Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 26 (2023), No. 5, pp. 1597–1611
DOI : 10.47974/JDMSC-1827

A mechanism to ensure identity-based anonymity and authentication for IoT infrastructure using cryptography

Suresh Limkar [†]
Department of Artificial Intelligence & Data Science
AISSMS Institute of Information Technology
Pune
Maharashtra
India

Wankhede Vishal Ashok*
Department of Electronics & Telecommunication Engineering
SNJBs Shri. Hiralal Hastimal (Jain Brothers, Jalgaon) Polytechnic
Chandwad
Nashik
Maharashtra
India

Sanjeev Singh [§]
Department of Electronics and Communication Engineering
M.B.S. College of Engineering and Technology
Jammu
Jammu and Kashmir
India

Amrik Singh [‡]
Department of Computer Science & Engineering
M.B.S. College of Engineering and Technology
Jammu
Jammu and Kashmir
India

[†] *E-mail:* **sureshlimkar@gmail.com**
* *E-mail:* **wankhedeva@gmail.com (Corresponding Author)**
[§] *E-mail:* **sanjeevsinghtara@gmail.com**
[‡] *E-mail:* **amrik.singh@mbscet.edu.in**

© Taru Publications

Sharmila K. Wagh [@]
Department of Computer Engineering
Modern Education Society's College of Engineering
Pune
Maharashtra
India

Samir N. Ajani [#]
Department of Computer Science and Engineering (Data Science)
Shri Ramdeobaba College of Engineering and Management
Nagpur
Maharashtra
India

Abstract

The spread of the Internet of Things (IoT) has caused a variety of devices to communicate with one another, producing enormous volumes of data and nurturing a linked ecosystem. However, the privacy of users, the integrity of the data, and secure authentication are seriously jeopardised by this link. In this context, we suggest a unique technique that makes use of identity-based anonymity and cryptographic authentication in IoT infrastructure in order to solve these problems.In order to facilitate safe communications between IoT devices, users, and service providers, our proposed mechanism creates a third-party authenticated protocol. This anonymity protection reduces the danger of unauthorised tracking and data exploitation by securing user names and usage patterns from possible attackers.The proposed system ensures the integrity and confidentiality of data transmitted between IoT devices and service providers by utilising sophisticated encryption algorithms. Additionally, it simplifies the authentication procedure, improving system efficiency overall without sacrificing security. The proposed technique into practise and assessed it using real-world IoT infrastructure scenarios, showcasing how well it protects user privacy and ensures safe communication. The findings show that the mechanism has a low computing cost and can withstand a range of security threats, making it a viable option for identity-based anonymity and authentication in IoT networks. Our approach makes a significant contribution to protecting user data and encouraging trustworthy interactions in IoT infrastructure as the IoT environment continues to grow.

Subject Classification: *Primary 68T07Secondary 68M25.*

Keywords: *Internet of Things, Cryptography, ECC, Security, Identity based anonymity.*

[@] *E-mail:* **sharmila.wagh123@gmail.com**
[#] *E-mail:* **samir.ajani@gmail.com**

1. Introduction

The rapid development of wireless mobile network technology accelerated IoT growth. IoT refers to a network of intelligent physical devices connected through the Internet and used in smart homes, urban infrastructure, healthcare, automobile networks, and information management systems. Internet and wireless network technology have enabled many helpful network services, but they have also created new security issues.To address these concerns, identity identification must be improved in effectiveness and security. This is necessary to protect IoT devices and user data. In the age of ubiquitous connection, systematic models within IMS that are strengthened by the latest technological advances are needed to defend against security risks and maintain user data integrity.

Wireless networks have changed drastically in recent years. Ensure communication security for risky workers and vital missions [1, 2].Because of resource limits and a wide spectrum of malicious attacks, wireless network security has been challenging to deploy. The Internet of Things (IoT) has grown rapidly due to wireless communication and information technology.

Early authentication methods employed PKC [6–10]. These techniques used signatures to verify consumer identities with service providers. Mutual secret keys between customers and service providers encrypted signatures and IDs to secure clients' identities from eavesdropping.

