energies

MDPI

Article

LIKA: Lightweight Identity Based Key Agreement Protocol for Secure Data Transmission in Advanced Metering Infrastructure of Smart Grid

Md Mehedi Hasan, Noor Afiza Mohd Ariffin * and Nor Fazlida Mohd Sani

Faculty of Computer Science and Information Technology, University Putra Malaysia, Seri Kembangan 43400, Malaysia

* Correspondence: noorafiza@upm.edu.my; Tel.: +60-125-415-738

Abstract: Integration of information communication technology via the Internet of Things devices and sensors can enable an efficient power service for utility providers to consumers in advanced metering infrastructure. Authentication and cryptographic mechanisms protect identity, data security and privacy from unauthorised interception for smart meters to servers. In the last couple of years, many key agreement protocols have been prescribed and deployed to fix those issues. Unfortunately, the deployed protocols did not work inside the same protocols, specifically parameter detection and recognition for session key generation, as they entail high computation time and communication bits overheads. Furthermore, the absence of forward secrecy and user anonymity affects the authentication. Therefore, we have proposed a lightweight identity-based key agreement (LIKA) utilising the Diffie–Hellman cryptography with a trusted authority. It seeks to cover both the security and performance criteria with equal weight. The protocol is evaluated by the Canetti and Krawczyk adversarial model, Avispa and cryptographic analysis released the session keys that were not considered as an adversary during mutual authentication. Moreover, as compared to related work, the proposed protocol took the least amount of time (5.319 ms and 1056 bits) for the entire process of session key generation. Furthermore, comparative analysis has shown that the LIKA adequately encompasses computation, communication, and security assessments. Consequently, it is more convenient for practical implementation for a smart grid.

Keywords: mutual authentication; trusted authority; cyber-security; Diffie–Hellman; advanced metering infrastructure; smart grid

check for updates

Citation: Hasan, M.M.; Mohd Ariffin, N.A.; Mohd Sani, N.F. LIKA: Lightweight Identity Based Key Agreement Protocol for Secure Data Transmission in Advanced Metering Infrastructure of Smart Grid. *Energies* **2022**, *15*, 8106. https://doi.org/10.3390/en15218106

Academic Editor: Alberto-Jesus Perea-Moreno

Received: 30 April 2022
Accepted: 13 June 2022
Published: 31 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A smart grid assembles an electrical grid that is superefficient, trustworthy, and secure for supplying information to consumers and utility providers, or vice versa. Researchers and industrialists are interested in smart grids because of their diverse roles. Integration of information communication technology plays a remarkable exordium in a smart grid by turning a traditional grid into a smart modern platform [1]. Instead, it has a next-generation power system [2]. Within a smart grid, one of the vital components is the advanced metering infrastructure (AMI), which is a kind of middle-wire communication platform between the consumer and the service provider [3]. There are a lot of device sensors applied to the generation centres among consumer premises to transfer services and monitor in terms of scalable grid environments [4]. The architecture of the AMI is planned by the service provider, customer premises and cognitive device, server and smart meter with a communication system [5,6]. Additionally, some resource-constrained devices perform end-user to server-end functions, such as smart meters [7].

Meanwhile, heterogeneous devices and sensors exist in advanced metering infrastructure, where authentication is of utmost important for checking and verifying between both the active number of entity service providers, smart meter, customer, computers,

gateways, servers etc [8]. The primary goals of authentication are identifying the accurate entity in the system, robustness and versatility for data security-preservation and protection [9]. Additionally, there is a key security concern in the advanced metering infrastructure that helps protect user anonymity, key exchange security, forward personal privacy, confidentiality, credibility, non-repudiation, and secure user authentication in order to provide reliable data transmission between two parties in smart grid environments [10]. Besides, it is crucial that data transmission should remain encrypted end to end between two parties [11]. An encryption mechanism is a quick tool to defend against cyber attacks [12]. Public cryptography-based approaches, namely encryption and decryption policy, are important to keep smart grid requirements for better message transmission without unauthorized access [13].

The performance and security aspects of the situation were the starting point for this investigation. This work endeavours to examine and analyse previous work to address its strengths and weaknesses. It has made significant progress in overcoming the aforementioned limitations [14]. Experts have already delivered ways to make advanced metering infrastructure more lightweight and resistant to unwanted access, although there are some drawbacks depicted, such as key constraints resulting in inadequate authentication between two parties causing security flaws triggered by the delay of time and bit operations, which give an opportunity to reveal the parameter identification during the key session or key generation in terms of either failure anonymity or forward secrecy. Moreover, hampered efficiency due to high latency of cryptographic operations has resulted in high computation time and communication bits, which were overwhelming for limited processing devices in the AMI.

The latest solution of encryption and authentication mechanisms, such as key agreement protocols, have been shown to fill security holes by using cryptographic operations, public keys, and authentication techniques [10,11]. This study focuses on adapting encryption and authentication to overcome the solution in terms of both security and performance considerations in the AMI. Therefore, a novel lightweight identity-based key agreement protocol adopting the Elliptic curve Diffie–Hellman cryptography and a trusted authority is proposed. The following contributions are made by this research.

- The proposed protocol takes less time and bit operation compared to existing protocols. The LIKA decreased the additional cryptographic operation burden for AMI context devices.
- According to its construction, the LIKA ensures mutual authentication between two parties, known as a smart meter and aggregator point, in terms of preserving forward secrecy and user anonymity.
- The proposed protocol was evaluated by the widely adopted security assessment tool, Avispa, and CK-adversarial confirmed formal security evaluation. Moreover, informal analysis proves multiple security features handled by LIKA against identified attacks.
- To draw against past records, the obtained results reflect the proposed protocol security and performance efficiency, both criteria equally utilised within the same protocol. These would serve as a fresh route for the AMI to improve the reliability and convenience of a smart grid.

The study is organised as follows: in the second section, we conducted literature research on the background of the proposed protocol. In the third section, we designed a brief methodology covering the current flaws. Afterwards, in the fourth section of the study, we critically examined the planned work in three separate parts, formal and informal, for security evaluation. Furthermore, the efficiency evaluation was related to comparing the existing protocols. Lastly, the fifth section draws a conclusion and future research direction is portrayed, respectively.

