American Journal of Engineering and Applied Sciences

Original Research Paper

A Secure Unicast/Multicast Authentication and Key Distribution Protocols for Smart Grid

[1]Samah Mohamed Sayed, [2,3]Heba Kamal Aslan and [1]Ayman Mohamed Hassan

[1]Department of Electronics and Communication, Benha Engineering University, Egypt
[2]Department of Informatics Science, Faculty of Information Technology and Computer Science, Nile University, Egypt
[3]Department of Informatics, Electronics Research Institute, Egypt

Article history
Received: 14-10-2022
Revised: 03-11-2022
Accepted: 04-11-2022

Corresponding Author:
Samah Mohamed Sayed
Department of Electronics and Communication, Benha Engineering University, Egypt
Email: samah_meral02@yahoo.com

Abstract: This study discusses the problem of key distribution protocol in IoT systems especially smart grids for two modes of communication: Unicast and multicast. In the present paper, we suggested two protocols for key distribution for both unicast and multicast communication. The proposed schemes are based on symmetric key encryption with cryptographic primitives: Hashes and nonces. In addition, the multicast key distribution protocol is based on a logical key hierarchy to reduce communication and computation overheads in case of any member change. The proposed solutions are compared to other protocols based on communication and computation overheads, and the ability to resist well-known attacks. The comparison shows that the proposed protocols have the lowest overheads while resisting known attacks. To ensure the correctness and security of the proposed protocol, it is analyzed using a logical tool (BAN logic). The analysis illustrates that the proposed protocols are free from bugs or redundancies.

Keywords: IoT Security, Key Management, Symmetric Key Distribution, SM Modes of Operations, Unicast Key Distribution, Multicast Key Distribution, Broadcast Key Distribution, BAN Logic

Introduction

Nowadays, the Internet of Things (IoT) allows cooperation between devices characterized by low resources. These devices can communicate messages, compute collaboratively, and then, make decisions. The designer has to deal with a list of IoT challenges including Minimizing power consumption, the best utilization of battery, constrained memory, and security (Morshed Aski *et al*., 2020; Mohammadali *et al*., 2016; Ali *et al*., 2019).

A smart electric grid is considered one of the major IoT applications. It incorporates the exchange of information through the advanced two way-communication. This information must be delivered securely so that only authorized entities can interpret these messages. This could be achieved using encryption algorithms. The main contributor to realizing security is the secure exchange of keys used for encryption (Al-Waisi and Agyeman, 2018), a function referred to as "key distribution protocol-KDP".

Key distribution protocols must be characterized by the following: Control, efficiency, economy, reliability, safety, and security (Mohamed *et al*., 2021; Eldefrawy *et al*., 2018). Key distribution can be classified into two types: Unicast key distribution protocols, and broadcast/multicast key distribution protocols. The main requirements for both are: Low computation overhead, low communication overhead, low storage, and the need for a key update and rekeying most simply. Unicast key distribution protocols are divided into the following categories: Symmetric-based, asymmetric-based, and hybrid-based key distribution protocols (Lee and Lee, 2018; Abualghanam *et al*., 2019). While broadcast/multicast key distribution protocols are divided into centralized, distributed, and contributory key distribution protocols (Rafaeli and Hutchison, 2003).

The above-mentioned types of protocols are subject to different attacks (Moghadam *et al*., 2020) such as node capture attack which is one of the hazardous attacks in WSNs where an attacker can gain full control of a node, especially the key gateway node. Man-in-the-Middle (MitM) attack is a type of cyber-attack where the attacker intercepts and then controls the entire system conversation by relaying messages between two participants who believe they are communicating directly with each other. The best way to avoid this type of attack is by having a strong encryption mechanism on wireless access points to prevent unauthorized users from joining the network through brute force attacks. A similar type is the replay attack which is a more specific type of man-in-the-middle attack where a hacker intercept transmitted

data, masquerades as an authorized browser, and resends older web requests which will cause system delay. An impersonation attack happens when the attacker pretends to be a legitimate user (or group of users) to gain access to information they are not authorized to read. A brute force attack is a hacking method of trial and error to crack passwords, login credentials, and encryption keys. These methods are being developed into automatic tools to do this guessing faster. A masquerade attack is an online attack in which the attacker masquerades as a legitimate user to gain access to a device with a fake identity, to gain unauthorized access to personal computer information through legitimate access identification. Anonymity is keeping your identity private, but not your actions. A desynchronization Attack means destroying synchronization between the system participants, which causes a permanent disabling of the authentication capability.

In this study, a secure key distribution protocol for both unicast and broadcast communication. The proposed solution is based on symmetric algorithms for unicast communication and centralized approaches. In addition, in multicast communication, Logical Key Hierarchy (LKH) is used (Lee and Lee, 2018; Dammak *et al*., 2020; Guo *et al*., 2018). The proposed protocols have the following characteristics: They overcome the replay and desynchronization attacks, they have low computation and communication overheads, and finally, it achieves the security goals.

The article is organized as follows: The next section shows a literature review concerning different key distribution protocols (Unicast/broadcast). Then, our proposals for unicast/broadcast communication are illustrated. Next, security analysis and proof of our proposed solutions are provided. Then, a comparative analysis of our proposed solutions with other protocols are depicted. Finally, the paper concludes in the last section.

Literature Review

In literature, many solutions for unicast/broadcast key management protocols have been provided. Key management has been identified in Automatic Meter Infrastructure (AMI) as the main process to ensure authenticity and secrecy under a special setup (Benmalek *et al*., 2018). A report published by the National Institute of Standards and Technology (NIST) mentioned that scenarios to provide security in AMI communications are required. In this section, we review the proposed protocols in the literature and analyze them. Key management protocols are classified according to the communication type into unicast and broadcast protocols. Unicast is classified into symmetric, asymmetric, and hybrid encryption (Lee and Lee, 2018). On the other hand, the broadcast can be classified into centralized, decentralized, and distributed/contributory protocols. In the following paragraph, we give a detailed listing of both unicast and broadcast protocols.

Unicast Key Distribution Protocols

Unicast key distribution protocols are classified into symmetric, asymmetric, and hybrid key distribution protocols. Rabiah *et al*. (2018), authors proposed a key distribution protocol. This protocol is based on the use of symmetric cryptography and nonces in a random frame sequence. This leads to a smaller number of encrypted messages. The keys must be changed periodically using a hash derivation function. However, this protocol suffers from the possibility of conducting a MITM attack and is characterized by a high communication overhead. Cheng *et al*. (2018), the authors proposed a solution known as Long Range (LoRa) with low power consumption. This is a hardware key distribution solution embedded in each smart meter. It is used to automatically update the meter's session key frequently. It is based on using symmetric key algorithms, nonces, and timestamps. It resists replay attacks; however, it suffers from the need for costly hardware. Jeya (2019) proposes a solution based on a secure tree hierarchy algorithm to compute the symmetric key using a tree of hashes. Then, the user verifies the received message using the appended hash. Pandiya *et al*. (2020), the authors proposed a key distribution protocol. This protocol is based on symmetric encryption and simple primitive cryptography tools such as nonce and hashing for security realization with a minimum number of messages and communications. The existence of nonce in each session prevents replay attacks and avoids system desynchronization attack. On the other hand, the main scheme's disadvantage is the repeating of symmetric encryption and decryption between sender and receiver for verification and authentication. Kang *et al*. (2020) developed an analysis and improvement protocol for IoT by identity authentication, random number password, hash, XoR, and timestamp. First, the registration phase is executed, followed by the login phase, and finally the steps of key agreement and authentication. The main drawback of this solution is the need for synchronization. If not realized, this leads to a desynchronization attack.

