A Comprehensive Analysis of Key Agreement Methods in IoT: Unveiling Multi-Classifications and Exploring Diverse Aspects

Rasoul Roustaei[1], Hamid Haj Seyyed Javadi[2] and Midia Reshadi[3]
[1,3]*Department of Computer Engineering, Science and Research Branch, Islamic Azad University*
[2]*Department of Computer Engineering, Shahed University*
RassoulRoustaei@gmail.com, h.s.javadi@shahed.ac.ir, reshadi@srbiau.ac.ir
Corresponding author: h.s.javadi@shahed.ac.ir

***Abstract*- In the ever-evolving landscape of technology, the pervasive impact of the Internet of Things (IoT) continues to reshape established paradigms. The need for secure key agreement protocols has become a significant focus. Unfortunately, prevailing classification schemes often fall short in desired comprehensiveness and neglect specific application contexts. A key challenge in this domain arises from insufficient consideration of the unique physical limitations and infrastructure in IoT environments, resulting in a proliferation of diverse yet frequently unsuitable key exchange methods. This paper addresses these challenges by presenting a novel methodology for classifying key agreement methods tailored to the demands of the IoT landscape. Our approach introduces four carefully crafted classifications, enhancing understanding based on key criteria: protocol process, features and capabilities, resource requirements, and communication models. Through these comprehensive classifications, we aim to provide a nuanced perspective on IoT key agreement protocols, offering insights into their suitability for diverse applications and alignment with IoT constraints. Our research, employing this multidimensional framework, contributes to a profound exploration of IoT security and key agreement methods, providing invaluable insights for researchers, practitioners, and decision-makers in the dynamic IoT landscape.**

***Index Terms*-** Key Agreement, Security, Authentication, Secure Communication, Key Exchange.

I. INTRODUCTION

The rise of the Internet of Things (IoT) marks a transformative era, where sensor networks and embedded devices seamlessly connect with the broader Internet. IoT has become an integral part of daily life, with applications spanning smart homes, healthcare, and smart cities. Securing these systems is crucial, as sensitive user information travels across networks, making IoT security a top

Manuscript received 15-Oct.-2022 and revised 21-Nov.-2022, *P- ISSN: 2322-4088*
Accepted on 28-Dec.-2022 *E- ISSN: 2322-3936*

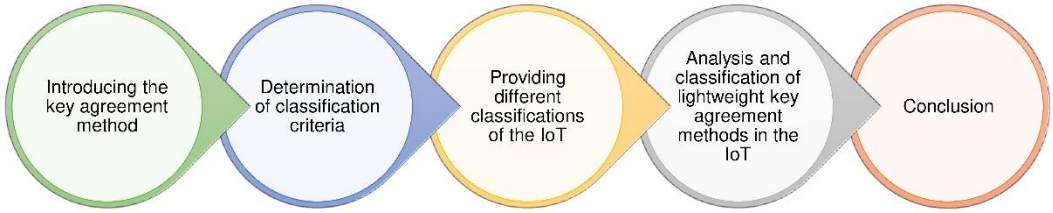

Fig. 1. The organization of the paper

priority. Addressing these security challenges, while ensuring confidentiality and privacy, has brought secure communication methods in IoT to the forefront of research.

Encryption is essential for protecting data during transmission. Symmetric encryption stands out in IoT applications due to its speed and superior performance compared to asymmetric methods. However, the key challenge in symmetric encryption lies in establishing a shared key securely for communication. Key agreement methods are critical mechanisms used to achieve this goal.

Understanding and classifying protocols and algorithms help in comparing their functionalities and making informed choices. While numerous studies have explored the classification of key agreement and key exchange methods, many have focused solely on protocol details and overlooked other important factors. Moreover, they often do not consider the specific requirements and limitations unique to IoT environments. For instance, expecting IoT devices to meet stringent biometric-based authentication standards is often impractical.

This paper presents new classifications that encompass the diverse aspects of key agreement methods, tailored to the unique environmental requirements and constraints of IoT. Additionally, it distinguishes between key agreement and key exchange methods, offering clearer guidance for their application.

The organization of this article unfolds as follows (Fig. 1): first, we delve into the key agreement method, followed by the delineation of classification criteria for key agreement methods in Section III. Subsequently, we present several classifications for these methods. Section IV focuses on the classification of lightweight key agreement methods within the aforementioned categories.

II. Key Agreement Methods

Ensuring communication security is a paramount objective in key management, and a critical aspect of this is the establishment of keys on both ends of the communication. Before transmitting any message securely, a decryption key must be available on the receiver's side, a requirement fulfilled through either the symmetric or asymmetric key agreement methods. Asymmetric key agreement methods uses a pair of public and private keys. Asymmetric key agreement methods use a pair of public and private keys. Each entity has its own private key, which is kept secret, and a public key that is shared with others. When one entity wants to send an encrypted message, it uses the recipient's public key to encrypt it, and the recipient uses its private key to decrypt the received message. The

exchange of public keys between communicating parties is the only requisite. Given that public keys are, by definition, publicly accessible, there are no inherent challenges related to the confidentiality of this exchange. Symmetric key agreement methods use the same shared key for both encryption and decryption. Both communicating entities must have access to this secret key, which they use to encrypt outgoing messages and decrypt incoming messages, making secure key distribution essential to maintain confidentiality. Establishing these keys securely on both ends necessitates a thoughtful approach that prioritizes security considerations. Unlike asymmetric methods, symmetric methods require both parties to securely share or distribute the same key, creating a unique set of challenges**.** Key exchange protocols play a crucial role in this process, serving as mechanisms that facilitate the collaborative creation of a shared encryption key for data encryption. In many key exchange systems, the sender typically generates the key and transmits it to the receiving party, who has no influence over the key generation process. However, the landscape shifts in key agreement protocols, a specific type of key exchange method. In these protocols, both communicating parties actively contribute to the generation of the final key. This nuanced distinction adds a layer of intricacy to the process, underscoring the significance of careful consideration and coordination between entities involved in the communication.

III. MULTI-CLASSIFICATIONS OF KEY AGREEMENT METHODS

Key exchange methods predominantly leverage asymmetric techniques to facilitate message transmission during the key agreement process. These methods can be broadly classified into two primary groups: Key Agreement and Key Transfer.