2. Literature Review

In recent years, numerous unique authentication and key agreement solutions have emerged to address performance and security concerns in smart grids of AMI environments. The authors of the [1,5,9,15–27] protocols primarily relied on the three-phase mechanism, which are initialization, registration, and authentication/key agreement with assistance of trusted authority or self-development of each phase. These protocols used the ECC, Diffie–Hellman, bilinear pairing and identity-based cryptography to build key agreements between smart meters to utility aggregator points. Table 1 presents the work related to this study to understand their objectives, methods, approaches, results, advantages and limitations. A broad analysis about the existing work is shown below.

According to Table 1 summarizing existing work, we discovered that most of the researchers have recently presented solutions managed by cryptographic mechanisms. Unfortunately, the time and bit operation costs associated with the security features evaluated were not equitably acquired earlier. Notably, past methods either concentrated on the authentication element or decreasing the time and bit issues, while paying little attention to the performance and security features of the same protocols in both instances. Most typical faults suffer from high computation time and communication bit overheads due to the excessive latency of cryptographic operations that are disrupting performances. In addition, security vulnerabilities caused by delays in parameter checking and identification operations, created opportunities for unauthorised users to intercept the session are also part of the faults. In terms of failures of forward secrecy and user anonymity, there was inadequate authentication between the smart meters and aggregator points.

Table 1. Related work.

Schema	Research Objectives	Compromise Method	Results: Performance and Security Assessments	Advantage	Limitations
[16]	Propose an efficient authentication with key agreements protocol for privacy and security issues of smart grid entity.	-Encryption: ECC with Diffie Hellman key exchange. -Authentication: Basis on ECDSA.	-Computation time: 30.80 ms. -Communication bit: 3744. bits -Authentication: Mutual authentication SM to AG. -Security proof: No security proof presented.	-Efficiently achieved Mutual authentication. -Comparative analysis presented for security and performance aspect.	-Session key generation requires more extensive time than other protocols. -Due to high latency and delay during key generation, the unauthorized party makes scope for taking over the session key.
[17]	Proposed an authentication and key agreement protocol utilizing an identity-based mechanism with public key infrastructure to improve efficiency and security.	-Encryption: Public key infrastructure. -Authentication: Based on self-signature.	-Computation time: 27.78 ms. -Communication bit: 1152 bits. -Authentication: authentication between SM and AG. -Security proof: security proof by Avispa.	-Total execution time covered the whole process of smart meter to the utility server. -Effectively minimize communication bits.	-Various cryptographic operations are required for each transmission. -Mathematical complexity of self-signature pinch for intruders.
[18]	Propose a novel key management protocol to resolve the man in the middle and replay attack in AMI infrastructure.	-Encryption: ECC with symmetric encryption. -Authentication: Needham-Schroeder authentication Protocol.	-Computation time: 7.56 ms. -Communication bit: 100 bits each handshaking. -Authentication: Only the sender (smart meter) authenticates. -Security proof: No security proof presented.	-Strongly managed MITM and replay attacks. -System model suitable for smart grid requirement.	-Need an extra server for authentication purposes thus increasing the processing of computation time cost. -Disclose the session key and vulnerable for DoS attack.
[19]	Proposed lightweight key agreement protocol in order to assure SM privacy, security, efficiency.	-Encryption: RSA with modular exponential. -Authentication: Not presented properly.	-Computation time: 18.31 ms. -Communication bit: 2457 bits. -Authentication: Did not mention. -Security proof: No security proof presented.	-RSA based approach covered tight security assessment. -The system architecture is a new path in the further research.	-RSA encryption takes more significant time for session key generation. -Vulnerable for session key due to no such parameter covered the session key security.
[20]	Proposed secure communication between the smart meter and service provider by using an identity base self-signatures mechanism.	-Encryption: ECC based cryptography -Authentication: Hash function	-Computation time:10.53 ms. -Communication bit: Required 4 steps per handshaking, took 100 bits. Authentication: Mutual authentication. -Security proof: Random oracle security proof presented.	-Competently discovered mutual authentication. -This protocol kept smart meter anonymity in a public channel.	-Utilized two different cryptosystems for authentication, raised time and bits operation. -Due to identity disclose utility server, faced guessing attack. -provide fewer security features.

Table 1. *Cont.*

Schema	Research Objectives	Compromise Method	Results: Performance and Security Assessments	Advantage	Limitations
[21]	Proposed identity based probably key agreement protocol to tackle session key security by trusted authority based approached.	-Encryption: ECC with ECQV properties. -Authentication: Hash function.	-Computation time: 23.81 ms. -Communication bit: 832 bits. -Authentication: Acquire trusted authority-based authentication. -Security proof: Random oracle, Avispa (formal), parameter-based (informal).	-Resist MITM and replay attack. -Demonstrated three-way security proof.	-Does not support forward secrecy and did not show DoS attack mitigate evaluation. -In every phase, utilized cryptographic operation for parameter detection increased computation time.
[1]	Presented ECC based anonymous key agreement for secure transmission smart meter to a service provider.	-Encryption: ECC based ECDSA cryptographic properties. -Authentication: Hash function.	-Computation time: 13.34 ms. -Communication bit: 1440 bits. -Authentication: Trusted authority handle the mutual authentication. -Security proof: Random oracle model, Proverif.	-Preserve mutual authentication between a smart meter to aggregator point. -Emerged self-signature for parameter identification.	-Following protocol owned several time hash operation, which is generated and retrieved raised time and bits cost.
[14]	Propose lightweight and key agreements with ECC based cryptography for cyber security in smart grid.	Encryption: ECC based cryptography Authentication: Hash function.	-Computation time: 28.83 ms. -Communication bit 1940 bits. -Authentication: authentication between smart meters to service provider server. -Security proof: Under random oracle model.	-Significantly reduced computation time in smart meter side.	-Take long time for established session key generation scope for vulnerable in attack scenarios.
[22]	The proposed paper deals with protection and privacy issues are the key subject of the smart gird.	-Encryption: Symmetric key with biometric approaches. -Authentication: HMAC.	-Computation time: 8.92 ms. -Communication bit: 1152 bits. -Authentication: Mutual authentication. -Security proof: only measures security features as informal assessment.	-Utilized hybrid encryption scalable the metering infrastructure beside it take low computation power.	-The symmetric key encryption vulnerable for many attack scenarios like: eavesdropping, MITM. -There is less attention to protect key security.
[9]	Propose Fully ECC Hashed Menezes-Qu-Vanstone key exchange for reduced computation overheads problem.	Encryption: ECC with FHMQV scheme. Authentication: Hash function.	-Computation time: 12.00 ms. -Communication bit: 322 bits (key generates by TA). -Authentication: present a mutual authentication between smart meter to NAN. -Security proof: Avispa and informal cryptographic analysis.	-Significantly reduced communication bits in term of less delay than above protocol. -Resists number of attacks scenario.	-Did not concentrate on massage integrity. -CA certificate revocation and generation increase the computation time instead. of unauthorized scope for intercepting the message.