The following algorithms represent the asymmetric key distribution protocols. Mehibel and Hamadouche (2021) proposed an authentication protocol that uses Elliptic Curve Digital Signature Algorithm (ECDSA) with two random values. The proposal's main advantages are Forward/backward secrecy, resistance to Man in the Middle (MitM) attacks, and impersonation attacks. Its main disadvantage is the high consuming time. Farooq *et al*. (2020) use ID-based authentication and key agreement mechanism for securing communication in AMI by using Elliptic Curve Cryptography (ECC) which is more secure than the selection of random value. In addition, it achieves a

lower execution time than using an elliptic curve digital signature due to the use of bilinear pairing and small size parameters. The main disadvantage is sending the private keys on a public unsecured channel. Mohammadali *et al*. (2016), the authors proposed a solution based on identity-based mechanisms. The protocol is based on ECC and Discrete Logarithm Protocol (DLP) using three phases: Setup, installation, and key agreement. The proposal has a high resilience to the following attacks: Replay attack, impersonation attack, MITM attack, and desynchronization. NIKE$^+$ is the improvement for NIKE by shifting some of the load to AMI Head End (AHE) to reduce the computational cost at the meter's side and complete the management of data collection and interaction. Its main disadvantage is the high communication overhead due to the need for a high number of communication and messages, especially in the key agreement phase for verification. The following algorithms represent the hybrid approach. Choudhary *et al*. (2020) proposed a lightweight mutual authentication and key exchange protocol for the Industrial Internet of Things (IIoT). It is based on ID authentication, port address and location, hash, XoR, timestamp, random secret, and Message Authentication Code (MAC). Kumar *et al*. (2018) shows a scalable scheme named Lightweight Authentication and Key Agreement (LAKA) for SM based on ECC, symmetric encryption, hash, and MAC. The Neighboring Gateway (NAN GW) is used as a trusted entity that performs the off-link tasks with its high memory for storing the whole SMs security parameters. The system needs several NANs GW each one works as a Service Provider (SP) for SMs registration. Gong *et al*. (2019) proposed cyber security protection of a distribution automation system based on a hybrid encryption algorithm. The scheme is based on symmetric, hash, MAC, and an authenticated one-time password which is based on a combination of secret algorithms. The advantages of this proposal include the existence of a dual protection scheme using a distribution automation system and Time Password (OTP). Naseer *et al*. (2020) showed a key transport protocol for AMI based on public key cryptography. The scheme uses symmetric encryption, timestamp, signature, and private and public keys. The high latency overhead calculations, and packet delivery ratio with an increasing number of nodes are considered the main disadvantages of the scheme.

Broadcast Key Distribution Protocols

Group key management is classified into three categories (Abualghanam *et al*., 2019; Rafaeli and Hutchison, 2003): Centralized Group Key Distribution (CGKD), Decentralized Group Key Management (DGKM), and Contributory Group Key Agreement (CGKA).

Centralized approaches are based on one authentication server. Eldefrawy *et al*. (2018), the author's main idea is the use of the Chinese Remainder Theorem (CRT) to provide the group key. In addition, they used the nested hash function and XoRing operations. The proposal lowers the number of exchanged messages and message length by storing secret symmetric seeds among nodes and gateway by applying CRT, then calculating Session Key (SK) in the gateway. It provides forward/backward secrecy, but its main disadvantages are the high computation overhead and the possibility of nodes capture attack. Kumar *et al*. (2020), authors propose a more efficient CGKD protocol with the main aim to minimize the cost of computation at the Key Server (KS) in the key update phase. This is fulfilled by executing one encryption, one addition, and one multiplication in case of a single member joins and one encryption, one division, and one subtraction in case of a single member leaves. In addition, they reduce the KS's storage complexity. Furthermore, the authors described an extension to the CGKD protocol. To provide efficiency and scalability, the proposed protocol is based on clustered trees. The authors compared their protocol with similar protocols, they showed that their protocol dramatically decreased the computation overhead and the storage requirements while having the same communication overhead and storage load for each group member. Other protocols are based on the use of Logical Key Hierarchy (LKH). In these protocols, a tree of the key is used where the leaves represent the keys shared between the users and the server, and the group key is represented by the root. To reduce the required messages and encryption/decryption operations, in case of any change, the remaining keys in the tree are utilized.

Decentralized protocols are based on dividing the members into groups. While the whole system is managed by a central entity, each group is managed by a sub-group manager. This reduces the computations required in case of a member change. Benmalek *et al*. (2018), the authors proposed a scalable solution named Versatile and Scalable AMI (VerSAMI). The base of the VerSAMI system is the use of a structure of the multi-group key graph. Then, the authors improved VerSAMI to VerSAMI$^+$ which adopts One-way Function Trees (OFT) which is the improvement of LKH protocol (adopted in VerSAMI) that allows reducing the number of rekeying messages. Finally, the article solves the main two problems of dynamic changes which are: The high communication overhead and the desynchronization attack. Adusumilli *et al*. (2005), authors proposed a scalable, simple, robust, and efficient protocol. This proposal of Distributed Group Key Distribution (DGKD) solves the main problem of dependency on trusted third parties by distributing the work done to calculate the new group key among all group members. Mittra (1997), authors present lolus as a framework for scalable secure multicast communication. In Iolus, the members are distributed among several subgroups. Thus, the member change will only affect the corresponding subgroup.

The last type is the distributed/contributory key distribution protocols. In these protocols, each member contributes to the calculation of the group key. The main advantage of these protocols is the elimination of the single point of failure. However, they suffer from high communication and computation overheads. Hu *et al*. (2018), the authors use blockchain technology to propose a decentralized key distribution protocol with asymmetric encryption which solves the main problems of centralized servers such as Leakage and integrity of messages in case of servers' compromise. In addition, to trace the tampering behavior, the modified public keys will be appended to the blockchain. In the next section, we detail our proposed solutions for both unicast and broadcast key distribution.

Materials and Methods

In this study, we recommended using the symmetric key distribution protocols. Characteristics of these protocols are summarized in the following two subsections. First section is proposed protocols including unicast and broadcast key distribution. However, each protocol includes the figure of the protocol system model and its main advantages such as join/leave members in broadcast key distribution protocol. The second section is security analysis using BAN logic postulate analysis through applying of logical notations, inference rules of the logic, and logical analysis of the unicast key distribution proposed protocol till concluding that the unicast protocol achieves the goals of authentication without bugs.

The Proposed Protocols

In the following subsections, we detailed our proposed solutions for the cases of unicast and broadcast communication.

Proposed Unicast Key Distribution Protocol

In this subsection, we propose a protocol that is based on symmetric key encryption algorithms to reduce the computation complexity. We recommend using the Advanced Encryption Standard (AES) (Daemen and Rijmen, 2002) which was chosen and tested against well-known attacks by NIST as the symmetric key encryption standard. In addition, we use nonces to withstand both replay and desynchronization attacks.