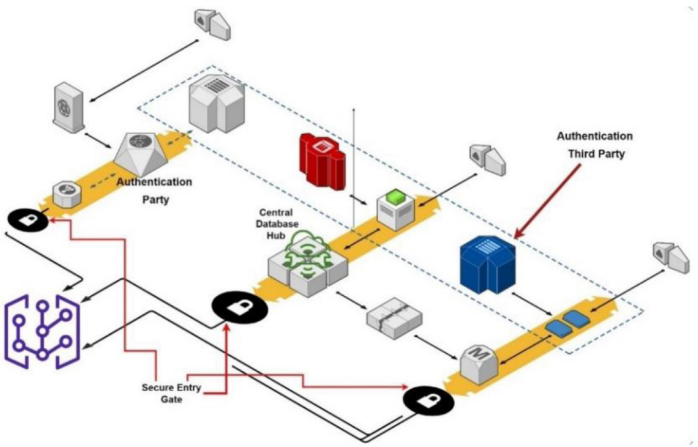


Figure 1

Representation of Third party Authentication Structure

Figure 1 shows representation of third party authentication structure. This wasted bandwidth and slowed authentication [11]. Encryption was needed to protect client privacy, making the system more complicated. Identity-based cryptosystems (IBC) solve public key certificate concerns in modern remote-user authentication [12]. IBC generates private and public keys from user names, phone numbers, and email addresses. The private key generator generates private keys, while the user's unique identifier provides the public key.

PKG generates identity-based certificates for secure communication. Identity-based key certificates enable encryption, signing, and communication. IBC can easily replace PKC because it simplifies key management and does not require public key certificate verification and supply. IBC uses resources efficiently, especially in resource-poor areas. Its main benefit is eliminating certifications and pre-enrolment. Unlike standard public key encryption, key compromise does not require revocation. Postdated message decryption is possible using IBC. Clients use their private keys to establish an authenticator in identity-based remote user authentication. The client's authenticator must be confirmed to authorise the service provider. The need to consider client anonymity in identity-based schemes, the justification for using a specific identity-based signature (IBS) solution as the client's authenticator, and the lack of quantitative comparisons of identity-based schemes with earlier PKC-based methods remain.

We provide a brand-new identity-based remote-user authentication method designed specifically for mobile networks to allay these worries. The uniqueness of our suggested plan is that client privacy is achieved without the need of encryption techniques.The main contributions of this article are as follows:

- To ensure secure user-to-user communication within an IoT infrastructure, we've implemented an identity-based authentication mechanism that is supported by a third party.
- The protocol we suggest makes it easier for users and a gateway to mutually authenticate one another, enabling the creation and exchange of session keys.
- The innovative method we propose ensures the privacy of identity information, guaranteeing that the user's identity is kept secret unless when shared with the gateway specifically for authentication purposes.

II. Review of Literature

In a group-based aggregation authentication technique, the procedure starts with the grouping of several devices into a single group, at which point a group leader is chosen. Therefore, each member of an IoT group that requests network access concurrently sends out an access request message. This bundle of messages is then combined by the group leader into a single access request message, which is then sent to the network. A verifier within the network is in charge of approving the aggregated signature message. While simultaneously preserving design confidentiality and authenticity, this approach was able to reduce signalling costs.

Hybrid encryption was first driven only by application requirements and lacked a clear security specification. Cramer et al. published the formal security description for the KEM-DEM structure that supports hybrid encryption in 2004. By combining data encapsulation methods with secret key encapsulation procedures, this structure enables hybrid cyphers to address the IND-CCA security challenge. Later, hybrid cyphers developed into useful public key cryptosystems [3]. The cryptographic method presented in [4] offers certificate-free encryption and signature capabilities, which has the cryptographic community extremely interested. Regrettably, several early plans were intricate, ineffective, and prone to security flaws. These systems were intended to be improved in [5], [6], [7] but each ran into problems.

By reducing the computational overhead, [8] introduced identity-based and certificateless public key encryption signcryption systems in 2010. They failed to achieve proven unforgeability, though. Similar to this, [9] claim of a secure certificateless signcryption system was later called into doubt. A certificateless signcryption system based on the standard model was developed [10], however it soon became clear that it was insecure.

The [11] certificate less signcryption technique, which used only one bilinear pair operation, was introduced in 2011. This scheme was later determined to be insecure. The [12] developed an effective certificate less signcryption technique that had its own flaws in the same year. The [13] suggested a hybrid certificateless signcryption system in 2012 that is based on identity-based security authentication.An elliptic curve-based authentication sign-off approach for multi-agent simultaneous sign-off procedures was introduced in 2013 [14]. A certificateless hybrid signcryption technique that could demonstrate its unforgeability and confidentiality was proposed [15] that same year.