Table 1. *Cont.*

Schema	Research Objectives	Compromise Method	Results: Performance and Security Assessments	Advantage	Limitations
[5]	Propose multi-factor authentication scheme for smart meter physical and data privacy.	Encryption: Fuzzy extractor-based cryptography. -Authentication: One way Hash function.	-Computation time:14.13 ms Communication bit: 908 bits. -Authentication: Acquire smart meter authentications. -Security proof: Utilized random oracle proof for security measures.	+Preserve the smart meter physical security.	-Increased computation time due to high mathematical complexity. -Do not reached massage integrity.
[15]	Propose password based key agreements protocol to ensure reduce computation time and communication bits.	-Encryption: ECC with symmetric block cipher cryptography. -Authentication: HMAC.	-Computation time: 17.80 ms. -Communication bit: 1184 bits. -Authentication: Mutual authentication. -Security proof: Avispa and informal cryptographic analysis.	-Mitigates multiples number of attacks.	-[23] argue that this protocol have problem in registration and authentication due to incorrect ECC point multiplication, which is not applicable for presence of multiple devices.
[24]	Proposed identity based key agreement protocols by using elliptic curve with bilinear pairing cryptography.	-Encryption: Public key infrastructure -Authentication: HMAC	-Computation time: 15.76 ms. -Communication bit: 621 bits. -Authentication: Mutual authentication. -Security proof: There is no security proof presented.	-The protocol eliminates highest number of time and bits operation.	-Though efficiently managing performance issues but unfortunately do not present any security model.
[25]	This study provides a novel secure authenticated key agreement mechanism for smart grids that considers the possibility of a hostile third party.	-Encryption: Bilinear pairing cryptography. -Authentication: One way hash.	-Computation time: 20.98 ms. -Communication bit: 1120 bits. -Authentication: Not present.-Security proof: Avispa and BAN logic (formal analysis).	-This protocol managed computation, communication overheads and security criteria at same time within their protocol.	-Every phase used cryptographic operation raised computation latency. -Proposed protocol offers limited security features. -Lack of forward secrecy due to inattention of session key security.
[26]	Proposed a certificate less key agreement protocol for solved a key escrow issues.	-Encryption: Bilinear pairing cryptography. -Authentication: Not available.	-Computation time: 13.04 ms. -Communication bit: 160 × 8 bits. -Authentication: Not presented. -Security proof: Security proof by eCK.	-The security model and proof offered by this paper will lead to a new paradigm for dealing with the following security assessments.	-Each phase were accommodate bilinear operation increased gradually computation time operation.
[27]	This protocol proposed ID-2PAKA protocol to ensure additional burden of cryptographic operation.	-Encryption: ECC with bilinear pairing cryptography. -Authentication: Trusted authority covers the two parties authentication.	-Computation time: 13.816 ms. -Communication bit: 1274 bits. -Authentication: Mutual authentication. -Security proof: security proof under random oracle model and informal analysis.	-Comparative analysis illustrated the protocol equally maintain performance and security aspect.	-The author stated that bilinear paining increased the computation cost that would be avoided in metering infrastructure especially for the fewer computation devices in AMI.

3. Methodology

This methodology aims to provide the best degree of protection by improving proper authentication during the session key generation between smart meters and aggregator points by using the elliptic curve Diffie–Hellman (ECDH) cryptography mechanism with assistance of trusted authority-based access token approaches. Meanwhile, the proposed protocol intends to preserve security and performance within the protocol. Our chosen methods aims to reduce time and bits overheads compared to other schemes. Additionally, formal, and informal assessment shows the proposed protocol conserves mutual authentication in terms of maintaining forward secrecy and anonymity, respectively. Moreover, the proposed protocol offers numerous security features and multiple performance features compared to existing work.

3.1. System Component

This methodology is applicable in the smart grid context, which is a core field for advanced metering infrastructure. Figure 1 demonstrates the proposed system model. In this work, we look at three entities: one trusted authority connected to a secure channel with a utility provider. The second component of this methodology is called the data aggregator point, while the smart meter is the third component.

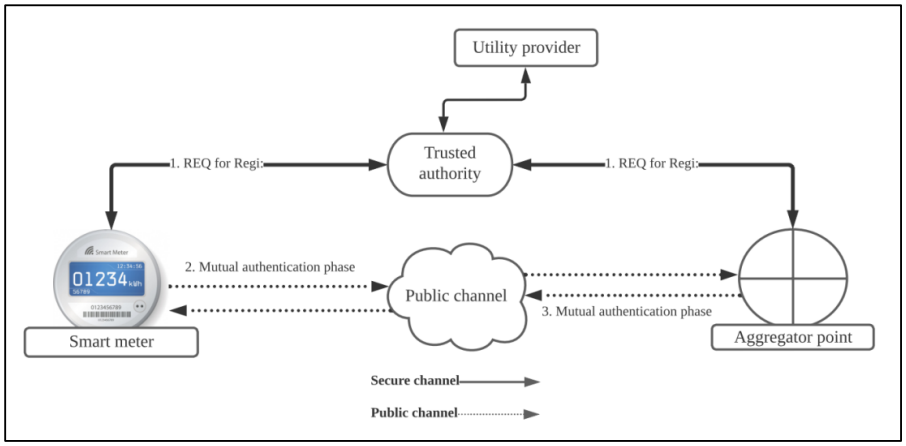


Figure 1. Proposed protocol system model.

Our proposed protocol is responsible for monitoring electric flows and transmitting real-time electric data to users as a smart meter and aggregator point, such that it acts as a utility side entity situated in a neighbourhood area network for executing multiple work scenarios. Besides, the smart meter is a computationally constrained electrical device that controls minimum processing power and is capable of collecting data and controlling smart appliances, which are installed in a home area network.