In key transfer protocols, one party is tasked with generating the key, which is subsequently securely transmitted to another party. In contrast, key agreement protocols involve the utilization of information from both parties to collaboratively create the key, with the option to transmit this information either directly or indirectly. It’s noteworthy that these distinctions can sometimes become blurred. Some references, such a [1] s, refer to both categories as ’key agreement,’ while others, including [2, 3], emphasize the clear differentiation between these two categories. Generally, these distinctions are somewhat reliant on definitions, and these definitions may vary across different contexts.

Various classifications and categories have been proposed to categorize secret methods of key establishment. For instance [4], provides a classification specific to key establishment methods in the context of the Internet of Things (IoT). According to this classification, protocols are divided into two primary categories: symmetric and asymmetric, based on the mechanism employed for key establishment.

In addition to this classification, there are multiple features and criteria that can be utilized to

Manuscript received 15-Oct.-2022 and revised 28-Nov.-2022,
Accepted on 29-Dec.-2022
P- ISSN: 2322-4088
E- ISSN: 2322-3936

Fig. 2. Multi-Classifications of Key Agreement Methods

further classify and categorize key establishment methods. Distinguishing criteria and characteristics of key agreement methods can be examined based on various factors, including the procedural approach, the communication model, the required resources, and the performance outcomes of the key agreement protocol (Fig. 2). As a starting point, our primary objective is to establish a comprehensive classification based on the procedure itself. This comprehensive approach enhances our understanding of the various methods, enabling a more detailed analysis and assessment of key establishment protocols.

A. Classification by Methodology

In the classification based on procedure, key agreement algorithms are systematically organized by how they execute key agreement operations and their underlying processes. Fig. 3 presents a comprehensive classification chart, enhancing our understanding and facilitating precise referencing within the taxonomy tree. This classification delves into key establishment on both sides, achieved through either key exchange or key distribution (the second level of the tree), with distribution orchestrated by a third entity before communication initiation.

Key Exchange vs. Key Transfer: Within the key exchange category, key establishment unfolds through the transfer of information between parties. When one party generates the key based on its knowledge and transmits it to another, this method is termed key transfer. In contrast, in key agreement, the secret key is created based on the information and secret components of both parties (the third level of the tree).

Participation and Interaction: Further classification of key agreement methods can be based on the level of participation and interaction among the involved parties in the key creation process. In one-party key agreement, the initiating party crafts the key using the secret components and information of all parties involved and then transfers it. Notably, the necessary information and components for key

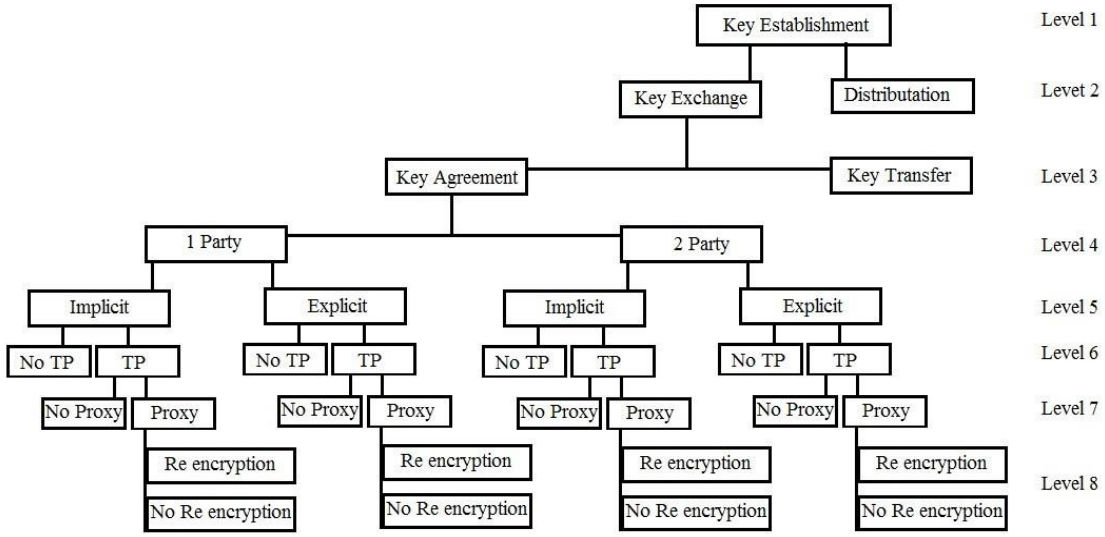

Fig. 3. Classification Chart of Key Agreement Algorithms Based on Procedure

creation can be supplied to the initiating party by the Key Distribution Center (KDC) or a trusted third party. Conversely, two-party key agreement methods involve both parties in the key creation process (the fourth level of the tree) [5].

Key Creation Methods: The key creation and agreement process can be executed explicitly or implicitly. In explicit methods, separate messages are transmitted to create the key, while implicit key extraction involves deriving a key from the sent message, often referred to as an automatic key. An explicit key is one that is not part of plain text (the fifth level of the tree) [1].

Key Agreement Initialization: In all key agreement methods, the presence of a Key Distribution Center (KDC) or Registration Center (RC) is crucial for initializing the parties and registering their specifications, typically in the initial setup phase through secure communication. Some methods, particularly when one of the communication parties is a resource-rich server, allow the server to assume the role of KDC/RC during the start-up phase, eliminating the need for a separate KDC or RC [6]. Additionally, there are methods where a third party, distinct from the KDC, contributes to establishing secure communication. The presence or absence of a third party is another classification criterion (the sixth level of the tree).

Classification of Third Parties: The third party can be categorized into several roles based on its level of trustworthiness and involvement. If the third party is highly trusted, it is referred to as a Trusted Third Party (TTP), responsible for tasks such as authenticating parties, facilitating message transmission, aiding in key creation, and re-encryption. In some scenarios, the third party is semi-trusted, serving as a proxy for data transmission between parties [7]. In other instances, the third party only plays a role in the initial key creation steps and remains uninvolved afterward, leading to its classification into two groups: key proxy and key generator [8] (the seventh level of the tree).

