Alam, (2017). Diffie-Hellman Key Exchange Protocol with Entities Authentication, *Int. J. Eng. Comput. Sci.*, 6(4);20831–20839, doi: 10.18535/ijecs/v6i4.06.

[18] K. S. Dhanalakshmi, D. B. Kannapiran, and A. Divya, (2014). Enhancing Manet Security Using Hybrid Techniques in Key Generation Mechanism," in *International Conference on Electronics and Communication System*, pp. 1–5.

[19] X. Li, J. Chen, D. Qin, and W. Wan, (2010). Research and realization based on hybrid encryption algorithm of improved AES and ECC," in *International Conference on Audio, Language and Image Processing*, pp. 396–400, doi: 10.1109/ICALIP.2010.5684554.

[20] R. K and U. K. Lilhore, (2016). Combined Cryptographic Standards for Minimizing the Decryption Time of Encrypted Data using E-AES and D-AES, *Int. J. Innov. Res. Comput. Commun. Eng.*, 4(11);19783–19788.

[21] M. Xin, (2015). A Mixed Encryption Algorithm Used in Internet of Things Security Transmission System," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 62–65, doi: 10.1109/CyberC.2015.9.

[22] M. P. Rewagad and M. Y. Pawar, (2013). Use of digital signature with Diffie Hellman Key Exchange and AES encryption algorithm to enhance data security in cloud computing," in *International Conference on Communication Systems and Network Technologies*, pp. 437–439, doi: 10.1109/CSNT.2013.97.

[23] A. M. S. Abdelgader, L. Wu, M. Y. E. Simik, and A. Abdelmutalab, (2015). Design of a Secure File transfer System Using Hybrid Encryption Techniques, in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2.

[24] M. Khan and N. Munir, (2019). A Novel Image Encryption Technique Based on Generalized Advanced Encryption Standard Based on Field of Any Characteristic, *Wirel. Pers. Commun.*, 109(2);849–867, doi: 10.1007/s11277-019-06594-6.

[25] S. G. Sophia and S. Prabakeran, (2016). Efficient and Secure Data Sharing Using AES and Diffie Hellman Key Exchange Algorithm in cloud KCG College of Technology , Chennai , India, *Middle-East J. Sci. Res.*, 24;126–131, doi: 10.5829/idosi.mejsr.2016.24.IIECS.23150.

[26] D. Adrian *et al.*, (2015) Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice, in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 5–17, doi: 10.1145/2810103.2813707.

[27] N. F. S. Pauzi, M. A. M. Isa, H. Hashim, S. F. S. Adnan, and L. Mazalan, "Performance measurement of secure TFTP protocol for smart embedded devices,(2014). *Proceedings, APWiMob 2014 IEEE Asia Pacific Conf. Wirel. Mob.* pp. 144–149, 2014, doi: 10.1109/APWiMob.2014.6920276.

[28] G. Horvat, D. Žagar, and G. Martinović, (2013). STFTP: Secure TFTP protocol for embedded multi-agent systems communication, *Adv. Electr. Comput. Eng.*, 13(2);23–32, doi: 10.4316/AECE.2013.02004.

[29] R. Zahid *et al.*, (2023). Secure Data Management Life Cycle for Government Big-Data Ecosystem: Design and Development Perspective, *Systems*, 11(8);1–18, doi: 10.3390/systems11080380.

[30] G N Kodanda Ramaiah (2022),View of Elliptic Curve Cryptography Based Homomorphic End-to-End Encryption Security in Cloud Computing.pdf. *MathematicalStatistician and Engineering Applications* 71(3);64 –75

[31] K. H. Park, S. J. Kim, J. Yun, S. H. Lim, and K. W. Park, (2021). Flex-IoT: Secure and resource-efficient network boot system for flexible-IoT platform, *Sensors*, 21(6);1–21, doi: 10.3390/s21062060.

[32] E. J. Yoon and K. Y. Yoo, (2009). An efficient Diffie-Hellman-MAC key exchange scheme, in *4th International Conference on Innovative Computing, Information and Control*, pp. 398–400, doi: 10.1109/ICICIC.2009.80.

[33] C. Zhang and Y. Lu, (2021). Study on artificial

intelligence: The state of the art and future prospects," *J. Ind. Inf. Integr.*, 23;1–9, doi: 10.1016/j.jii.2021.100224.

[34] M. Kumar, H. Raj, N. Chaurasia, and S. S. Gill, (2023). Blockchain inspired secure and reliable data exchange architecture for cyber-physical healthcare system 4.0," *Internet Things Cyber-Physical Syst.*, 3;309–322, doi: 10.1016/j.iotcps.2023.05.006.

[35] M. A. Khattak *et al.*, (2018). Progress in Energy and Environment Global Energy Security and Malaysian Perspective: A Review, *Prog. Energy Environ.*, 6;1–18.