3.2. Initialization Phase

This is the initial phase based on $y^2 = x^3 + 2x + 2$ (mod), which is executed first by selecting G, which subsequently generates a public/secret key with the help of an elliptic curve {P, a, b, G, n, h}; here p is large prime numbers in a specific elliptic curve. The first coefficient of the curve is a, whereas b is the second. Parameter G is the generator point (base point) where the parameter G denotes <a, b>, while n is the prime order of G and h is the cofactor of the group. The following parameter computes a private and public key pair and stores it in a directory for later use in the authentication phase. It is used to perform scalar multiplication via the ECDH properties in the basic two-group function, which are point addition and point doubling, with the aid of a generator point G. Scalar multiplication should be used to generate shared session keys [28]. The next phase follows these parameters effectively by utilising them.

3.3. Registration Process with Trusted Authority

In this phase, the smart meter and aggregator point asks a trusted authority for permission to join the mutual authentication process. The smart meter and aggregator point send credentials to a trusted authority which are verified, based on pre-loaded databases and generates TokenSM and TokenAG. The registration process is shown below.

Process SM to TA: Smart meter selects ID_{SM} and corresponding password hash to request for registration. Here, ID_{SM} and PW_{SM} are considered as identification and the security parameter, therefore, SM send as: $(ID_{SM}||PW_{SM})$ to the TA in a secure channel.

Process TA to SM: After receiving the credentials, TA checked ID_{SM} and PW_{SM} into the database. If matched with the TA database, then it will generate the service token for the smart meter $Token_{SM;}$ if it is not true, the request will be aborted.

Process SM: Upon receiving a message from TA, it will store the $Token_{SM}$ into the directory. The $Token_{SM}$ consists of the smart meter ID and MAC address, token generation and expiry time, and destination identification, mean aggregator point ID are all included in the token header. Furthermore, the following token serves as an access control for the SM between AG to initiate the authentication phase. At this stage, it is ready for mutual authentication with the aggregator point. According to the above, the same process is done by the aggregator point (AG) with a trusted authority (TA) for registration purposes when it communicates with the smart meter. Figure 2 illustrates the entire process of registration and mutual authentication phase between smart meters and aggregator points.

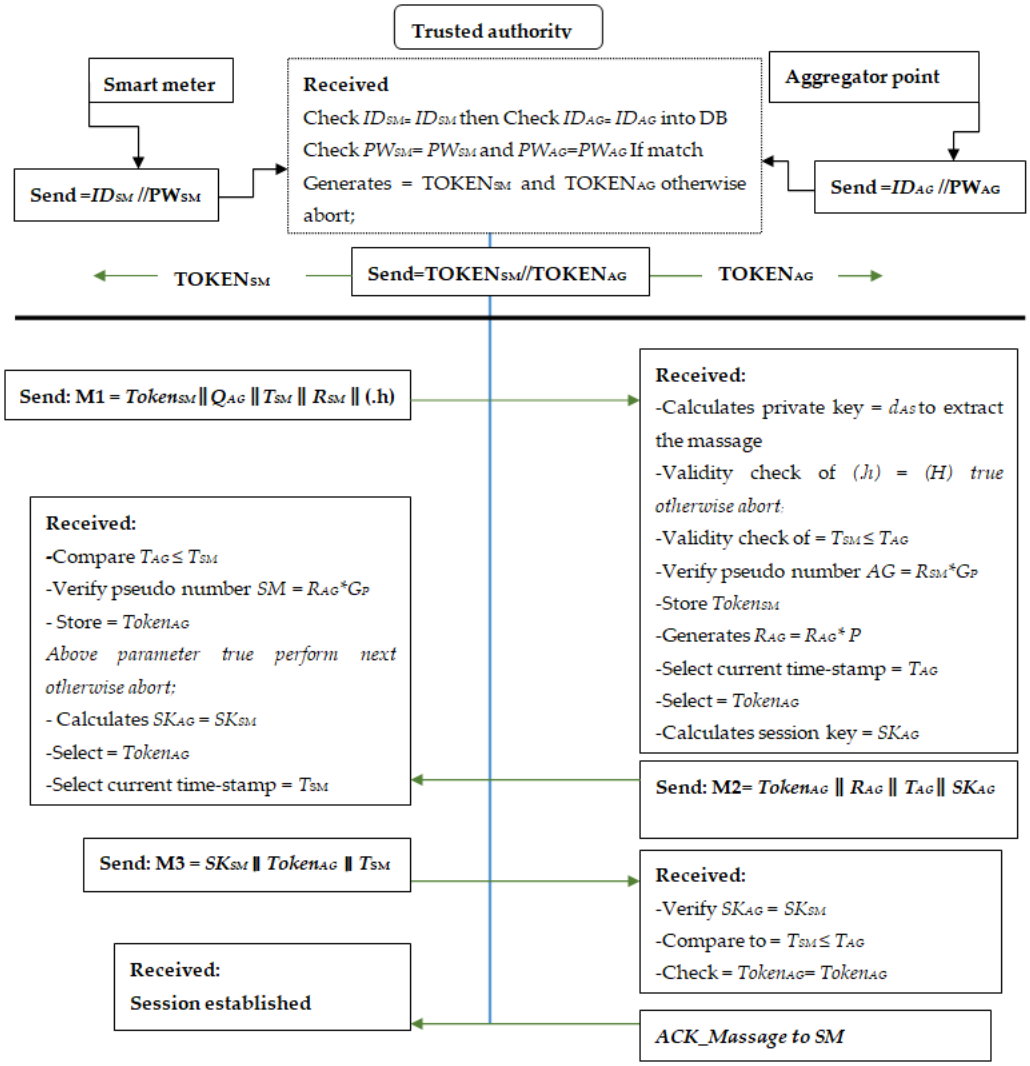


Figure 2. An illustration of Registration and Mutual Authentication Process.

3.4. Mutual Authentication Process Smart Meter with AG

Once the registration process is completed, a mutual authentication is performed between the smart meters belonging to the home area network and aggregator points that exist in the neighbourhood area network. This is known as a utility side regional server; it collects data from the smart meter after a bunch of data is finally sent to the utility main server for the next process. In this section, the following work will focus on smart meters and aggregator points moving to a mutual authentication between both sides together with passing valuable parameters and vice versa for session key generation.

Step 1: The smart meter initiates a mutual authentication and key agreement to AG. According to help of the initialization phase, it selects a prime number to compute private key d_{SM} into P$\in \times$G point and public key Q_{SM} of smart meter to $d_{SM} \times G^P$ point.