Manuscript received 15-Oct.-2022 and revised 28-Nov.-2022,
Accepted on 29-Dec.-2022
P- ISSN: 2322-4088
E- ISSN: 2322-3936

Table I: Criteria for Classification of the Key Agreement Algorithms by procedure

Criteria	Classifications		
Exchange Type	Key Agreement	Key Transfer	
Level of participation	1 Party	2 Party	
Key Transfer	Explicit	Implicit	
Third Party	No TP	TP	
		Semi-Trusted Entity	Trusted Third-party
Role of TP	Gateway	Key Establish	Key Proxy
Re Encryption	Re Encryption	Non Re Encryption	
Relation of KDC	Interactivity(Online)	Non-Interactivity(Offline)	

Occasionally, the third party operates as a gateway (access point) and plays a role only in message transmission during key agreement operations.

Proxy Re-Encryption Scheme (PRE): The Proxy Re-Encryption Scheme allows a proxy to convert encrypted text using the sender's public key into intended encryption for the receiver, enabling the sender to send encrypted messages to the receiver temporarily without revealing the sender's secret key. A key feature of proxy re-encryption is that the proxy is not completely trustworthy, meaning it does not possess the secret keys of the parties [9]. In this scheme, third parties (proxies) can modify encrypted text, enabling decryption by other parties [10]. The sender transmits a message to a proxy, which doesn't view the original message but facilitates the receiver in deriving the message by making changes in it. In proxy encryption, there's no need to re-encrypt the original message entirely; only a portion of the message that includes the decryption key is re-encrypted [11] (the eighth level of the tree). Table I lists the classification criteria, including an interactive criterion indicating interaction with the KDC, which is not shown in Fig. 3 due to a reduction in the classification levels. In a non-interactive scheme, the sender can generate an encryption key offline using its secret key and the receiver's generic values, without the need for the KDC, proxy, or receiver [9]. In contrast, interactive designs require the involvement of entities like KDCs to generate re-encryption keys. These interactive methods are referred to as "online" while non-interactive methods are termed "offline".

B. Classification by Properties and Performance Specifications

After classifying key agreement methods based on their procedural aspects, we will now explore a second classification focused on the properties and performance specifications of these methods. More precisely, we will consider certain attributes as classification criteria that are integral to the results achieved [9, 11]. In this field, several criteria are identified, such as anonymity, bi-

Table II:Classification Criteria Based On Results of the Key Agreement Methods

Criteria	Classifications	
link ability	link ability	Unlink ability
Anonymity	Yes	No
Transitive	Transitivity	Non-Transitivity
Encryption Direction	Bidirectional	Unidirectional
Forward Secrecy	Yes	No
Backward Secrecy	Yes	No
Authentication	Yes	No
	Implicit / Explicit	
Multiple Use	Yes	No

directionality of the secret key, key transitivity, the possibility of connecting source and destination of messages, among others, which are listed in Table II.

Decoding Direction: Key agreement methods can be classified into two modes of decoding direction: one-way and two-way. In a one-way scheme, encrypted messages from user A are exclusively decrypted by user B, with user A unable to decrypt them. In contrast, a bidirectional scheme allows the sender or receiver to use the same decryption key for messages exchanged between the two parties, regardless of their role.

Multiple Usability: When a proxy re-encryption scheme has the capability to re-encrypt an encrypted message for different entities multiple times (multiple consecutive encodings on an encrypted message), it is referred to as multiple-use. In simpler terms, the proxy can repeatedly re-encrypt and send the same message to new recipients. For example, a message initially sent from sender A to receiver B can be re-encrypted by the proxy for receiver C [9]. Conversely, when a proxy can perform only a single re-encryption, it is termed a single-use scheme.

Transitive: A method is classified as transitive if the two encryption keys between A and B and between A and C can be used to derive the encryption key between A and C. In a non-transitive scheme, the proxy cannot grant new decryption rights by combining keys with re-encryption keys.

Anonymity: An encryption method is considered anonymous if an outsider cannot identify the sender and the receiver of messages simply by viewing the message [7].

Link ability: If the collection and tracking of sent messages over an extended period lead to a specific pattern that can identify the origin and destination of messages, the encryption method is said to have link ability.

Forward Secrecy: Often referred to as Perfect Forward Secrecy (PFS), this feature in key agreement protocols ensures that compromising a session key only endangers the data of that particular session. By generating a unique session key for each message transmission session, if a session key is

Manuscript received 15-Oct.-2022 and revised 28-Nov.-2022, *P- ISSN: 2322-4088*
Accepted on 29-Dec.-2022 *E- ISSN: 2322-3936*

compromised, only the data for that specific session is exposed, without affecting the keys and data of other sessions. This assures the confidentiality of past sessions against future key compromises [12, 13] . In some references, this feature is termed past key confidentiality and ensures that keys from previous sessions remain uncompromised [14].

Backward Secrecy: Unlike forward secrecy, backward secrecy focuses on ensuring that the compromise of the current key does not affect future sessions [14]. It safeguards the privacy of future keys, often referred to as future keys privacy.

Authentication: In the realm of computer communications, authentication is the process of verifying the identity of communicating parties, assuring the message recipient that the message originated from the declared source. Any mechanism that effectively confirms or rejects a person's identity constitutes an authentication service [5]. While authentication can be achieved through key agreement, some key agreement protocols do not provide authentication, leaving them vulnerable to Man-in-the-Middle attacks. Consequently, it is imperative to combine authentication with key establishment to enhance security [15].

Key Authentication: Beyond party authentication, key agreement procedures may also incorporate key authentication for the session key [16]. This key authentication can be categorized into two types:

- **Implicit Authenticated Key Agreement**: A key agreement protocol provides implicit key authentication (from party B to party A) if party A is assured that no entities other than party B can extract the value of the secret key. It's important to note that the authentication of the implicit key does not necessarily confirm whether party B has the key or participated in the protocol. A key agreement protocol that provides implicit key authentication for both participating parties is known as an Authenticated Key (AK) agreement protocol [17, 18].
- **Explicit Authenticated Key Agreement**: In the case of explicit authenticated key agreement, entity A is certain that entity B possesses the secret key resulting from the agreement. This protocol provides both implicit authenticated key (assuring key confidentiality) and key authentication (ensuring that the session key is received). A key agreement protocol that offers explicit authenticated keys for both participating entities is referred to as an Authenticated Key Agreement with Key Confirmation (AKC) [17, 19].

By considering these criteria and classifications, we gain a more comprehensive understanding of the nuances within key agreement methods, and these criteria are vital for evaluating and choosing the most suitable method for various communication scenarios.