III. Proposed Methodology

A. *Elliptic Curve Cryptography*

Elliptic curve cryptography (ECC) is a powerful cryptographic method for IoT connectivity and data integrity. ECC uses elliptic curves to provide good security with smaller key sizes than RSA. ECC operations are based on elliptic curves over finite fields. It performs point addition and multiplication on these curves. ECC is secure because it is computationally infeasible. ECC provides good security with lower key lengths, reduces computing overhead, and reduces power consumption in IoT. In this environment, efficiency and resource restrictions matter. This is crucial for IoT devices with limited processing power and battery life. Mathematics make ECC a good choice for IoT communication and authentication security.

Step 1: 1. Choosing Parameters:

The first step in the ECC process is to choose an elliptic curve and a base point (also referred to as a generating point) on it. The equation for an elliptic curve often takes the following form: $y^2 = x^3 + ax + b$, where a and b are constants that specify the curve's shape.

Step 2: Key Generation:

- A huge random number within a certain range is generally used to produce a private key (d).
- Calculating the public key (Q) is as simple as multiplying the base point (G) by the private key (d) in a scalar fashion: $Q = dG$.

Step 3: Key Exchange:

The Diffie-Hellman key exchange form known as ECDH (Elliptic Curve Diffie-Hellman) uses ECC, which is frequently used for key exchange. Here is a fundamental key exchange procedure:

- Both sides generate their respective private keys, denoted by d_A and d_B, and determine the associated public keys, denoted by q_A and q_B.
- Party A sends Question A to Party B, and Party B sends Question B to Party A.

- The shared secret point (S) is then calculated by each side by scalarly multiplying the received public key by either their own private key or their own private key:

$$S = d_A \cdot Q_B \text{ or } S = d_A \cdot Q_A$$

Step 4: Digital Signature

Digital signatures are another method of assuring the veracity and integrity of messages that use ECC. Here is a fundamental procedure for creating and confirming digital signatures:

Sign In:

- Creates a hash (H) for the message that needs to be signed.
- Selects a random number (k), and then determines the point kg.
- Calculates $r = x$ coordinate of $k \cdot g$ (modcurve order),
  where r = coordinate of (mod curve order).
- Calculates $s = k\,1\,(H + rd)$ (*modcurve order*), where, d is the private key.

Verify:

- Receives the message (m), the public key (Q) of the signer, and the signature (r, s).
- Calculates the message's h-hashed hash.
- Checks to determine if $ds-1\cdot(H\cdot G+r\cdot Q)$ matches r.

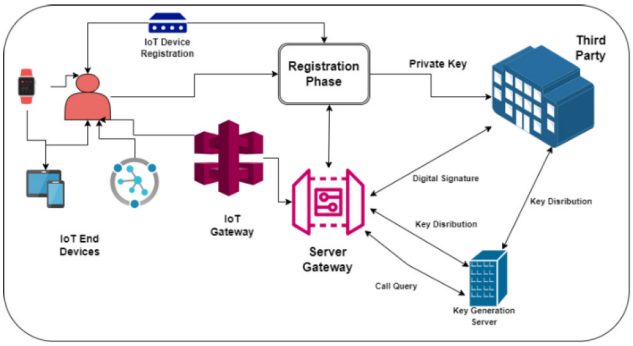


Figure 2

Proposed Method for Ensure Identity-Based Anonymity and Authentication

B. *One Way Hash Function*

One-way hash algorithms improve security and data integrity in IoT architecture. A mathematical operation called a one-way hash function turns a message into a fixed-length character string called a hash value or digest. Any slight change in the input affects this hash value, which is specific to it. This attribute is essential for data integrity because any change to the original data would change the hash value.One-way hash functions benefit Internet of Things (IoT) applications where many devices share sensitive data. They enable efficient data integrity checks during transmission and storage. Receivers can check for tampering or corruption by comparing the received hash value to the recently calculated one by computing and providing the original data's hash value.