[36] C. Ananya, K. Mohit, and C. Nisha, (2023). Secure framework for IoT applications using Deep Learning in fog Computing, *J. Inf. Secur. Appl.*, 77;103569.

[37] R. Kumar, S. Singh, D. Singh, M. Kumar, and S. S. Gill, (2023). A robust and secure user authentication scheme based on multifactor and multi-gateway in IoT enabled sensor networks, *Secur. Priv.*, p. E335.

[38] Rama Lakshmi BOYAPATI, & Radhika YALAVARTHI. (2024). RESNET-53 for Extraction of Alzheimer's Features Using Enhanced Learning Models. *International Journal of Computational and Experimental Science and Engineering*, 10(4);879-889. https://doi.org/10.22399/ijcesen.519

[39] Guven, M. (2024). A Comprehensive Review of Large Language Models in Cyber Security. *International Journal of Computational and Experimental Science and Engineering*, 10(3);507-516. https://doi.org/10.22399/ijcesen.469

[40] Agnihotri, A., & Kohli, N. (2024). A novel lightweight deep learning model based on SqueezeNet architecture for viral lung disease classification in X-ray and CT images. *International Journal of Computational and Experimental Science and Engineering*, 10(4);592-613. https://doi.org/10.22399/ijcesen.425

[41]S.P. Lalitha, & A. Murugan. (2024). Performance Analysis of Priority Generation System for Multimedia Video using ANFIS Classifier. *International Journal of Computational and Experimental Science and Engineering*, 10(4);1320-1328. https://doi.org/10.22399/ijcesen.707

[42]P., A. M., & R. GUNASUNDARI. (2024). An Interpretable PyCaret Approach for Alzheimer's Disease Prediction. *International Journal of Computational and Experimental Science and Engineering*, 10(4);1223-1232. https://doi.org/10.22399/ijcesen.655

[43]Venkatraman Umbalacheri Ramasamy. (2024). Overview of Anomaly Detection Techniques across Different Domains: A Systematic Review. *International Journal of Computational and Experimental Science and Engineering*, 10(4);898-910. https://doi.org/10.22399/ijcesen.522

[44]Türkmen, G., Sezen, A., & Şengül, G. (2024). Comparative Analysis of Programming Languages Utilized in Artificial Intelligence Applications: Features, Performance, and Suitability. *International Journal of Computational and Experimental Science and Engineering*, 10(3);461-469. https://doi.org/10.22399/ijcesen.342

[45]ÇOŞGUN, A. (2024). Estimation Of Turkey's Carbon Dioxide Emission with Machine Learning. *International Journal of Computational and Experimental Science and Engineering*, 10(1);95-101. https://doi.org/10.22399/ijcesen.302

[46]Naresh Babu KOSURI, & Suneetha MANNE. (2024). Revolutionizing Facial Recognition: A Dolphin Glowworm Hybrid Approach for Masked and Unmasked Scenarios. *International Journal of Computational and Experimental Science and Engineering*, 10(4);1015-1031. https://doi.org/10.22399/ijcesen.560

[47]Nuthakki, praveena, & Pavankumar T. (2024). Comparative Assessment of Machine Learning Algorithms for Effective Diabetes Prediction and Care. *International Journal of Computational and Experimental Science and Engineering*, 10(4);1337-1343. https://doi.org/10.22399/ijcesen.606

[48]R. Deepa, V. Jayalakshmi, K. Karpagalakshmi, S. Manikanda Prabhu, & P.Thilakavathy. (2024). Survey on Resume Parsing Models for JOBCONNECT+: Enhancing Recruitment Efficiency using Natural language processing and Machine Learning. *International Journal of Computational and Experimental Science and Engineering*, 10(4);1394-1403. https://doi.org/10.22399/ijcesen.660

[49]ÖZNACAR, T., & ERGENE, N. (2024). A Machine Learning Approach to Early Detection and Malignancy Prediction in Breast Cancer. *International Journal of Computational and Experimental Science and Engineering*, 10(4);911-917. https://doi.org/10.22399/ijcesen.516

[50]guven, mesut. (2024). Dynamic Malware Analysis Using a Sandbox Environment, Network Traffic Logs, and Artificial Intelligence. *International Journal of Computational and Experimental Science and Engineering*, 10(3);480-490. https://doi.org/10.22399/ijcesen.460

[51]Polatoglu, A. (2024). Observation of the Long-Term Relationship Between Cosmic Rays and Solar Activity Parameters and Analysis of Cosmic Ray Data with Machine Learning. *International Journal of Computational and Experimental Science and Engineering*, 10(2);189-199. https://doi.org/10.22399/ijcesen.324

[52]Ponugoti Kalpana, L. Smitha, Dasari Madhavi, Shaik Abdul Nabi, G. Kalpana, & Kodati , S. (2024). A Smart Irrigation System Using the IoT and Advanced Machine Learning Model: A Systematic Literature Review. *International Journal of Computational and Experimental Science and Engineering*, 10(4);1158-1168. https://doi.org/10.22399/ijcesen.526