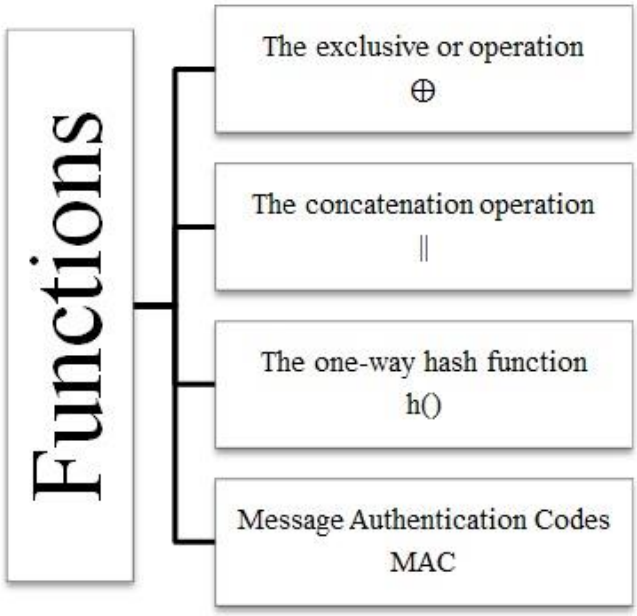

Fig. 4. Functions and Operators used in the lightweight key agreement schemes

C. Classification by Resource Requirements and Protocol Prerequisites

When considering cryptographic algorithms and key management protocols, resource consumption plays a pivotal role. Neglecting the resource requirements can lead to a substantial drain on available resources. This is particularly critical in the context of modern wireless and mobile systems, such as cell phones and embedded computers used in devices within the Internet of Things (IoT), all of which typically have resource constraints. These limitations can encompass factors like limited storage

capacity, processing power constraints, energy scarcity, and restrictions on long-distance wireless communication [5]. In light of these challenges, researchers have dedicated significant efforts to streamline their algorithms and protocols, making them more lightweight and resource-efficient.

One fundamental approach to achieve this is the elimination of resource intensive operations. Operations like exponentiation and discrete logarithms are examples of computationally expensive functions. Key agreement algorithms are generally designed to be lightweight and do not rely on such resource-intensive operations. Instead, they often utilize operations like one-way hash functions, XOR, and similar lightweight computational techniques. Additionally, lightweight key agreement methods frequently employ the involvement of a trusted third party for secure communication and message conveyance [20]. An overview of these actions is presented in Fig. 4.

Towards the end of this section, Fig. 5 outlines the hardware and software requirements essential for the implementation of key agreement methods. These specified requirements are primarily geared

Manuscript received 15-Oct.-2022 and revised 28-Nov.-2022,
Accepted on 29-Dec.-2022
P- ISSN: 2322-4088
E- ISSN: 2322-3936

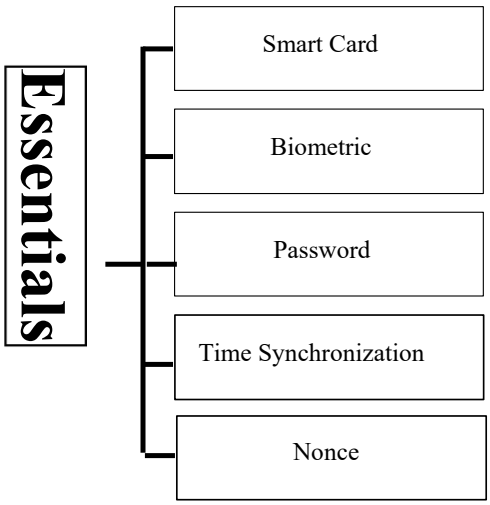

Fig. 5. Practical Requirements and Prerequisites for Key Agreement schemes

toward implementing the authentication component of the algorithms. Given the multifaceted challenges posed by the IoT, including resource constraints and the absence of human users in many scenarios, not all authentication algorithms are suitable for IoT applications. For instance, authentication methods relying on biometric parameters or ID cards are typically tailored for human users and may not align with the resource limitations characteristic of the IoT. Consequently, the careful consideration of protocol requirements becomes paramount in determining their feasibility for implementation within each distinct IoT environment.

D. Classification by Communication Models

A crucial consideration in the proposed methods for key agreement lies in the communication architecture and the types of devices involved in the plan. Initially, the presented methods begin by outlining a communication model and defining the types of devices that are part of the plan. Fig. 6 illustrates various communication models, and this classification plays a pivotal role in the selection and preference of one design over another. The type of communication parties involved significantly impacts the available resources and computational power. Importantly, it also determines the level of trust in the communication.

The first communication category is the device-to-device model, which focuses on communication between two devices within the IoT environment. Generally, this model represents one of the most challenging scenarios as both parties have limited resources and a minimum trust level. In the user-to-device communication model, the objective is to grant users access to IoT devices. In this model, the trust level associated with users is higher due to the various authentication methods available. As discussed in section III.C, some of the specified items in this section, such as biometric authentication, are typically tailored for human use and may not be easily applicable to IoT devices.

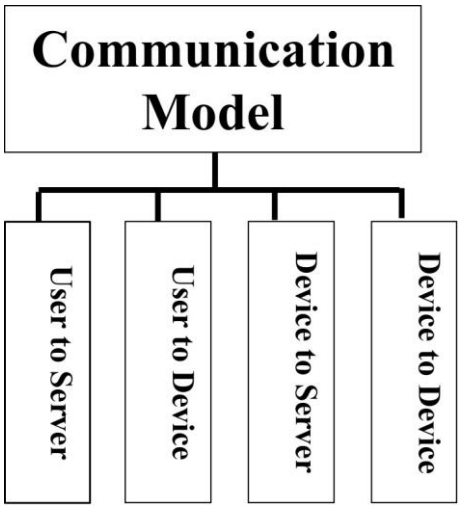

Fig. 6. Types of communication model in the Key agreement methods

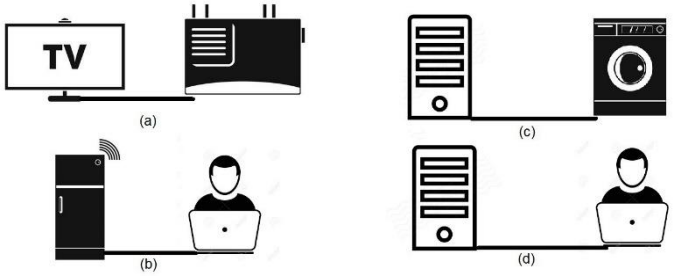

Fig. 7. The Communication Models Types: Device-To-Device, Device to Server, User To Device And User To Server

The device-to-server communication method (Fig. 6-c) is characterized by one side being a reliable, high-computing device with no resource constraints. The last communication model is the user-to-server model (Fig. 6-d), which stands out as the most resource-rich of all communication models. Moreover, both sides in this model can be considered trustworthy by utilizing existing authentication methods.