In our model, we have the following participants:

- A represents the Smart Meter (SM)
- B represents a Gateway (GW)
- AS is an authentication server

In addition, we use the following abbreviations:

- ID_A: *A*'s Identity
- ID_B: *B*'s Identity
- N_A: Nonce generated by *A*
- N_B: Nonce generated by *B*
- K_{AS}: *A* key shared between *A* and *AS*
- K_{BS}: *A* key shared between *B* and *AS*
- K_{AB}: *A* key shared between *A* and *B*

The steps of the proposed unicast protocol are shown in Fig. 1. Our protocol consists of the following phases: Initialization phase, key distribution phase, authentication phase, and message exchange phase. First, in the initialization phase, the initial secret symmetric keys, which will be used with the Authentication Server (AS), are stored in each SM and GW during manufacturing. To overcome the brute force attack, these keys will be updated frequently. AS broadcasts a message to all participants (smart meters and gateways) asking for key updates. Then, all participants calculate the new key $K_{(AS)}^{i+1}$ by applying a hash function (*H*) to the old key $\left(K_{(AS)}^{i}\right)$. This process is called batch rekeying and is illustrated below:

$$K_{(AS)}^{i+1} = H\left(K_{(AS)}^{i}\right)$$
$$K_{(BS)}^{i+1} = H\left(K_{(BS)}^{i}\right)$$

The key distribution begins by sending a request for communication from *A* to *B*. *A* sends *B* a message containing its *ID*, *B*'s *ID*, and a nonce N_A. After the message receipt, B generates N_B and adds it to the received message from *A* before sending it to *AS*. When *AS* receives the message, it verifies the real existence of both ID_A and ID_B which are the two parts of communication. If the verification is successful, *AS* generates a session key (K_{AB}) for secure communication between *A* and *B*. After that *AS* creates a message containing N_A, N_B, and K_{AB}. Then, it encrypts the message twice using K_{AS} (to be sent to *A*) and K_{BS} (to be sent to *B*). Upon receiving the message, *A* decrypts the message using K_{AS}, while, to get the session key, *B* deciphers the message using K_{BS}. For authentication and confirmation of the received key, *A* and *B* check the freshness of the received messages using N_A and N_B respectively. Then, *B* encrypts Na using K_{AB} and sends the message after encryption to *A*. Upon receiving the message, *A* decrypts the message and ensures the freshness of the message using N_A and ensures that *B* can obtain the session key K_{AB}. Similarly, *A* encrypts N_B using K_{AB} and sends the message after encryption to *B*. Upon receiving the message, *B* decrypts the message and ensures the freshness of the message using N_B and ensures that *A* can obtain the session key K_{AB}. After mutual authentication, *A* and *B* can use K_{AB} to communicate securely. In the next subsection, the proposed broadcast key distribution protocol is detailed.

Proposed Two-Level Broadcast Key Distribution Protocol

In some cases, the authentication server needs to broadcast control commands/notifications to all gateways. Similarly, GWs need to broadcast control commands/notifications to all SMs. The main problem in broadcast secure communication is the need to re-key the group key every join/leave. In this subsection, we propose a broadcast key distribution protocol to minimize the number of rekeying messages. Our proposal consists of two levels. The first level which contains the AS and the GWs is flat since the number of GWs is generally small. While the other level consists of the GWs with their corresponding SMs. This level is represented by an LKH tree since the number of SMs is relatively large to minimize the number of rekeying messages. It has to be noted that the main problem in broadcast communication is the leaving nodes rather than the joining nodes (for keeping forward/backward secrecy). In our model, we assume that we have 5 GWs and each GW is responsible for 256 SMs. While Fig. 2 illustrates the flat representation of the AS and the GWs, Fig. 3 show jatas the LKH representation consisting of one GW (GW_1) and the corresponding SMs. We concentrate on the two most important criteria in member join/ leave procedures: The number of encryptions required for the re-key operation, and the communication overhead needed for the re-key message.