Based on the initialisation and registration phase, the SM select a $Token_{SM}$, which is collect from TA during registration phase. Then it computes the public key of the aggregator point in Q_{AG}.G point. After this, it selects the current T_{SM} of the smart meter and generates and computes a pseudo-random number $R_{SM} \times G^P$ using ECC point multiplication. Then a one way hash is generated that relies on function h: {0, 1}*$\rightarrow$ {0, 1} = (.h) and wrapped following parameter as: {h($Token_{SM}\|T_{SM}\|R_{SM}$).Lastly, the encryption M1 is sent as: {Q_{AG} ($Token_{SM}\|T_{SM}\|R_{SM}\|$(.h)} to the aggregator point.

Step 2: Following the credentials received from the SM, it will first decrypt the packet by d_{AG}; after it validates the received hash (.h), if the value has changed it will drop the session. Otherwise, true check the value for the current time-stamped $T_{SM} \leq T_{AG}$ and probe the pseudo random number find in $R_{SM} \times$G point. Afterwards, save the $Token_{SM}$ for further processing. Subsequently for the next round of validation the AG selects its own identity $Token_{AG}$ afterwards, it selects the current time-stamp T_{AG} and calculates the session key by help of the initialisation phase using scalar multiplication operation. The AG computes the shared session key as: {$SK_{AG} = d_{AG} \times Q_{SM} \times G$}. Next, it computes a random number $R_{AG} \times G^P$ for key freshness and wrap the parameter by M2 send as: {R_{AG} ($Token_{AG}\|T_{AG}\|SK_{AG}$)} to the SM.

Step 3: Upon receiving the appended message from AG, the smart meter validates the authenticity of the $R_{AG} \times G^P$, which belongs to SM in G^P point. Then, it compares the current $T_{AG} \leq T_{SM}$; if the appended parameter values match, further the $Token_{AG}$ is stored for next handshaking. Finally, it computes the shared session key of SK_{SM} as: {$SK_{SM} = d_{SM} \times Q_{AG} \times G$}; if the session key SK_{AG} and SK_{SM} are the same, it moves toward to the next section, otherwise the session will fail. Afterwards, SM takes B_M which contains real-time billing information. Then, select T_{SM} along the wrapped parameter by M3 send as: {SK_{SM} ($B_M\|Token_{AG}\|T_{SM}$)}to the AG.

Step 4: Afterwards, AG takes the message; it is extracted by SK_{AG} if equivalent with SK_{SM}, then is computed $T_{SM} \leq T_{AG,}$ then the equality of $Token_{AG}$ and $Token_{AG}$ is checked to see if it is true to ensure the received message source is real. Otherwise, if the values do not match, the session will fail.

Step 5: Finally, the aggregator points send the ACK_SM to the smart meter. The overall steps are shown in Algorithms 1 and 2, which further explains the proposed method.

Algorithm 1. Registration Smart Meter to Trusted Authority

1: SM inputs ID_{SM}
2: SM inputs PW_{SM}
3: SM sends {ID_{SM}, PW_{SM} } towards to TA
4: TA Check the credential by TA database
5: if (ID_{SM} = ID_{SM}, PW_{SM} = PW_{SM} match in TA database) then
6: TA accept the SM request, creates: (Token) for SM
7: TA send {$Token_{SM}$ } toward to SM
8: SM store {$Token_{SM}$ } into own dashboard
10: return (Success)
11: return (Failure)
12: **else**
13: **end if**

Algorithm 2. Mutual Authentication and Data Transmission

1: initiate private key d_{SM} of SM by $P_{SM} \in Zp^*$
2: calculates public key $Q_{SM} = d_{SM} \times G$
3: select inputs SM = $Token_{SM}$ **then**
4: select public key of AG = Q_{AG} on key directory
5: generates time-stamp T_{SM} for verification purpose
6: computes SM = $R_{SM} \times G_P$ for session freshness
7: SM send M1 = ($Token_{SM} \| Q_{AG} \| T_{SM} \| R_{SM} \| (.h)$) appended with hash toward to AG
8: **while** AG extract the M_1 by own private key d_{AG} and **do**
9: check = $T_{SM} \leq T_{AG}$ **then**
10: if received R_{SM} with generated AG computes $R_{SM} \times G_P$ **same**
11: if received hash((.h) = (h)) generated true **then**
12: AG store $Token_{SM}$ into dashboard
13. return (Success)
14: **else**
15: return (declined)
16: **end if**
17: select current AG = T_{AG}
18: AG computes R_{AG} = AG $\times$ G_P
19: calculates AG = SK_{AG} then
20: AG send M2 = ($Token_{AG} \| R_{AG} \| T_{AG} \| SK_{AG} \| (.h)$) toward to SM.
21: **end while**
22: **while** SM compares ($T_{AG} \leq T_{SM}$) belonging G^{*p} **do then**
23: verify pseudo number SM = $R_{AG} \times G_P$ **then**
24: store = $Token_{AG}$
25. return (Success)
26: **else**
28: return (declined)
29: **end if**
30. **end while**
31: **if** calculatesSK_{AG} = SK_{SM} true **then**
32: select = $Token_{AG}$
33. select current time-stamp = T_{SM}
34. send M3 = $SK_{SM} \| Token_{AG} \| T_{SM}$
35: if AG verify (SK_{AG} = SK_{SM}) truethen
36: compare to = $T_{SM} \leq T_{AG}$
37: check = TokenAG= TokenAG
38. return (Success)
39: **else**
40: return (declined)
41: **end if**
42. return (Success)
43: **else**
44: return (declined)
45: **end if**

4. Results and Analysis

This section shows the results of security analyses and performance efficiency analyses based on the authors of [15,27,28]. It is accomplished by the steps: an evaluation of security and its functionality aspects, after comparison of computation and communication overheads are deployed. Additionally, we add comparative demonstrations at the end of the analysis. The security analysis section illustrates an informal interpretation wherein the ECDH parameters are given mathematical complexity known as an Elliptic curve discrete

logarithm problem and an Elliptic curve Diffie–Hellman problem [29], A one-way hash function with an appended pseudo random number, a time-stamped parameter were built as an additional burden for the unknown user in the session key generation in both the smart meter and aggregator points. The CK adversarial model and Avispa were employed in the formal analysis. Furthermore, the formal study examines how forward secrecy and user anonymity deal with well-recognised tools to maintain mutual authentication according to the suggested technique. The performance analysis section contains two analysis criteria, which are commutation time and communication bits, both of which are assessed to compare the proposed protocol against existing schema.