Beyond the variations in facilities and trust levels across these communication models, it's essential to acknowledge that the type of device on each side of the communication is inherently linked to the specific communication mode.

IV. LITERATURE REVIEW

In this section, we will conduct a review of several lightweight key exchange methods. Subsequently, we will provide three classifications of these methods in Table III,Table **IV**,Table **V**for comparative analysis. Table IIITable provides a comparison of the examined methods based on the functions and operators used. This comparative study will aid in the definition of lightweight methods.

Manuscript received 15-Oct.-2022 and revised 28-Nov.-2022,
Accepted on 29-Dec.-2022
P- ISSN: 2322-4088
E- ISSN: 2322-3936

Table III: Examination different key agreement methods based on requirements of method

Approach	Smart Card	Biometric	Password	Time Synchronization	Nonce/ Random
[21]	YES	NO	YES	NO	YES
[20]	NO	NO	NO	NO	YES
[22]	YES	YES	YES	YES	NO
[23]	YES	NO	YES	YES	NO
[6]	YES	NO	YES	YES	NO
[24]	YES	YES	YES	NO	NO
ALPKA1 [7]	NO	NO	NO	NO	YES
ALPKA2 [7]	NO	NO	NO	NO	YES
[11]	NO	NO	NO	NO	YES
[25]	NO	NO	NO	YES	YES
[26]	YES	YES	YES	YES	YES
[27]	YES	YES	YES	YES	YES
[28]	NO	NO	NO	NO	YES
[8]	YES	YES	YES	YES	YES
[29]	NO	NO	NO	YES	YES
[30]	YES	YES	YES	YES	YES
P1 [31]	NO	NO	YES	YES	YES
P2 [31]	YES	NO	YES	YES	YES
P3 [31]	YES	YES	YES	YES	YES
[32]	YES	YES	YES	YES	YES
[33]	NO	NO	YES	YES	YES
[34]	NO	NO	YES	YES	NO
[35]	NO	YES	YES	YES	NO
[36]	NO	NO	YES	YES	NO
[37]	NO	NO	YES	YES	NO
[38]	NO	YES	YES	YES	NO
[39]	NO	NO	YES	YES	NO
[40]	NO	YES	YES	YES	NO
[41]	YES	YES	YES	YES	YES
[42]	NO	YES	YES	YES	YES
[43]	NO	YES	YES	YES	NO
[5]	NO	NO	NO	NO	YES
[47]	NO	YES	NO	YES	YES
[48]	NO	NO	YES	YES	YES
[49]	NO	NO	YES	YES	YES

[50]	NO	YES	NO	NO	YES
[51]	NO	NO	YES	YES	YES
[52]	NO	YES	YES	YES	YES
[53]	NO	YES	YES	YES	YES
[54]	NO	NO	NO	NO	YES
[55]	NO	NO	NO	YES	YES
[56]	YES	YES	YES	YES	YES
[57]	NO	NO	NO	YES	YES
[58]	NO	NO	YES	NO	YES
[59]	NO	YES	NO	YES	YES
[60]	NO	NO	NO	NO	YES
[61]	NO	NO	YES	NO	YES
[62]	NO	NO	NO	YES	YES
[63]	NO	NO	NO	YES	YES
[64]	NO	YES	NO	YES	YES

Table IV focuses on evaluating the same methods in terms of the protocol's requirements and functional prerequisites. These requirements may encompass the need for a smart card or a timestamp, among others.

Furthermore, in Table V, we present a classification that delves into both the structural and functional characteristics of these protocols.

Table IV: Examination different key agreement methods based on used Operators

Functions used Approach	The exclusive or operation ⊕	The concatenation operation ‖	The one-way hash function h()	MAC	key generation /extraction function
[21]	YES	YES	YES	NO	-
[20]	NO	YES	YES	YES	HKDF
[22]	YES	YES	YES	YES	BK( )
[23]	YES	YES	YES	NO	-
[6]	YES	YES	YES	NO	-
[24]	YES	YES	YES	NO	Hash
ALPKA1 [7]	YES	YES	YES	NO	-
ALPKA2 [7]	YES	YES	YES	NO	-
[11]	YES	YES	YES	YES	FE
[25]	YES	NO	YES	NO	-

Manuscript received 15-Oct.-2022 and revised 28-Nov.-2022,
Accepted on 29-Dec.-2022
P- ISSN: 2322-4088
E- ISSN: 2322-3936

[26]	YES	YES	YES	NO	-
[27]	YES	YES	YES	NO	-
[28]	NO	YES	YES	YES	EC
[8]	YES	YES	YES	NO	-
[29]	YES	YES	YES	NO	-
[30]	YES	YES	YES	NO	-
[31]	YES	YES	YES	NO	-
[32]	YES	YES	YES	NO	-
[33]	YES	YES	YES	NO	-
[34]	YES	YES	YES	NO	-
[35]	YES	YES	YES	NO	-
[36]	YES	YES	YES	NO	-
[37]	YES	YES	YES	NO	-
[38]	YES	YES	YES	NO	-
[39]	YES	YES	YES	NO	-
[40]	YES	YES	YES	NO	-
[41]	YES	YES	YES	NO	-
[42]	YES	YES	YES	NO	-
[43]	YES	YES	YES	NO	-
[5]	YES	YES	YES	NO	-
[47]	YES	YES	YES	NO	-
[48]	YES	YES	YES	NO	-
[49]	YES	YES	YES	NO	-
[50]	YES	YES	YES	NO	FE
[51]	YES	YES	YES	NO	-
[52]	YES	YES	YES	NO	ECC
[53]	YES	YES	YES	NO	-
[54]	YES	YES	YES	NO	Ex,DH
[55]	YES	YES	YES	NO	ECC
[56]	YES	YES	YES	NO	-
[57]	YES	YES	YES	NO	ECC,Block chain,CHA
[58]	YES	YES	YES	NO	LC
[59]	YES	YES	YES	NO	ECC
[60]	YES	YES	YES	NO	LC
[61]	YES	YES	YES	NO	GMW
[62]	YES	YES	YES	NO	ECDH
[63]	YES	YES	YES	NO	ECC
[64]	YES	YES	YES	NO	-