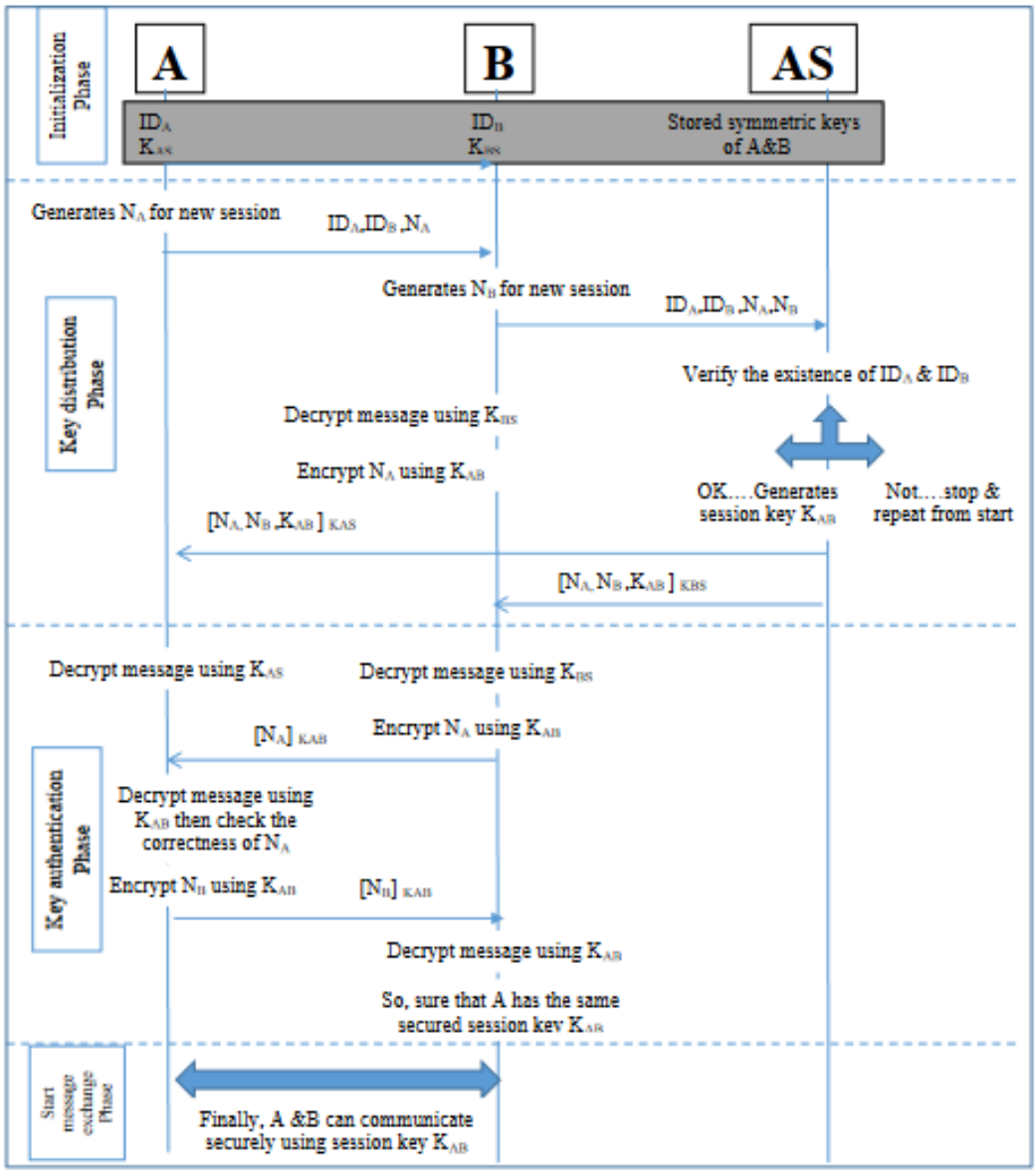

Fig. 1: Unicast key distribution protocol

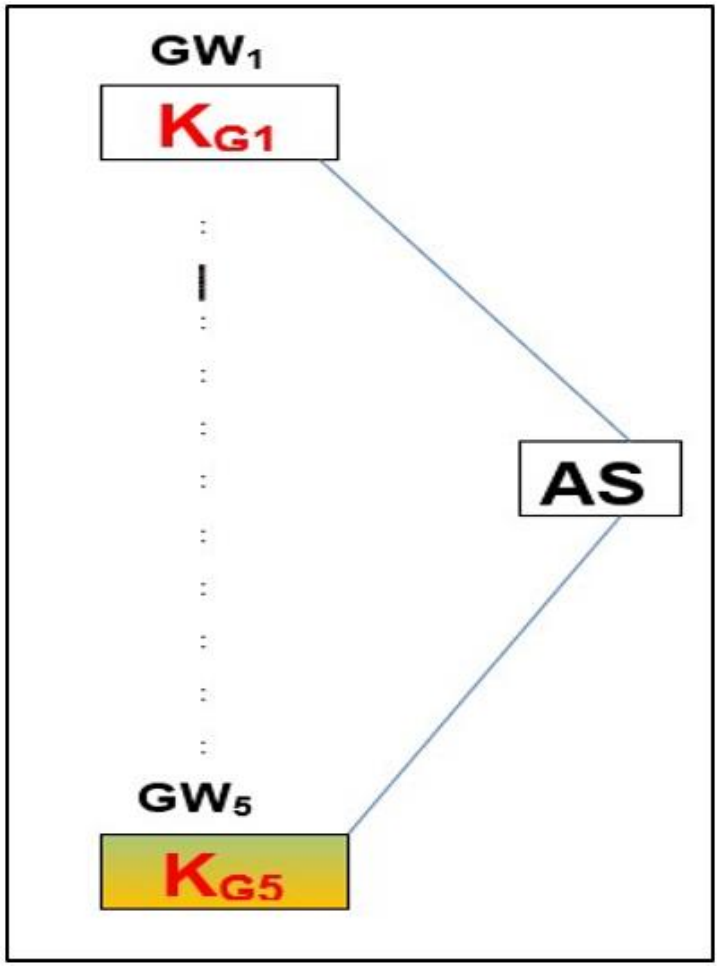

Fig. 2: Flat connection between AS and GW's

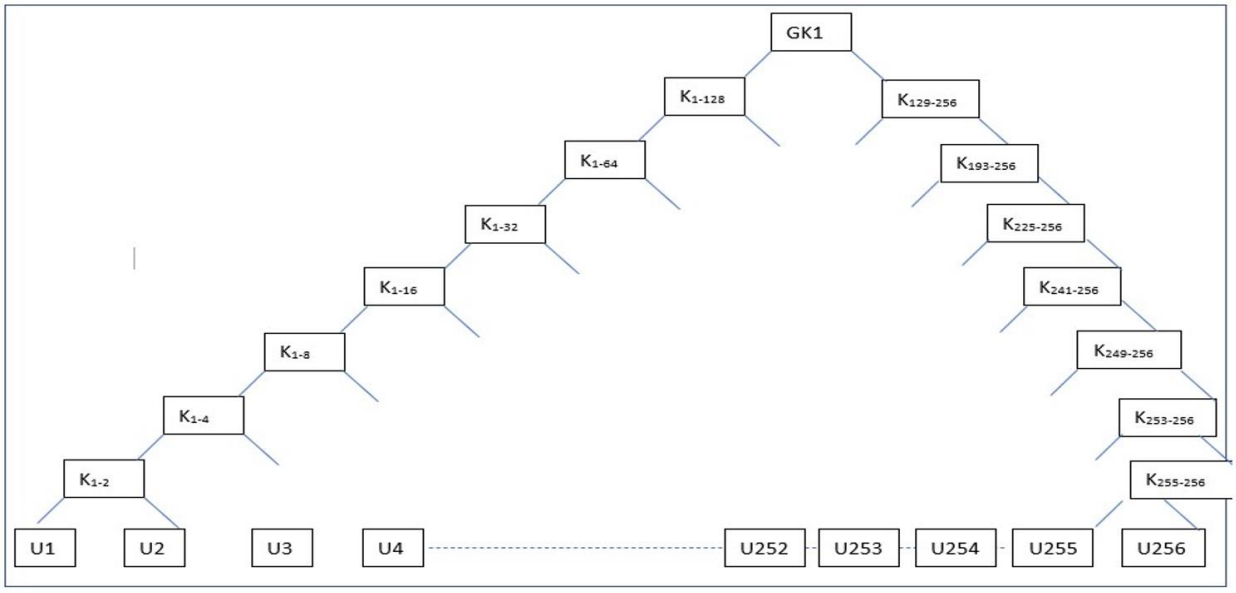

Fig. 3: LKH key distribution proposal

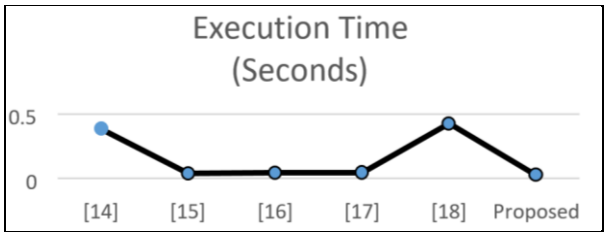

Fig. 4: Execution time comparative chart

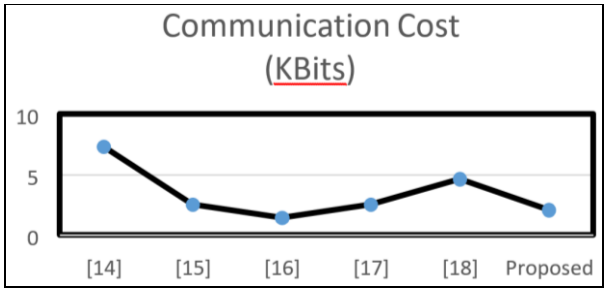

Fig. 5: Communication cost comparative chart

Member Join

Assume a new member U_{256} wants to join the group, the authentication server needs to change the corresponding keys from the root to the leaf by applying the following procedures:

1. Key server generates GK_{1new}, $K_{129\text{-}256\ new}$, $K_{193\text{-}256\ new}$, $K_{225256\ new}$, $K_{241\text{-}256\ new}$, $K_{249\text{-}256\ new}$, $K_{253\text{-}256\ new}$, and $K_{255\text{-}256\ new}$, and broadcasts the following message: $\{GK_{1new}\}$ GK1, $\{K_{129\text{-}256\ new}\}$ $K_{129\text{-}256}$, $\{K_{193\text{-}256\ new}\}$ $K_{193\text{-}256}$, $\{K_{225\text{-}256\ new}\}$ $K_{225\text{-}256}$, $\{K_{241\text{-}256}$ new$\}$ $K_{241\text{-}256}$, $\{K_{249\text{-}256\ new}\}$ $K_{249\text{-}256\ new}$, $\{K_{253\text{-}256\ new}\}$ $K_{253\text{-}256}$, $\{K_{255\text{-}256\ new}\}$ $K_{255\text{-}256}$, $\{G_{K1new}$, $K_{129\text{-}256\ new}$, $K_{193\text{-}256\ new}$, $K_{225\text{-}256\ new}$, $K_{241\text{-}256\ new}$, $K_{249\text{-}256\ new}$, $K_{253\text{-}256\ new}$, $K_{255\text{-}256\ new}\}$ K_{256}
2. Then, all users U_1, U_2, ……U_{256} get GK_{1new} by decrypting the part encrypted using the old value K_1.
3. At U_{129}-U_{256}, $K_{129\text{-}256\ new}$ is obtained $K_{129\text{-}256}$.
4. At U_{193}-U_{256}, $K_{193\text{-}256\ new}$ is obtained using $K_{193\text{-}256}$
5. At U_{225}-U_{256}, $K_{225\text{-}256\ new}$ is obtained using $K_{225\text{-}256}$
6. At U_{241}-U_{256}, $K_{241\text{-}256}$ new is obtained using $K_{241\text{-}256}$
7. At U_{249}-U_{256}, $K_{249\text{-}256}$ new is obtained using $K_{249\text{-}256}$
8. At U_{253}-U_{256}, $K_{253\text{-}256\ new}$ is obtained using $K_{253\text{-}256}$
9. At U_{255}-U_{256}, $K_{255\text{-}256\ new}$ is obtained using $K_{255\text{-}256}$
10. Finally, U_{257} obtains G_{K1new}, $K_{129\text{-}256\ new}$, $K_{193\text{-}256\ new}$, $K_{225\text{-}256\ new}$, $K_{241\text{-}256\ new}$, $K_{249\text{-}256\ new}$, $K_{253\text{-}256\ new}$, $K_{255\text{-}256\ new}$, and $K_{255\text{-}256\ new}$ using K_{256}

In our proposed protocol, the tree height (h) has a value of 8. Therefore, in the case of a member joining, the required number of encryption is equal to 2 h = 16 encryptions. On the other hand, the re-key message length is 2 h keys, assuming keys of 128 bit. Therefore, the length of the transmitted message is 2048 bits (Aslan, 2005).

Member Leave

Assume that member U_{256} leaves the group, the authentication server has to change the corresponding keys from the root to the leaf by applying the following procedures:

1. Key server generates GK_{1new}, $K_{129\text{-}256\ new}$, $K_{193\text{-}256\ new}$, $K_{225256\ new}$, $K_{241\text{-}256\ new}$, $K_{249\text{-}256\ new}$, $K_{253\text{-}256\ new}$, and $K_{255\text{-}256\ new}$, then, the following is broadcasted: $K_{255\text{-}256\ new}\}K_{255}$, $\{K_{253\text{-}256\ new}\}$ $K_{255\text{-}256\ new}$, $\{K_{253\text{-}256\ new}\}$ $K_{253\text{-}254\ new}$, $\{K_{249\text{-}256\ new}\}$ $K_{253\text{-}256\ new}$, $\{K_{249\text{-}256\ new}\}$ $K_{249\text{-}252}$, $\{K_{241256\ new}\}$ $K_{249\text{-}256\ new}$, $\{K_{241\text{-}256\ new}\}$ $K_{241\text{-}248}$, $\{K_{225\text{-}256\ new}\}$ K, $\{K_{225\text{-}256\ new}\}$ $K_{225\text{-}240}$, $\{K_{193\text{-}256\ new}\}$ $K_{225\text{-}256new}$,$\{K_{193\text{-}256\ new}\}$ $K_{225\text{-}256\ new}$, $\{K_{193\text{-}256\ new}\}$ $K_{193\text{-}224}$, $\{K_{129\text{-}256\ new}\}$ $K_{193256new}$, $\{K_{129\text{-}256\ new}\}$ $K_{129\text{-}192}$, $\{GK_{1\ new}\}$ $K_{129\text{-}256\ new}$, $\{GK_{1\ new}\}$ $K_{1\text{-}128}$.
2. Then, all users obtain the new keys using the corresponding keys.

Therefore, in our proposed protocol where h = 8, the number of encryptions, in the leave case, equals 15 encryptions. The re-key message length is 2 h-1 keys, assuming keys bits 128 bit. Thus, the length of the transmitted message equals 1920 bits (Aslan, 2005) instead of 32640 bits in the case of flat distribution.

Security Analysis

In this section, the security analysis of our unicast protocol has been conducted using the BAN logic. We will, first, introduce the logical notations of the logic. Then, the logic rules are illustrated. Next, the logic is applied to our protocol.

Logical Notations of the Logic

To describe the logic, the following notations will be used: *P* and *Q* range over principals, *X* and *Y* are statements, and *K* denotes the encryption key. In addition, certain symbols of logic are used as follows (Aslan, 2004):

$P \equiv X$: = Read" *P* believes *X*", it means *P* may act as though *X* is true

$P \triangleright X$: = Read "*P* sees *X*", it means someone has sent a message containing *X* to *P*, who can read and repeat it

$P \sim X$: = Read "P has jurisdiction over *X*", it means *P* has authority on *X* and should be trusted on this matter

(*X*): = Read "*X* is fresh", it means *X* has not been sent in a message at any time before the current run of the protocol

$P \overset{K}{\longleftrightarrow} Q$: = Read "*P* and *Q* has *K* as a shared key between them", it means *K* is only known to *P* and *Q*

$P \Leftrightarrow Q$ = Read" The formula *X* is a secret only known to *P* and *Q*", it means *P* and *Q* may use *X* to prove their identities to one another

$[X]_K$ = Read "*X* is encrypted under *K*", which means the formula *X* is encrypted by *K*

$<X>Y$: = Read "*X* is combined with the formula *Y*", which means Y plays the role of proof of origin for *X*

Inference Rules of the Logic

The above statements are combined to deduce some rules on which the logic is based. In the following paragraphs, the rules that will be used to analyze the protocol are explained.

Seeing Rules

If *P* believes that *K* is a shared key between it and *Q*, and *P* sees a message encrypted under *K* under the condition that *P*≠*Q*, this implies that *P* believes that at some time *Q* once said *X*:

$$\frac{P \equiv P \xrightarrow{K} Q, P > [x]K}{P \equiv Q \sim X}, \quad P \neq Q \qquad \text{Rule (1)}$$

The Verification Rules

If P believes that *X* is a recent message and that *Q* once said *X*, then *P* believes that *Q* believes *X*. This is the only postulate that transforms ~ to ≡:

$$\frac{P \equiv \#(X), P \equiv Q \sim X}{P \equiv Q \equiv X} \qquad \text{Rule (2)}$$

The Jurisdiction Rules

If *P* believes that *Q* has jurisdiction over *X*, and *P* believes that *Q* believes *X*, then *P* believes *X*:

$$\frac{P \equiv Q \Rightarrow X, P \equiv Q \equiv X}{P \equiv X} \qquad \text{Rule (3)}$$

Logical Analysis of the Unicast Key Distribution Proposed Protocol

The first step in the analysis will be the determination of initial assumptions, what keys are initially shared between principals, which principal may use nonces, and which principals are trusted in a certain way as follow:

$$Assumption\ 1: A \equiv A \xrightarrow{K_{AS}} AS \qquad (1)$$

$$Assumption\ 2: B \equiv B \xrightarrow{K_{BS}} AS \qquad (2)$$

$$Assumption\ 3: A \equiv \#(N_a) \qquad (3)$$

$$Assumption\ 4: B \equiv \#(N_B) \qquad (4)$$

$$Assumption\ 5: A \equiv AS \Rightarrow A \xrightarrow{K_{AB}} B \qquad (5)$$

$$Assumption\ 6: B \equiv AS \Rightarrow A \xrightarrow{K_{AB}} B \qquad (6)$$

After making assumptions, we will present the protocol messages as follows:

$$Step1(A \rightarrow B): ID_A, ID_B, N_A$$
$$Step2(B \rightarrow AS): ID_A, ID_B, N_A, N_B$$
$$Step3: (AS \rightarrow A): [N_A, N_B, K_{AB}]K_{AS}$$
$$Step4: (AS \rightarrow B): [N_A, N_B, K_{AB}]K_{BS}$$
$$Step5: (B \rightarrow A): [N_A]K_{AB}$$
$$Step6: (A \rightarrow B): [N_B]K_{AB}$$

where:

A : User/Smart meter *A*
B : Gate Way (GW) *B*
AS : Authentication Server
ID_A : Smart meter identity
ID_B : GW identity
N_A : *A*'s nonce
N_B : *B*'s nonce
K_{AS} : Symmetric key between *A* and *AS*
K_{BS} : Symmetric key between *B* and *AS*
K_{AB} : Session key between *A* and *B*

The steps of the above protocol messages will be transformed into logical idealized steps as follow:

$$Step3: \left[N_A, N_B, A \xleftrightarrow{K_{AB}} B\right]_{KAS}$$
$$Step4: \left[N_A, N_B, A \xleftrightarrow{K_{AB}} B\right]_{KBS}$$
$$Step5: \left[N_A, A \xleftrightarrow{K_{AB}} B\right]_{KAB}$$
$$Step6: \left[N_B, A \xleftrightarrow{K_{AB}} B\right]_{KAB}$$

where the first two steps from the protocol messages are omitted because it does not contain encryption and it does not contribute to the logical properties of the protocol. We will consider the authentication and key distribution is complete between two principals *A* and *B* if there is a key" K_{AB}" that each principal believes that it shares with the other principal and that each believes that the other principal believes in this key. Thus, the authentication and key distribution between *A* and *B* will be complete if realize the following goals:

$$A \equiv A \xleftarrow{K_{AB}} B \quad \text{(Goal1)}$$

$$B \equiv A \xleftarrow{K_{AB}} B \quad \text{(Goal2)}$$

$$A \equiv B \equiv A \xleftarrow{K_{AB}} B \quad \text{(Goal3)}$$

$$B \equiv A \equiv A \xleftarrow{K_{AB}} B \quad \text{(Goal4)}$$

Finally, the rules stated above will be applied to these logical steps as follow.

From Step 3, using the message meaning rule and Assumption 1, we can deduce:

$$A \equiv AS \left| \sim \left(N_A, N_B, A \xleftarrow{K_{AB}} B \right) \right. \quad (7)$$

Applying the nonce verification rule using Eq. (7) and Assumption (3), one can deduce the following:

$$A \equiv AS \equiv \left(A \xleftarrow{K_{AB}} B \right) \quad (8)$$

Applying the jurisdiction rule using above Eq. (8) and Assumption (5), one can conclude:

$$A \equiv A \equiv A \xleftarrow{K_{AB}} B \quad (9)$$

Similarly for the message sent in Step 4, one can conclude that:

$$B \equiv A \equiv \left(A \xleftarrow{K_{AB}} B \right) \quad (10)$$

From Step 5, using the message meaning rule and Eq. (9):

$$A \equiv B \left| \sim \left(N_A, A \xrightarrow{K_{AB}} B \right) \right. \quad (11)$$

But *A* beliefs in the freshness of N_A (Assumption 3). Using the freshness rule, one can obtain:

$$A \equiv B \equiv A \xleftarrow{K_{AB}} B \quad (12)$$

Similarly, for Step 6, one can deduce the following:

$$B \equiv A \equiv A \xleftarrow{K_{AB}} B \quad (13)$$

We conclude finally that the unicast proposed protocol achieves the goals of authentication and key distribution without bugs as concluded from Eqs. (9-13).

Results and Discussion

This section presents the discussion of the two proposed protocols through the following two sections. First section is the security proof including the unicast key distribution and broadcast key distribution protocols main security advantages. The second section is the performance of our two proposed protocols. However, the performance comparative study includes the comparative study tables and charts for computation and communication overhead between our symmetric unicast key distribution protocol and the other similar symmetric schemes. Attacks comparative table also included in the same section for comparison.

The discussion of each comparative table is being presented with the summarized conclusion as a result in the performance matrices comparative study.

Security Proof

This section discusses security proof of the proposed unicast and broadcasts key distribution protocols. This is achieved through the satisfaction of some security properties such as those given below.

Unicast Key Distribution

- Using any type of nonce either random or serial number. (sequential) with its main properties which are predictability and uniqueness provide freshness property and enhance the resistance to replay attacks
- Storing the unique secret symmetric key between nodes and *AS* in the manufacturing initialization phase as a secret authenticated key enhances the system security with a smaller number of messages
- Enhance system security by applying periodic updates of the stored secret symmetric keys of nodes and Gateways by (*AS*)
- Enhance the difficulties of an attacker to calculate the current stored symmetric keys by making them accumulative based on their previous one by applying the next formula:

$$K_{(AS)}^{i+1} = H\left(K_{(AS)}^{i}\right)$$
$$K_{(BS)}^{i+1} = H\left(K_{(BS)}^{i}\right)$$

- A new generation of the session key by (*AS*) in each new session
- In case of a failure of authentication, the (*AS*) terminates and stops the protocol

Broadcast Key Distribution

- Enhance the security with scalability by applying Logic Key Hierarchy (LKH) among nodes with their relative GW
- The proposal provides forward/backward secrecy

- The storing nodes to symmetric session keys with their relatives GWs
- The storing of GWs to symmetric session keys with AS
- The proposed protocol considers robust addition and revocation as well as fast rekeying

Performance Matrices Comparative Study

In the present paper, we aim to reduce the complexity of computation. Therefore, we use symmetric encryption as a base for the proposed protocol. In addition, we use nonces to withstand both replay and desynchronization attacks. Our proposed scheme which is based on (Pandiya *et al*., 2020), lowers the execution time by reducing the number of encryption/decryption operations, the number of exchanged messages, and the message length. To avoid sending the secret key on a public channel, we propose to store the initial key during the manufacturing phase. To avoid the brute-force attack, the *AS*, frequently, generates a batch rekeying process to update all stored symmetric keys. To calculate the new keys, a hash function is used using the previous key.

In this section, the performance of our proposed protocol is compared to other symmetric protocols concerning the following metrics (Srinivas *et al*., 2020): The computation cost with its parameters and the ability to overcome some well-known attacks.

Computation and Communication Overhead Analysis

The computation cost is split into two main parameters with their relative matrices: Execution cost and communication costs. Execution cost represents the time required to complete the protocol. The communication cost depends on the number of communications and the total message length in bits. Table 1 contains the execution timing of different operations in (ms) (Sani *et al*., 2020) using Tiny OS and conducted on a MacBook Pro Machine (Intel Core i5-6500 CPU@ 3.20GHz with 16GB RAM). These values are used to compare the execution time of the proposed protocol and the other protocols as shown in Table 2.