4.1. Assumption

This study is based on the following assumptions.

- The threat model is deduced to be Dolve–Yao, with an adversary having connected to the channel and being able to interrupt the message in the smart meter and aggregator point.
- The smart meter and aggregator point in the registration phase with TA build in secure channel.
- The intruder can resend and disrupt messages to the SM and aggregator point. The intruder also can eavesdrop and inject harmful messages into the relayed communication between the two parties during session key generation.
- The intruder can impersonate the SM and/or aggregator point, the AG is securely connected to the utility server. On the other hand, the channel between the AG and the SM is assumed to be public is insecure as a result, the configured proposed protocol guarantees security between the two parties.

4.2. Mathematical Complexity

According to the elliptic curve cryptography (ECC), one of the major Elliptic curve discrete logarithm problem-based properties includes the ECDH. This provides some mathematical functionality without the knowledge value of ECDH parameter and does not reach into the point, which is given below.

4.2.1. Elliptic Curve Discrete Logarithm Problem (ECDLP)

Let P be the large prime integer number, g is an additive cyclic group G is the generator point or in terms of base point of the curve with prime order G. For the given point a, bX where X∈G and a $\in Z_P$, it is computationally not feasible to find a, [1]. In this proposed scenario, if we have a prime number and generator point of G like $d_{SM} \times G$, then it will be easy to find Q_{SM}, but we have reached G and Q_{SM}, which is not an easy process to get d_{SM} points for the ECDLP problem.

4.2.2. Elliptic Curve Diffie–Hellman Problem (ECDHP)

Let P be the large prime integer number, g is an additive cyclic group and G is the generator point or in term of base point of the curve with prime an order G. For the given point aX and bX where X ∈ Z and a, b ∈ Z* the point is computationally not infeasible to find abX [23,24].

4.2.3. Hash Function

The hash function is a collision-resistant one-way hash function. Hence, finding x is hard in a hash-based function like h(x): Hash function = h is secured one-way = h: $\{0, 1\}^* \rightarrow \{0, 1\}^n$. This object takes the output of x ∈ {0, 1} and h(x) $\{0,1\}^n$ of a certain range of length n.

4.3. Security Analysis (Informal)

This section discusses the informal security analysis that relies on ECDH mathematical complexity, time stamp, pseudo random number, and hash operation. By combining these features, the proposed system is protected from unauthorized access. Additionally, the proposed protocols provide mutual authentication between the smart meter and the aggregator point. The following section describes an informal examination strategy searching the relevance of the security criteria.

4.3.1. Mutual Authentication

The proposed protocol preserved mutual authentication in smart meters between aggregator points. Additionally, the proposed mechanism is supported in an insecure channel as it keeps domain authentication because in a smart grid, millions of smart meters and thousands of aggregators exist, so it should be able to identify the exact source and destination. The proposed mechanism proved domain authentication during the registration phase of smart meter (SM) to a trusted authority (TA) by using SM identity = ID_{SM}, which was previously stored in TA. The same process for AG can confirm the domain authentication in an insecure channel. Another feature confirms that the system mutually transmits information generated by the smart meter sign $TOKEN_{SM}$ and $TOKEN_{AG}$, which were rechecked in the authentication phase; if the SM= $TOKEN_{SM}$ = AG = $TOKEN_{AG}$ is the same, then exchange the message, otherwise the session will be aborted.

After that, we introduced the mutual authentication, which made it more evident that a smart meter initiates the process and calculates the key $P_{SM} \in Z^*_P$, the private key d_{SM}, and the public key are calculated by $Q_{SM} = d_{SM} \times G$ for the secret key generation, the same as on the AG side to generate the session key = SK_{AG}, if the appended values and SK_{AG} = SK_{SM} mutually match, then it is authenticated. Supposing the adversary tries to intercept the value, it will be computationally infeasible to retrieve the hash values without a private key in this scenario, as only the private key knows their own.

4.3.2. Replay Attack

A replay attack is kinds of network attack, whereby attackers send a malicious or dishonest transmission of the valid data over the network. The malicious entities shall replay or postpone the transmitted data that retrieves and sends the information [30]. The proposed protocol in smart meters for aggregator points resists the replay attack following the proposed mechanism that provides every message transferred between them a check of time-stamped values T = T_{SM} = T_{AG}. If the values change, the connection is dropped immediately. Another way to avoid the attack is by refreshing in every step on the SM to AG to generate key freshness in $R_{SM} = R_{SM} \times P$ and $R_{AG} = R_{AG} \times P$ without previous knowledge of the parameter, it did not retrieve. Hence, the proposed system is saved from replay attack.

4.3.3. Man in the Middle Attack

In security and cryptography, a man in the middle intervention is an attack where the perpetrator secretly intercepts, transmits messages or changes communication between two domains [29,30]. Both users think they are communicating with each other individually. The proposed protocol resists the MITM attack for the following reason: somehow A gets the message same as M1 = ($TOKEN_{SM} \| Q_{AG} \| T_{SM} \| R_{SM} \| (.h)$) but A gets the Q_{SM} and Q_{AG} which is only a parameter publicly available in the channel. In terms of the adversary A trying to extract the appended value, due to the ECDLP and ECDHP they do not acquire the private key. Therefore, A cannot open and adjust the other parameter if a change happened, which is a mess for all parameters, so he/she perceives to receive the value that did not come from an exact source; hence, the process is immediately stopped. Thus, the proposed protocol withstands the MITM attacks.

4.3.4. Impersonation Attack

In this attack, the attacker attempts to mark himself as an authorized device or individual to communicate with other entities on the network [31]. The designed system also resists type of impersonation attack in between SM and AG as well during registration with TA. In this phase, A can try to impersonate as a legitimate user to interact with others; it is quite impossible because this system mutually authenticates by ($TOKEN_{SM}||Q_{AG}||T_{SM}||R_{SM}||(.h)$) with ($TOKEN_{SM}||Q_{SM}||T_{AG}||R_A||(.h)$). Here creates the $SK_{AG} = SK_{SM}$ contained all the appended value if match then create the session otherwise not. Without previous knowledge of the parameter, it does not get it, only the SM and AG kept the information, so this protocol saves from impersonation attack.