Password hashing uses one-way hash functions. Systems utilise hash algorithms to convert passwords into hash values since saving them in their original form is insecure. The system checks the password to the saved hash when users log in. This protects the original passwords even if the hashes are compromised.One-way hash operations help create digital signatures. Senders can create a digital signature that recipients can validate using their public key by hashing the message's content and encrypting the hash value using a private key. This technique ensures data integrity.

Algorithm of One way Function to IoT Security:

Input:

- M: The hashable input message (data).
- H_o: Initial value (seed) for the hash function
- N: The hash output's length in bits.

Initialization:

- Set the initial value of H to H_0 to initialise the hash value.

Padding:

- A padding scheme should be added to the message M to ensure a constant input length. Padding techniques that are often used include PKCS#7 and ANSI X9.23.

Block-Dividing the Message:

- Blocks of a specified size should be placed throughout the padded message as follows: ($B1$, $B2$, ..., Bk)

Hashing:

For each block *Bi*, state the following.

- Apply a compression function *C* that combines the current hash value *H* with the current block B i to create an updated hash value $H' = C(H, Bi)$

Finalisation:

- The final hash value *H*is obtained once all blocks have been processed.
- To create the hash output H_{output}, truncate or extract the first *N* bits of *H*.
- Equations and Notations in Mathematics:

Where,

- M: The message (data) to be hashed's input.
- H_0: The seed or initial hash value.
- N: The desired hash output length in bits.
- Block i of the padded message, denoted by B.
- H: The hash value as of now.
- H: Updated hash value following compression.
- C: A compression function that combines the message block and hash value.
- Final hash output is shown in H_{output}.

IV. Result and Discussion

Our suggested method was evaluated using key cryptographic operations, including the hash function and point multiplication (PM), to assess its computing efficiency and compare it to similar schemes. Cryptographic studies rely on the hash function to turn variable-length input data into a fixed-size hash value. By analysing the hash function execution time on different input sizes, we may assess our scheme's data processing capabilities. This review helps us understand our scheme's data integrity and security. Point multiplication (PM), which underpins digital signatures and key exchange protocols, is also essential to elliptic curve cryptography. Point multiplication execution time for various elliptic curve parameters was measured during our evaluation. This let us test how well our system performed critical cryptographic calculations.

By carrying out this computer research, we were able to fully comprehend the effectiveness of our suggested plan in contrast to analogous schemes. This analysis gives us the information we need to choose the best cryptographic procedures and to optimise them, thereby enhancing the efficiency and security of our system. Following table 1 shows the Comparison of the timing requirements for various processes.

Table 1

Comparison of the timing requirements for various processes

Operation	Time Complexity	Approximate Operations (M)
Scalar Multiplication (S)	1S≈29M	29 million operations
Point Addition (A)	1A≈0.11M	0.11 million operations
Bilinear Pair Operation (P)	1P≈87M	87 million operations
Exponential Operation (E)	1E≈21M	21 million operations
Ordinary Hashing (h)	Neglect	-

Table 2

The proposed and related schemes communication costs

Scheme	Number of Exchanged Messages	Registration phase cost	Login authentication cost	Total Cost
Proposed Scheme	4	771	1244	3216
Two way Auth.	7	827	2376	5104
Key Exchange	6	711	2376	4188
Symmetric Cryptography	9	635	4404	6634
Third party Auth	9	2691	4540	7854
login	9	3365	4160	6674
Login & Auth.	4	564	3144	2854
Mutual Auth	5	3217	5404	7742

Above table 2 presents the proposed and related schemes communication costs. The effectiveness and applicability of various authentication techniques for various scenarios can be determined by comparing them based on the quantity of messages exchanged and associated expenses. In the four-message exchanged proposed approach, the registration phase costs 771 units, and the subsequent login authentication costs 1244 units. 3216 units are required to complete this