From Table 2, we can deduce that our proposed protocol has the minimum execution time of the other protocols. Table 4 shows the comparative communication cost (bits) with its two parameters number of communications and total messages length which is based on values extracted from (Srinivas *et al*., 2020) as shown in Table 3. From Table 4, the following remarks are deduced:

- The proposed protocol and the protocol proposed by Cheng *et al*. (2018) have the minimum number of communications compared to the other protocols. However, (Cheng *et al*., 2018) are based on a high-cost hardware platform called LoRa technology. In addition, it suffers from a higher message length and a higher execution time
- Although the message length of (Jeya, 2019) is lower than our proposed protocol, our protocol has the following advantages over (Jeya, 2019): A lower number of communications and a lower execution time

Figures 4 and 5 show a comparison of the execution time and communication cost of our protocol with the other protocols. The comparison shows that our protocol has the highest performance among all protocols.

To conclude, the proposed protocol has the minimum execution time compared to the other protocols with its minimum number of communications, and several message bits. Keep in consideration that it is also based on the manufacturing initialization phase which is useful in realizing system security with a lower number of communications and messages between system participants.

Table 1: Execution time of different cryptographic operations

Notation	Description	Time cost (ms)
Trn	Random number	0.0070 msec
Tnonce	Nonce	0.0005 sec
Thf	Hash function	0.0092 sec
THM	Keyed Hash MAC	0.0183 sec
Tse	AES symmetric encryption	0.0017 sec
T_{sd}	AES symmetric decryption	0.0016 sec

Table 2: Comparison of execution time cost

Ref.	Computation	Execution cost (sec)
Rabiah *et al*. (2018)	36E + 36D + 3H + 5 Rand +13 Mac	0.3840
Cheng *et al*. (2018)	6E + 6D + 2H+ 4 Rand	0.0380
Jeya (2019)	2E + 2D + 4H (Pandiya *et al*., 2020)	0.0435
Kang *et al*. (2020)	14E + 14D + 6 Rand	0.0460
Proposed	43H + 31X + 7 Ran	0.4260
	8E + 8D + 2 Rand	0.0264

E: Encryption, D: Decryption, H: Hashing, Rand: Random no., X: Xo Ring

Table 3: Number of bits of different operations

Operations	Total messages length (bits)
Random Number (RN)	128
Nonce (N)	128
Identity (ID)	160
Hash function (H)	160
Keyed Hash MAC (HMAC)	160
AES symmetric Encryption (E)	128
AES symmetric Decryption (D)	128

Table 4: Comparison of communication cost

Ref.	No. of comms	Communication cost	Communication n cost (kbits)
Rabiah *et al*. (2018)	15	11ID + 7E + 5N + 6RN + 12MAC + H	7.30
Cheng *et al*. (2018)	6	6ID + E + 10N + RN	2.50
Jeya (2019)	9	6E + 4H	1.40
Pandiya *et al*. (2020)	9	6ID + 4E + 8N	2.50
Kang *et al*. (2020)	8	25ID + RN + 3H	4.60
Proposed	6	4ID + 9N + 2E	2.05

Table 5: Attacks comparison table

Scheme	A	B	C	D	E	F	G	H
Rabiah *et al*. (2018)	P	√	P	√	√	P	√	√
Cheng et al. (2018)	P	√	√	√	P	X	√	P
Jeya (2019)	P	√	X	√	√	X	P	P
Pandiya *et al*. (2020)	P	√	√	√	√	P	P	√
Kang *et al*. (2020)	P	√	√	√	P	P	√	√
Proposal	P	√	√	√	√	√	√	√

Acronyms: √: Protected against attacks, X: Vulnerable against attacks, P: Partially achieved
A-Node capture attack, **B**-Impersonation attack, **C**-replay attack, **D**-Brute force attack, **E**-Man in the Middle, **F**-De-synchronization, **G**-Masquerade message attacks, **H**-Identity anonymity

Attacks

In this section, we compare some symmetric schemes and our proposal against various types of attacks as shown in Table 5. The table shows that all compared protocols partially resist node capture attacks. All protocols resist both impersonation and brute force attacks. However, only our protocol resists the remaining attacks. This makes it applicable in various Industrial Internet of Things (IIoT) fields with its low communication and computation overheads, scalability, and security.

Conclusion and Future Work

In the study, the problem of key distribution protocol in IoT networks suitable for smart grid applications is discussed. IoT devices used in smart grids generally have the following constraints: Low computation capacity, low power, and low memory. In literature, different protocols are proposed based on the public key, symmetric key, and hybrid encryption techniques. We propose a key distribution protocol for both unicast and multicast communication. Our protocol fits in various fields such as smart meters in smart grid systems. The proposed solutions (unicast and multicast key distribution) are based on symmetric encryption as it has the following advantages over public and hybrid encryption: Lower key length, lower computation time, lower power consumption, and needs lower memory. To provide authenticity and to resist replay attacks, our protocol is based on primitive cryptography tools such as hashing and nonce. Random nonce generation is the main property for security freshness avoiding the main drawback in timestamps which is clock synchronization. For multicast key distribution protocol, we use the LKH tree to reduce the communication and computation overheads in case of a member change. The comparative study shows that our protocol has the lowest communication overhead compared to the other protocols. In addition, it has almost the minimum computation overhead which leads to minimizing power consumption. In addition, our protocol withstands the most well-known attacks. Thus, the proposed solution meets the different needs of the power utility. The security analysis of the proposed protocol is done using BAN logic. The analysis shows that it achieves the goals of authentication and key distribution. In the present paper, we didn't investigate the node attack which enables the attacker to break in into the system's components (SM, GW, and AS). For future work, we propose to add another security level such as Firewalls and/or intrusion detection systems to stop such attack.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which

have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work, and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

The authors have not received any financial support or funding to report.

Author's Contributions

Samah: First writing of manuscript, put main idea's of manuscript, apply any requested modifications.

Heba Aslan: Designed the research plan, reviewing, participated in all results, final revision to submit.

Ayman Hassan: Gives final approval of the version to be submitted.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all other authors have read and approved the manuscript and no ethical issues have been involved.

References

Abualghanam, O. R. I. E. B., Qatawneh, M. O. H. A. M. M. A. D., & Almobaideen, W. E. S. A. M. (2019). A survey of key distribution in the context of internet of things. *Journal of Theoretical and Applied Information Technology*, *97*(22), 3217-3241.

Adusumilli, P., Zou, X., & Ramamurthy, B. (2005, June). DGKD: Distributed group key distribution with authentication capability. *In Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop* (pp. 286-293). IEEE. https://doi.org/10.1109/IAW.2005.1495965

Ali, I., Sabir, S., & Ullah, Z. (2019). Internet of things security, device authentication, and access control: A review. *arXiv preprint arXiv*:1901.07309. https://doi.org/10.48550/arXiv.1901.07309

Al-Waisi, Z., & Agyeman, M. O. (2018, September). On the challenges and opportunities of smart meters in smart homes and smart grids. In Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control (pp. 1-6). https://doi.org/10.1145/3284557.3284561

Aslan, H. K. (2004). Logical analysis of AUTHMAC_DH: A new protocol for authentication and key distribution. Computers & Security, 23(4), 290-299. https://doi.org/10.1016/j.cose.2003.11.002

Aslan, H. K. (2005). Two-level controllers hierarchy for a scalable and distributed multicast security protocol. *Computers & Security*, 24(5), 399-408. https://doi.org/10.1016/j.cose.2005.01.003

Benmalek, M., Challal, Y., Derhab, A., & Bouabdallah, A. (2018). VerSAMI: Versatile and Scalable key management for Smart Grid AMI systems. *Computer Networks*, 132, 161-179. https://doi.org/10.1016/j.comnet.2018.01.010