ECC: Elliptical Curve Cryptography

ECQ: Elliptic Curve Qu-Vanstone

GMW: Goldreich-Micali-Wigderson

ECDH: Elliptic Curve Diffie-Hellman

CM: Chaotic Map

LC: Lattice Cryptography

FE: Fuzzy Extractor

DH: Diffie-Hellman

Table V: Examination of the different lightweight key agreement methods based on the structural properties

Approach	Exchange Type	Level of Participation	Key Transfer	Tird Party	Role of TP	Re Encryption	Relation of KDC	Model
[21]	Key Agreement	1 Party	Implicit	NO	No Proxy	NO	Non-Interactive	U-S
[20]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	D-D
[22]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	U-D
[23]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	U-S
[6]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	U-S
[24]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-S
ALPKA1 [7]	Key Agreement	1 Party	Implicit	YES	Proxy	Yes	Non-Interactive	D-D
ALPKA2 [7]	Key Agreement	1 Party	Implicit	YES	Proxy	Yes	Interactive	D-D
[11]	Key Agreement	2 Party	Explicit	YES	Proxy	Yes	Non-Interactive	D-D
[25]	Key Agreement	2 Party	Explicit	YES	Gateway	NO	Non-Interactive	D-S
[26]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	U-S
[27]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	U-S
[28]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-D
[8]	Key Agreement	2 Party	Explicit	YES	Key Proxy	YES	Non-Interactive	U-D
[29]	Key Agreement	2 Party	Explicit	YES	Gateway	NO	Non-Interactive	D-S
[30]	Key Agreement	2 Party	Explicit	YES	key Proxy	YES	Non-Interactive	U-D
[31]	Key Agreement	2 Party	Explicit	YES	key Proxy	YES	Non-Interactive	U-D
[32]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	U-D
[33]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	D-S
[34]	Key Agreement	2 Party	Explicit	YES	Gateway	NO	Non-Interactive	D-S
[35]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Interactive	D-S
[36]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-D
[37]	Key Agreement	2 Party	Explicit	YES	Key Proxy	YES	Non-Interactive	D-D
[38]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Interactive	U-D
[39]	Key Agreement	2 Party	Explicit	YES	No Proxy	NO	Non-Interactive	U-S
[40]	Key Agreement	2 Party	Explicit	YES	Gateway	NO	Non-Interactive	U-S
[41]	Key Agreement	2 Party	Explicit	YES	Proxy	NO	Non-Interactive	U-S

Manuscript received 15-Oct.-2022 and revised 28-Nov.-2022,
Accepted on 29-Dec.-2022
P- ISSN: 2322-4088
E- ISSN: 2322-3936

[42]	Key Agreement	2 Party	Explicit	YES	Proxy	NO	Non-Interactive	U-D
[44]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-D
[43]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Interactive	D-S
[45]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-S
[46]	Key Agreement	2 Party	Explicit	NO	No Proxy	YES	Non-Interactive	D-S
[5]	Key Agreement	1 Party	Implicit	YES	Proxy	YES	Interactive	D-D
[47]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-S
[48]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Interactive	D-S
[49]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-S
[50]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	D-S
[51]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	D-S
[52]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	U-D
[53]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	U-D
[54]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-D
[55]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Interactive	D-S
[56]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Interactive	U-D
[57]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Non-Interactive	D-S
[58]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-D
[59]	Key Agreement	2 Party	Explicit	YES	Proxy	YES	Interactive	D-D
[60]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-D
[61]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	U-S
[62]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	D-S
[63]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Non-Interactive	U-S
[64]	Key Agreement	2 Party	Explicit	NO	No Proxy	NO	Interactive	D-S

V. CONCLUSION

This study provided a comprehensive exploration of key agreement methods in the context of the Internet of Things. Instead of just presenting a simple classification of protocols, we examined various aspects, including communication models and resource requirements. One of the key contributions was the introduction of new classifications aimed at helping select suitable protocols for different IoT environments. Our analysis included criteria such as procedures, communication models, resource usage, and functional characteristics, alongside a review of lightweight key agreement methods that fit these classifications. We hope this research will be a valuable resource for researchers in the field of key management, offering insights for making informed choices based on specific criteria and environmental needs. It also serves as a guide for those developing new key agreement methods, encouraging consideration of the diverse aspects discussed. This comprehensive study aims to inspire further research and progress in the evolving landscape of key agreement methods for the IoT.

References

[1] A. Kumar, "Survey and Taxonomy of Key Management Protocols for Wired and Wireless Networks," *International Journal of Network Security & Its Applications,* 2012.

[2] M. Just, "Key agreement," in *Encyclopedia of Cryptography and Security*, Boston, Springer, 2005, pp. 319-344.

[3] B. Vesteras, "Analysis of Key Agreement Protocols," Department of Computer Science and Media Technology, 2006.

[4] K. T. Nguyen, M. Laurent and N. Oualha, "Survey on secure communication protocols for the internet of things," *Elsevier BV,* pp. 17-31, 2015.

[5] H. H. S. J. M. R. Rasoul Roustaei, "Implicit Lightweight Proxy Based Key Agreement for the Internet of Things (ILPKA)," *Wireless Personal Communications,* vol. 130, pp. 1833-1860, 2023.

[6] M. Nikooghadam, R. Jahantigh and H. Arshad, "A lightweight authentication and key agreement protocol preserving user anonymity," *Multimedia Tools and Applications,* p. 13401–13423, 2017.

[7] M. L. A. D. J. An Braeken, "Anonymous lightweight proxy based key agreement for IoT (ALPKA)," *Wireless Personal Communications,* vol. 106, pages 345–364, 2019.

[8] J. Srinivas, A. K. Das, M. Wazid and N. Kumar, "Anonymous Lightweight Chaotic Map-Based Authenticated Key Agreement Protocol for Industrial Internet of Things," *IEEE Trans. Dependable and Secure Computing,* pp. 1133-1146, 2020. DOI:10.1109/TDSC.2018.2857811

[9] M. Green and G. Ateniese, "Identity-Based Proxy Re-encryption," in *Applied Cryptography and Network Security. ACNS 2007*, Berlin, Heidelberg., 2007.