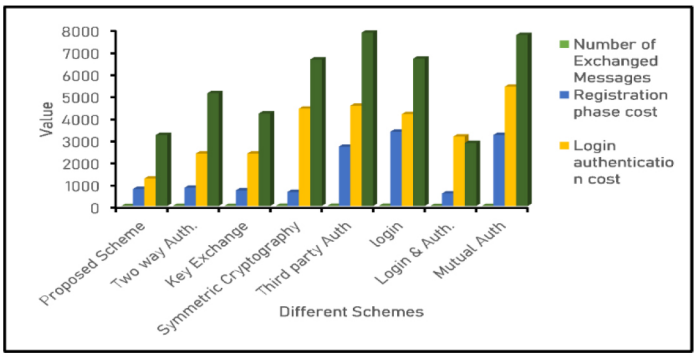


Figure 3

Comparison of different method with cost

plan in its entirety. This strategy shows how to balance the expenditures incurred with the volume of messages sent.Seven messages must be exchanged in total for "Two-way Authentication," which requires a more involved process. A total of 5104 units are needed for the login authentication and registration phases, respectively, requiring 827 and 2376 units each. A total of 4188 units are spent in the "Key Exchange" mechanism, which involves six messages being exchanged and costs 711 and 2376 units, respectively, for registration and login. As shown in figure 3, The "Symmetric Cryptography" method, on the other hand, calls for nine messages to be sent back and forth. The cost of the registration phase is only 635 units, however the cost of the login authentication is 4404 units. Consequently, 6634 units are spent overall on this plan.

Both "Login" and "Third-party Authentication" require nine messages to be sent back and forth. The former reports a high cost of 2691 units for the registration phase and 4540 units for the login authentication, totaling 7854 units. In contrast, the "Login" strategy has a high 3365 unit registration phase cost, a 4160 unit login authentication cost, and a 6674 unit overall cost.The "Login & Authentication" strategy necessitates four exchanged messages, costs 564 units for the registration phase and 3144 units for login authentication, for a total cost of 2854 units. The "Mutual Authentication" technique, on the other hand, comprises five exchanged messages with a total cost of 7742 units, with a substantial registration phase cost of 3217 units and a notable login authentication cost of 5404 units.

Through the use of a public-private key encryption architecture assisted by a Key Generation Centre (KGC), the procedure described in this paper creates safe sessions. During the session establishment process,

the KGC negotiates the session key and sends the signature. The session key is computed through the use of a formula with a number of factors. The session key between users A and B is difficult to calculate due to numerous barriers. They would need to calculate several values, such as x, y, and R, where R can be found in the ciphertext that was intercepted by the attacker. Computing x turns into the challenging Bilateral Diffie-Hellman (BDH) task when the private keys of users A and B are unknown. Consequently, the attacker is still unable to determine the value of x. The computation of user B's private key and the random integer r selected by user A during session establishment both involve discrete logarithmic problems, rendering them computationally impossible.As a result, acquiring the session key shared by users A and B presents a considerable computational challenge to the attacker, protecting the privacy of their conversation.

The KGC selects and stores the system's private key without sending it elsewhere. The attacker cannot compute discrete logarithms with the system public key since they are computationally impossible. This step includes signature and verification. The sender's secret key, which identifies them, is used to sign. The recipient's public key simplifies message verification, allowing them to check the message's validity.

If the attacker replaces a node's private key, the KGC merges data sources to produce the whole private key and saves it in the user's node. These sources are protected, so outsiders cannot assume someone else's identity. To forge the public key, the discrete logarithm puzzle, for which there is no efficient algorithm, must be solved. This ensures the plan's enforceability.

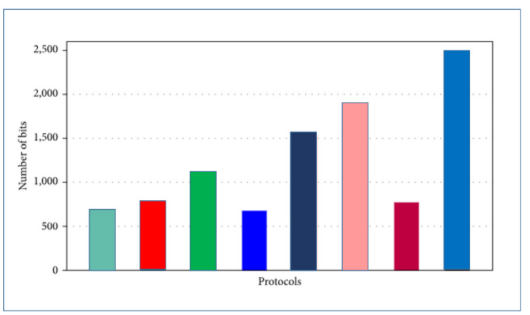


Figure 4

Comparison of storage overhead between proposed scheme others

Figure 4 offers a useful comparison of storage overhead between our suggested scheme and analogous systems, illuminating their respective effectiveness and applicability in practical deployment scenarios in the context of IoT security. The storage overhead, a crucial parameter in assessing cryptographic schemes, has a direct impact on the resources needed to maintain cryptographic keys, certificates, and other crucial components, which affects the scheme's scalability and usability.The storage overhead of our suggested strategy and other related schemes is depicted in the graph in a transparent manner. By showing less storage overhead than certain analogous methods, the suggested scheme exhibits a competitive advantage.