Cheng, Y., Saputra, H., Goh, L. M., & Wu, Y. (2018, January). Secure smart metering based on LoRa technology. *In 2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis* (ISBA) (pp. 1-8). IEEE. https://doi.org/10.1109/ISBA.2018.8311466

Choudhary, K., Gaba, G. S., Butun, I., & Kumar, P. (2020). Make-it-a lightweight mutual authentication and key exchange protocol for industrial internet of things. *Sensors*, 20(18), 5166. https://doi.org/10.3390/ s20185166

Daemen, J., & Rijmen, V. (2002). The design of Rijndael (Vol. 2). New York: *Springer-verlag*. https://doi.org/10.1007/978-3-662-60769-5_3

Dammak, M., Senouci, S. M., Messous, M. A., Elhdhili, M. H., & Gransart, C. (2020). Decentralized lightweight group key management for dynamic access control in IoT environments. *IEEE Transactions on Network and Service Management*, 17(3), 1742-1757. https://doi.org/10.1109/TNSM.2020.3002957

Eldefrawy, M. H., Pereira, N., & Gidlund, M. (2018). Key distribution protocol for industrial Internet of Things without implicit certificates. *IEEE Internet of Things Journal*, 6(1), 906-917. https://doi.org/ 10.1109/JIOT.2018.2865212

Farooq, S. M., Hussain, S. S., Ustun, T. S., & Iqbal, A. (2020). Using ID-based authentication and key agreement mechanism for securing communication in advanced metering infrastructure. *IEEE Access*, 8, 210503-210512. https:// doi.org/10.1109/ ACCESS.2020.3038813

Gong, D., Chen, R., Ding, K., Xi, W., Yao, H., Yu, Y., ... & Zhang, Y. (2019, November). Cyber Security Protection of Distribution Automation System Based on Hybrid Encryption Algorithms. In 2019 *IEEE 3rd Conference on Energy Internet and Energy System Integration* (EI2) (pp. 916-920). IEEE. https://doi.org/ 10.1109/EI247390.2019.9062182

Guo, H., Zheng, Y., Li, X., Li, Z., & Xia, C. (2018). Self-healing group key distribution protocol in wireless sensor networks for secure IoT communications. *Future Generation Computer Systems*, 89, 713-721. https://doi.org/10.1016/j.future.2018.07.009

Hu, Y., Xiong, Y., Huang, W., & Bao, X. (2018, August). Key Chain: Blockchain-based key distribution. In 2018 4th *International Conference on Big Data Computing and Communications (BIGCOM)* (pp. 126-131). IEEE. https://doi.org/ 10.1109/BIGCOM.2018.00027

Jeya. J., (2019). A secure hierarchy tree algorithm for efficient and secure key distribution in cloud computing. *Fifth International Conference on Science Technology Engineering and Mathematics (Iconstem)*, India, pp. 33-36. http://doi.org/10.1109/ICONSTEM.2019.8918859

Kang, B., Han, Y., Qian, K., & Du, J. (2020). Analysis and improvement on an authentication protocol for IoT-enabled devices in distributed cloud computing environment. *Mathematical Problems in Engineering*, 2020. https://doi.org/ 10.1155/2020/1970798

Kumar, P., Gurtov, A., Sain, M., Martin, A., & Ha, P. H. (2018). Lightweight authentication and key agreement for smart metering in smart energy networks. IEEE *Transactions on Smart Grid*, 10(4), 4349-4359. https://doi.org/10.1109/TSG.2018.2857558

Kumar, V., Kumar, R., & Pandey, S. K. (2020). A computationally efficient centralized group key distribution protocol for secure multicast communications based upon RSA public key cryptosystem. *Journal of King Saud University-Computer and Information Sciences*, 32(9), 1081-1094. https://doi.org/10.1016/j.jksuci.2017.12.014

Lee, D. H., & Lee, I. Y. (2018). Dynamic group authentication and key exchange scheme based on threshold secret sharing for IoT smart metering environments. *Sensors*, 18(10), 3534. https://doi.org/10.3390/s18103534

Mehibel, N., & Hamadouche, M. H. (2021). Authenticated secret session key using elliptic curve digital signature algorithm. *Security and Privacy*, 4(2), e148. https://doi.org/10.1002/spy2.148

Mittra, S. (1997). Iolus: A framework for scalable secure multicasting. *ACM SIGCOMM Computer Communication Review*, 27(4), 277-288. https://doi.org/10.1145/263109.263179

Moghadam, M. F., Nikooghadam, M., Mohajerzadeh, A. H., & Movali, B. (2020). A lightweight key management protocol for secure communication in smart grids. *Electric Power Systems Research*, 178, 106024. https://doi.org/10.1016/j.epsr.2019.106024

Mohamed, S., Hassan, A. M., & Aslan, H. K. (2021). IoT Modes of Operations with Different Security Key Management Techniques: A Survey. *Journal Homepage*11(6), 641-651. https://doi.org/10.18280/ijsse.110604

Mohammadali, A., Haghighi, M. S., Tadayon, M. H., & Mohammadi-Nodooshan, A. (2016). A novel identity-based key establishment method for advanced metering infrastructure in smart grid. *IEEE Transactions on Smart Grid*, 9(4), 2834-2842. https://doi.org/10.1109/TSG.2016.2620939

Morshed Aski, A., Haj Seyyed Javadi, H., & Shirdel, G. H. (2020). A full connectable and high scalable key pre-distribution scheme based on combinatorial designs for resource-constrained devices in IoT network. *Wireless Personal Communications*, 114(3), 2079-2103. https://doi.org/10.1007/s11277-020-07466-0

Naseer, H., Bhutta, M. N. M., & Alojail, M. A. (2020, October). A Key Transport Protocol for Advance Metering Infrastructure (AMI) Based on Public Key Cryptography. In 2020 *International Conference on Cyber Warfare and Security (ICCWS*) (pp. 1-5). IEEE. https://doi.org/10.1109/ICCWS48432.2020.9292385

Pandiya, C., Singh, S., & Chandavarkar, B. R. (2020, December). Mitigating Masquerade using Nonce in Symmetric Key Distribution-Survey. In 2020 *International Conference on Interdisciplinary Cyber Physical Systems (ICPS)* (pp. 44-50). IEEE. https://doi.org/10.1109/ ICPS51508.2020.00014

Rabiah, A. B., Ramakrishnan, K. K., Liri, E., & Kar, K. (2018, February). A lightweight authentication and key exchange protocol for IoT. *In Workshop on Decentralized IoT Security and Standards* (Vol. 2018, pp. 1-6). sn. https://doi.org/10.14722/DISS.2018.23004

Rafaeli, S., & Hutchison, D. (2003). A survey of key management for secure group communication. *ACM Computing Surveys* (CSUR), 35(3), 309-329. https://doi.org/10.1145/937503.937506

Sani, A. S., Yuan, D., Bao, W., & Dong, Z. Y. (2020). A universally composable key exchange protocol for advanced metering infrastructure in the energy Internet. *IEEE Transactions on Industrial Informatics*, 17(1), 534-546. https://doi.org/10.1109/TII.2020.2971707

Srinivas, J., Das, A. K., Li, X., Khan, M. K., & Jo, M. (2020). Designing anonymous signature-based authenticated key exchange scheme for Internet of Things-enabled smart grid systems. IEEE *Transactions on Industrial Informatics*, 17(7), 4425-4436. https://doi.org/10.1109/ TII.2020.3011849