[10] I. Bhattarai, C. Pu, K.-K. Raymond Choo and D. Korać, "A Lightweight and Anonymous Application-Aware Authentication and Key Agreement Protocol for the Internet of Drones," *IEEE Internet of Things Journal,* pp. 19790-19803, 2024.

[11] L. Wei, J. Cui, H. Zhong, I. Bolodurina and L. Liu, "A Lightweight and Conditional Privacy-Preserving Authenticated Key Agreement Scheme With Multi-TA Model for Fog-Based VANETs," *IEEE Transactions on Dependable and Secure Computing,* pp. 422-436, 2023.

[12] D. He, Y. Cai, S. Zhu, Z. Zhao, S. Chan and M. Guizani, "A Lightweight Authentication and Key Exchange Protocol With Anonymity for IoT," *IEEE Trans. Wireless Communications,* pp. 7862-7872, 2023.

[13] L. Chen, J. Wang, B. Yin, K. Yu and J. Han, "A provably secure and PUF-based authentication key agreement scheme for cloud-edge IoT," *China Communications,* pp. 198-216, 2023.

[14] M. Zia, M. S. Obaidat, K. Mahmood, S. Shamshad, M. A. Saleem and S. A. Chaudhry, "A Provably Secure Lightweight Key Agreement Protocol for Wireless Body Area Networks in Healthcare System," *IEEE Transactions on Industrial Informatics,* pp. 1683-1690, 2023.

[15] S. Itoo, A. A. Khan, M. Ahmad and M. J. Idrisi, "A Secure and Privacy-Preserving Lightweight Authentication and Key Exchange Algorithm for Smart Agriculture Monitoring System," *IEEE Access,* pp. 56875-56890, 2023.

[16] Y. Ming, P. Yang, H. Mahdikhani and R. Lu, "A Secure One-to-Many Authentication and Key Agreement Scheme for Industrial IoT," *IEEE Systems Journal,* pp. 2225-2236, 2023.

[17] H. Wang, J. Wen, J. Liu and H. Zhang, "ACKE: Asymmetric Computing Key Exchange Protocol for IoT Environments," *IEEE Internet of Things Journal,* pp. 18273-18281, 2023.

Manuscript received day-Month-year and revised day-Month-year,
Accepted on day- month-year

P- ISSN: 2322-4088
E- ISSN: 2322-3936

[18] K. A. Yadav and P. Vijayakumar, "An Efficient Lightweight Mutual Authentication and Key Exchange Protocol for Roaming Vehicle," *IEEE Access,* pp. 49540-49548, 2023.

[19] Y. Han, H. Guo, J. Liu, B. B. Ehui, Y. Wu and S. Li, "An Enhanced Multifactor Authentication and Key Agreement Protocol in Industrial Internet of Things," *IEEE Internet of Things Journal,* pp. 16243-16254, 2024.

[20] A. Badshah, G. Abbas, M. Waqas, F. Muhammad, Z. H. Abbas, M. Bilal and H. Song, "Blockchain-Assisted Lightweight Authenticated Key Agreement Security Framework for Smart Vehicles-Enabled Intelligent Transportation System," *IEEE Transactions on Automation Science and Engineering,* pp. 1-15, 2024.

[21] D. S. Gupta, "PiLike: Post-Quantum Identity-Based Lightweight Authenticated Key Exchange Protocol for IIoT Environments," *IEEE Systems Journal,* pp. 15-23, 2024.

[22] S. Manivannan, R. S. Chakraborty, I. Chakrabarti and J. Rangasamy, "Practical and Efficient PUF-Based Protocol for Authentication and Key Agreement in IoT," *IEEE Embedded Systems Letters,* pp. 118-121, 2024.

[23] G. Wei, K. Fan, K. Zhang, H. Wang, H. Li and Y. Yang, "Quantum-Safe Lattice-Based Certificateless Anonymous Authenticated Key Agreement for Internet of Things," *IEEE Internet of Things Journal,* pp. 9213-9225, 2024.

[24] M. K. Hasan, M. M. Hasan, A. K. Budati, S. Islam, N. Safie, F. R. A. Ahmed, K. A. A. Bakar, N. B. M. Babiker and T. M. Ghazal, "A hybrid key agreement scheme utilized elliptic curve Diffie-Hellman for IoT based advanced metering environment," *Earth Science Informatics,* 2024.

[25] D. K. Singh, "Lightweight secure authentication and key agreement technique for smart grid," *Peer-to-Peer Networking and Applications,* 2024.

[26] B. L. P. S. J. P. Prarthana J. Mehta, "PF-AKA: PUF-FSM based Authentication and Key Agreement Framework for IoT based Smart Grid Networks," *Cluster Computing,* 2024.

[27] I. Simsek, "Authentication, Authorization, Access Control, and Key Exchange in Internet of Things," *Association for Computing Machinery,* 2024.

[28] M. N. J. Arshad, "A lightweight authentication and key agreement protocol preserving user anonymity," *Multimedia Tools and Applications,* 2017.

[29] M. G. a. G. Ateniese, "Identity-Based Proxy Re-encryption," *Springer Berlin Heidelberg,* pp. 288-306, 2007.

[30] V. R. Thakare and J. S. K, "Advanced Methodologies and Technologies in System Security, Information Privacy, and Forensics," *IGI Global,* pp. 71-81, 2019.

[31] N. O. M. L. Kim Thuat Nguyen, "Authenticated Key Agreement Mediated by a Proxy Re-encryptor for the Internet of Things," *Computer Security ESORIC,* 2016.

[32] G. Itkis, "Forward security, adaptive cryptography: Time evolution," 2004.

[33] H. Alzaid, D. Park, J. G. Nieto, C. Boyd and E. Foo, "A Forward & Backward Secure Key Management in Wireless Sensor Networks for PCS/SCADA," in *Advances in Wireless Technologies and Telecommunication*, IGI Global, 2011, pp. 41-60.

[34] D. Forsberg, "Use Cases of Implicit Authentication and Key Establishment with Sender and Receiver {ID} Binding," 2007.

[35] W. Diffie, P. C. V. Oorschot and M. J. Wiener, "Authentication and authenticated key exchanges,," *Designs, Codes and Cryptography,* pp. 107-125, 1992.

[36] L. L. a. A. Menezes, M. Qu, J. Solinas and S. Vanstone, "An Efficient Protocol for Authenticated Key Agreement," *Designs, Codes and Cryptography,* pp. 119-134, 2003.