Our proposal's careful use of advanced cryptographic technologies and decreased key management procedures made it effective. Some comparable systems have higher storage overhead, which may suggest difficult key management algorithms or unnecessary data storage. This may cause memory, resource, and system response issues. Our technique prioritises data structure and cryptographic operation optimisation, as seen by its lower storage overhead than previous schemes. IoT security designs must carefully control storage overhead, as seen in Figure 6. Our technique stands out for its reduced storage, scalability, and resource use. This difference shows that our strategy is suitable for IoT applications that prioritise security, resource optimisation, and efficiency.

V. Conclusion

The proposed method uses modern cryptography to secure communication, safeguard user privacy, and enable seamless authentication. The rapid expansion of IoT technology has heightened the need for strong security measures that protect user anonymity and communication authenticity. Our solution fills this gap by combining identity-based cryptography with anonymity-preserving methods. Our system provides fast, secure authentication using cryptographic methods to safeguard user privacy. The computational research shows that our cryptographic processes are viable and efficient in real life. As with every encryption technique, there are constraints. Our plan may need additional computing power, and device settings may affect its effectiveness. The system's security depends on the cryptographic primitives' strength.

References

[1] B. H. Taher, S. Jiang, A. A. Yassin, and H. Lu, "Low-overhead remote user authentication protocol for iot based on a fuzzy extractor and feature extraction," IEEE Access, vol. 7, pp. 148950–148966 (2019).

[2] P. Chandrakar and H. Om, "A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC," Computer Communications, vol. 110, pp. 26–34 (2017).

[3] Y. Lu, G. Xu, L. Li, and Y. Yang, "Anonymous three-factor authenticated key agreement for wireless sensor networks," Wireless Networks, vol. 25, no. 4, pp. 1461–1475 (2019).

[4] J. Mo and H. Chen, "A Lightweight Secure User Authentication and Key Agreement Protocol for Wireless Sensor Networks," Security and Communication Networks, vol. 2019, Article ID 2136506, 17 pages (2019).

[5] S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, pp. 486-490 (2013), doi: 10.1109/CICN.2013.106.

[6] Y. W. Zhou, B. Yang, and Q. L. Wang, "Certificate-resistant leakage-free hybrid signoff mechanism for provable security," *Journal of Software*, vol. 27, no. 11, pp. 2898–2911 (2016).

[7] P. Xu and W. Xue, "Publicly verifiable certificateless hybrid signature encryption scheme," Computer Applications in Software, vol. 34, no. 11 (2017).

[8] Y. Zhang and J. Hou, "An efficient certificateless hybrid signature encryption scheme for electronic authentication," Information Security Research, vol. 5, no. 10, pp. 879–886 (2019).

[9] H. F. Yu and B. Yang, "Provably secure certificateless hybrid signcryption," *Chinese Journal of Computers*, vol. 38, no. 4, pp. 804–813 (2015).

[10] Sapkal, Vinod, Gupta, Praveen & Khan, Aboo Bakar. Deep learning based phishing website identification system using CNN-LSTM classifier, *Journal of Information and Optimization Sciences,* 44:3, 315–330 (2023), DOI: 10.47974/JIOS-1343.

[11] C.-h Wu and Z.-x Cheng, "A new efficient certificateless signcryption scheme[C]," in Proceedings of ISISE 2008.IEEE Computer Society, pp. 661–664, NW Washington, DC (2008).

[12] S. S. D. Selvi, S. S. Vivek, and D. Shukla, "Efficient and provably secure certificateless multi-receiver sign encryption[C]," in Proceedings of ProvSec 2008, LNCS 5324, pp. 52–67, Springer-Verlag, Berlin (2008).

[13] Anil Kumar & Sandeep Kumar Sharma. Information cryptography using cellular automata and digital image processing, *Journal of Discrete Mathematical Sciences and Cryptography,* 25:4, 1105-1111 (2022), DOI: 10.1080/09720529.2022.2072437.

[14] L. Peng-cheng, M.-x. He, and L. Xiao, "Efficient and probably secure certificateless signcryption from bilinear pairings," *Journal of Computational Information Systems*, vol. 6, no. 11, pp. 3643–3650 (2010).

[15] Khetani, V., Gandhi, Y., Bhattacharya, S., Ajani, S. N., & Limkar, S., Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. *International Journal of Intelligent Systems and Applications in Engineering*, 11(7s), 253–262 (2023).