4.3.5. Support Forward Secrecy

Forward secrecy, also known as perfect forward secrecy, is a precise feature of key agreement protocol. It gives assurance that the session key will not be compromised in short- or long-term considerations in terms of without any parameter information [32]. The proposed protocol supports forward secrecy as the session key $SK_{AG} = SK_{SM}$ is built by ($TOKEN_{SM}||Q_{AG}||T_{SM}||R_{SM}||(.h)$) and ($TOKEN_{AG}||Q_{SM}||T_{AG}||R_{AG}||(.h)$). The following information that are time-stamped $T_{SM} = T_{AG}$ and random number $R_{AG} = R_{SM}$ ensure each message session freshness along both private keys (d_{SM}, d_{AG}) provide confidentiality to both entities. Thus, the adversary will have trouble cracking the combination of constantly shifting variables and highly protected private keys.

4.3.6. Support Anonymity

To prevent a hacker who has recorded previous conversations from learning the identity of those involved, computer security and encryption provide a feature known as forward secrecy [33,34]. The proposed designed protocol considers anonymity issues by following the method we mentioned earlier in the smart meter (SM) and aggregator (AG) point conversation during the authentication and data transmission phases. There is SM with AG passing all messages between them to accommodate with the time-stamped value $T_{SM} = T_{AG}$. Due to the cause of randomness, this will return different values to execute each conversation. Thus, A cannot get the real identity of the user, so this protocol kept the identity anonymous. Moreover, SM and AG collect TOKEN from TA as it relies on anonymity performed in a public channel.

4.3.7. Key Freshness

Each step uses a new key in this proposed scheme, which is a time-stamped relay random number. Thus, for each session, key freshness condition prevails.

4.3.8. Message Authentication

The AG receives the message M1 = ($TOKEN_{SM}||Q_{SM}||T_{AG}||R_{AG}||(.h)$). The first private key d_{SM} opens the message after verifying the time-stamped value $T_{SM} = T_{AG}$ by matching with the hash value (.h) = (H). The SM receives the credential M2= ($TOKEN_{AG}||R_{AG}||T_{AG}||SK_{AG}||(.h)$). Here, the originality of the time-stamped $T_{SM} = T_{AG}$ is checked with the hash value (.h) = (H), with the verifier $SM = R_{SM} \times P$ and $R_{AG} = R_{AG} \times P$. Through this, stable messages can be achieved under authentication conditions and hash values that are not an attempt to guess at any adversary. Hence, the proposed protocol stands with secure message authentication with uncompromised message security and privacy.

4.3.9. Denial of Service Attack

A denial-of-service (DoS) attack is one that attempts to shut down a system or network, rendering it inaccessible to its intended users. The DoS attacks happen by flooding the target with traffic, or by flooding the target with information, which can cause a crash [35]. This proposed protocol is active in mitigating DoS attacks within SM to AG as well as transmission. Because of (M1, M2, M3), when any entity receives the first check validated

time-stamped as $T_{SM} = T_{AG}$, it will have to verify the pseudo random number $R_{SM} = R_{AG}$. In this case, A tries to send a request to the intended user, but this is not possible without any knowledge of the above parameter. Besides, each execution generates new values and if somehow A can perform, the scenario request information will not match as it will detect an illegitimate user.

4.3.10. Session Key Security

This proposed scheme smart meter (SM) and aggregator point (AG) computes its session key by ECC based scalar multiplication group operation [36] thereby append such as, ($TOKEN_{SM}||Q_{AG}||T_{SM}$ $R_{SM}||$(.h)) and ($TOKEN_{AG}||R_{AG}||T_{AG}||SK_{AG}||$(.h)).This clearly indicates that $SK_{AG} = SK_{SM}$ is similar in both directions, Thus, communication between (SM) and (AG) is secure. Furthermore, if $SK_{AG} = SK_{SM}$ smart meter appended his/her $TOKEN_{SM}$ and received $TOKEN_{AG}$, it will match session keys for the next message transmission. After getting the session key with appended message, the aggregator point (AG) checks the smart meter (SM),which is stored previously, if the same value appears, it will look a safe session key as well as during transmission, no values change. Hence, $TOKEN_{SM}$ and $TOKEN_{AG}$ preserve the session key security.

4.4. Security Analysis A (Formal)

The possibility of an attacker is included in the CK model. Three participants were involved in the proposed protocol: SM, AG, and TA. The act of TA is only performed in the registration phase. After that, handshaking between SM and AG is frequent. A can interact with a protocol participant $\mathcal{O}$ which is SM_i AG_j. Into the attacker interactions with the queries, every oracle backs to one of them, which is reject, accept, and $\perp$. A can gain access to the public communication channel like an outcome, the message can be improved, replayed, new messages launched, and some messages rejected. Furthermore, because the trust authority is considered secure, A does not obtain any TA private information, thus, A can access all component and TA publicly available information; A can be an intruder or dishonest user for any system of the environment. Following queries are essential for description:

1. h (messages): For this oracle inquiry, A supposes compute a random value $1 \leq d \leq n - 1$ and h:$\{0, 1\}^L$ as message hash value.
2. Execution (SM, AG): This quiz is used to translate passive attacks, where the adversary detects messages that the participant changes during the protocol real-time execution.
3. Transmit ($\mathcal{O}$, messages): The query is fit for summarize an active attack, and it allows A to send message to $\mathcal{O}$ and received a response in accordance with the protocol description.
4. Session key-Reveal ($\mathcal{O}$): Provides oracle with information about an ephemeral secret to A.
5. Session-Reveal ($\mathcal{O}$): In this examination, A can be informed of the session key construct by $\mathcal{O}$.
6. Corrupt-Massage ($\mathcal{O}$): This module indicates A can be used to obtain an oracle inlong time halt secret attributes from $\mathcal{O}$.
7. Expire-session ($\mathcal{O}$): This operation deletes the session secret parameter from the oracle-built session.
8. Test-Session ($\mathcal{O}$): Assuming that the goal of this query is to determine well-formed security of SK,$\mathcal{O}$. When a Test SK () request is issued, a neutral coin c is tossed; if c equals 1, the real SK is returned; if c equals 0, A randomly generated value of identical length is chosen and returned.

The following security definitions are addressed in order to ensure the proposed protocol conceptual security.