[37] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings," in *16th {IEEE} Computer Security Foundations Workshop, 2003. Proceedings*, 2023.

[38] A. Bin-Rabiah, K. K. Ramakrishnan, E. Liri and K. Kar, "A Lightweight Authentication and Key Exchange Protocol for IoT," in *Proceedings 2018 Workshop on Decentralized {IoT} Security and Standards*, 2018.

[39] A. Braeken, "Efficient Anonym Smart Card Based Authentication Scheme for Multi-Server Architecture," *International Journal of Smart Home,* pp. 177-184, 2015.

[40] S. A. a. S. F. A. a. H. Mala, "A new lightweight authentication and key agreement protocol for Internet of Things," in *2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology ISCISC*, 2016.

[41] S. Kumari, M. K. Khan and X. Li, "An improved remote user authentication scheme with key agreement," *Computers & Electrical Engineering,* pp. 1997-2012, 2014.

[42] A. Irshad, S. A. Chaudhry, S. Kumari, M. Usman, K. Mahmood and M. S. Faisal, "An improved lightweight multiserver authentication scheme," *International Journal of Communication Systems,* 2017.

[43] C.-M. Chen, B. Xiang, T.-Y. Wu and K.-H. Wang, "An Anonymous Mutual Authenticated Key Agreement Scheme for Wearable Sensors in Wireless Body Area Networks," *Applied Sciences,* p. 1074, 2008.

[44] D. Mishra, A. K. Das and S. Mukhopadhyay, "A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards," *Expert Systems with Applications,* pp. 8129-8143, 2014.

[45] K. C. Baruah, S. Banerjee, M. h. P. Dutta and C. T. Bhunia, "An Improved Biometric-based Multi-server Authentication Scheme Using Smart Card," *International Journal of Security and Its Applications,* pp. 397-408, 2015.

[46] M. A. S. Jr., M. V. Silva, R. C. Alves and T. K. Shibata, "Lightweight and escrow-less authenticated key agreement for the internet of things," *Computer Communications,* pp. 43-51, 2017.

[47] A. Ostad-Sharif, M. Nikooghadam and D. Abbasinezhad-Mood, "Design of a lightweight and anonymous authenticated key agreement protocol for wireless body area networks," *International Journal of Communication Systems,* 2019.

[48] S. Shin and T. Kwon, "A Lightweight Three-Factor Authentication and Key Agreement Scheme in Wireless Sensor Networks for Smart Homes," *Sensors,* p. 2012, 2019.

[49] I. S.-G. a. A. Rivero-Garcia, M. Burmester, J. Munilla and P. Caballero-Gil, "Secure lightweight password authenticated key exchange for heterogeneous wireless sensor networks," *Information Systems,* p. 101423, 2020.

[50] J. Mo and H. Chen, "A Lightweight Secure User Authentication and Key Agreement Protocol for Wireless Sensor Networks," *Security and Communication Networks,* pp. 1-17, 2019.

[51] Z. Xu, C. Xu, W. Liang, J. Xu and H. Chen, "A Lightweight Mutual Authentication and Key Agreement Scheme for Medical Internet of Things," *IEEE Access,* pp. 53922-53931, 2019.

[52] W. Tsu-Yang, L. Wang, X. Guo, Y.-C. Chen and S.-C. Chu, "SAKAP: SGX-Based Authentication Key Agreement Protocol in IoT-Enabled Cloud Computing," *Sustainability,* 2022.

[53] Y. Zhang and F. Wen, "A Lightweight Secure and Efficient Authentication and Key Agreement Protocol for VANET," in *IOP Conference Series: Earth and Environmental Science*, 2019.

[54] C. Wenchao, R. Cheng, K. Wu, Y. Su and Y. Lei, "A Certificateless Authenticated Key Agreement Scheme for the Power IoT," *Energies,* 2021.

[55] C.-M. Chen, X. Deng, W. Gan, J. Chen and S. K. H. Islam, "A secure blockchain-based group key agreement protocol

Manuscript received 15-Oct.-2022 and revised 28-Nov.-2022,
Accepted on 29-Dec.-2022
P- ISSN: 2322-4088
E- ISSN: 2322-3936

for iot," *The Journal of Supercomputing,* 2021.

[56] Y. Yicheng, L. H. and J. Chu, "A Secure Authentication and Key Agreement Scheme for IoT-Based Cloud Computing Environment," *Symmetry,* 2022.

[57] M. Safkhani, N. Bagheri, S. Kumari, H. Tavakoli, S. Kumar and J. Chen, "RESEAP: An ECC-Based Authentication and Key Agreement Scheme for IoT Applications," *IEEE Access,* pp. 200851-200862, 2020.

[58] B. A. Alzahrani, "Secure and Efficient Cloud-based IoT Authenticated Key Agreement scheme for e-Health Wireless Sensor Networks," *Arabian Journal for Science and Engineering,* 2021.

[59] C.-T. Chen, C.-C. Lee and I.-C. Lin, "Efficient and secure three-party mutual authentication key agreement protocol for WSNs in IoT environments," *PLOS ONE ,* 2020.

[60] R. Vinoth, L. J. Deborah, P. Vijayakumar and N. Kumar, "Secure Multifactor Authenticated Key Agreement Scheme for Industrial IoT," *IEEE Internet of Things Journal,* pp. 3801-3811, 2021.

[61] S. Rana, M. S. Obaidat, D. Mishra, A. Mishra and Y. S. Rao, "Efficient design of an authenticated key agreement protocol for dew-assisted IoT systems," *The Journal of Supercomputing,* 2022.

[62] V. Thakur, G. Indra, N. Gupta, P. Chatterjee, O. Said and A. Tolba, "Cryptographically secure privacy-preserving authenticated key agreement protocol for an IoT network: A step towards critical infrastructure protection," *Peer-to-Peer Networking and Applications,* p. 206–220, 2022.

[63] A. Braeken, "Authenticated key agreement protocols for dew-assisted iot systems," *The Journal of Supercomputing,* 2022.

[64] S. Rostampour, N. Bagheri, Y. Bendavid, M. Safkhani, S. Kumari and J. J. P. C. Rodrigues, "An Authentication Protocol for Next Generation of Constrained IoT Systems," *IEEE Internet of Things Journal,* pp. 21493-21504, 2022.