(1) If the Oracle O = SM and the Oracle O = AG are mutually checked and identified establish a similar SK_{SM} and SK_{AG} that are said to be partners.
(2) If the session key and corresponding session does not reveal any situation, we can call the session key is fresh.
(3) If Pr(succ.) imply the possibility that the opponent will succeed in achieving goal, then the opponent advantage in breaching LIKA semantic security is a)Adv_{LIKA} (A) = $|2_{Pr}(success) - 1|$.
(4) The proposed protocol is secure against the adversarial model if $Adv_{LIKA}(A) \leq \varepsilon$ holds [37].

Lemma 1. *(Difference Lemma). Suppose K_1, K_2 are the context of potential function K be the error event, in such a way: $K_1 \wedge \neg K \Leftrightarrow K_2 \wedge \neg$, K Then $|Pr[K1] - Pr[K2]|$ $Pr \leq [K]$.*

Theorem 1. *Assume that A is a computation adversary of semantic security that comes close to challenges like to M_S sending the query, M_E executing query, M_H hash inquiries, and M_T pseudo-random number inquiries. $|D|$ denotes the set equally spaced attributes. Adv_{EK}^{SE} Is advantage A over encryption or decryption ENC_K [38]. The benefits from LIKA protocol A are taken by:*

$$\frac{(M_L + M_P)}{L+1} + \frac{(M_P + M_E)^2}{q} + \frac{2M_P}{|D|} + Adv_{EK}^{SE} + 2M_H max\{Adv_{ECDLP}(A),\ Adv_{ECDHP}(A)\}$$

Proof. To demonstrate LIKA semantic security, we present the game theory progression, where game 0 represents the active and passive attack holt behind game 5 represents the game in which A has no advantage. □

Game 0: The simulation of GM 0 is similar to the actual attack in the random oracle model we have

$$Adv_{LIKA}(A) = |2Pr[Succ_0]\ 1| \quad (1)$$

Game 1: In GM1, the following outcomes of emulating oracles for multiple executions of the queries are stored and recorded, making it impossible for an opponent to distinguish between two games. As a result:

$$Pr[Succ1] = Pr[Succ0] \quad (2)$$

Game 2: Game 2 scenarios are similar to game 1, with the exception that GM2 terminates if there is a collision across hash queries. According to the birthday paradox, the possibility of hash collision is at maximum and $\frac{(M_L+M_P)}{L+1}$ the likelihood of a transcribed collision is at $\frac{(M_P+M_E)^2}{q}$ maximum, where L is the number of the hash function length.

Game 3: GM3 simulation is parallel to GM2 except that GM3 will be terminated if the verifier conditions are correctly predicted without consulting the oracle. Unless this valid authentication value is declined by the server instance, GM3 and the last game are indistinguishable. Thus:

$$Pr[Succ3] = Pr[Succ2] \leq \frac{(M_P)}{L} \quad (3)$$

Game 4: In GM4, we consider the suggested protocol session key reliability. The security goal of LIKA is that A cannot achieve the mutual session key. The goal SK wrap, zero knowledge by A thus the SK holds by

$$Pr[Succ4] = Pr[Succ3] \text{ holds } SK_{SM} \times Q_{AG}.G \text{ and } SK_{AG} \times Q_{SM}.G = SK_{SM} = SK_{AG} \quad (4)$$

Event 1: Corrupt (SM) and Corrupt (AG). This event is supposed to be A obtaining SM and AG private keys, i.e., (Ọ) and SK, but not their temporal secrets in terms of them being hidden by $Q_{AG} \times G$ point without having knowledge that it does not reveal.

Event 2: Session-reveal (SM) and session-Reveal (AG). In this case, we assume that A obtains SM and AG public parameter but not their secret keys. Without initially getting the session key, it is impossible to solve the SK in the following instances it should settle the hash function or crack ECDLP and ECDHP mathematical solution which is infeasible due to the hard and complex equation. As a result, the ECDHP or ECDLP holds, and it is concluded that:

$$\Pr [\text{Succ4}] = \Pr [\text{Succ4}] \leq SK_{AG} \times Q_{SM}.G + M_L \{Adv_{ECDLP} (A), Adv_{ECDHP} (A)\}$$

Game 5: GM5 is the same as the previous game, except that if you publish a test ID instead into the act as anonymous SM_{TOKEN}, AG_{TOKEN} query to get the actual identity which is not easy to retrieve due to it exits secure channel between trusted authority, the otherwise revealed test inquiry of this game will be disclose and A will win the game 5. As a result, we deduce:

$$\Pr [\text{Succ5}] = \Pr [\text{Succ4}] \leq \frac{2M_P}{|ID|} + Adv_{EK}^{PE}(A) \quad (5)$$

As a result, A without performing a hash query and private key with the true input has no position in distinguishing the actual SK from use by a random one, Pr [Succ6]. We can conclude that the theorem holds by adding all of these probabilities. In Algorithm 3 illustrate the random oracle simulation model.

Algorithm 3. Simulation of Random Oracles.

Simulation Queries
Hash queries (p). If p is found in the index list of T_L (p), the value L (p) is returned. Otherwise, a randomly generated value will be appended to L:{0, 1} for A
For the send(SM, start) query, the SM oracle must first log in(Registration) to the server with a login request attemptID and PW are chosen
Select ID = SM_{ID}
Select pw = SM_{PW}
Then respond with message C = { SM_{ID}, SM_{PW}}
For the send (SM, {SM_{ID}, SM_{PW}}) query the TA oracle simulate the following process:
Check $ID_{SM=}$ ID_{SM} into TA database
Check PW = PW into corresponding ID
$ComputeToken_{SM}$
Then respond with message C = {$Token_{SM}$}
For the send (TA, {$Token_{SM}$}) query the SM oracle simulate:
Store $Token_{SM}$
For the send(SM, start) query, the SM oracle secondly toward to mutual authentication: A random number $P_{SM} \in Zp^*$ private key, and public key is calculated by as $Q_{SM =} d_{SM} \times G$
Select = $Token_{SM}$
Select public key = Q_{AG}
Compute = T_{SM}
Compute SM = $R_{SM} \times P$
Compute hash (H) = (.h)
Then respond with message E { M1 = $Token_{SM} \| Q_{AG} \| T_{SM} \| R_{SM} \| (.h)$}

Algorithm 3. *Cont.*

For the send (SM, { $Token_{SM}||Q_{AG}||T_{SM}||R_{SM}||(.h)$}) query the AG oracle simulate the following steps:
Calculates private key = d_{AG} to extract message
Validity check of (H) = (.h)
Validity check of = $T_{SM} \leq T_{AG}$
Verify pseudo number AG = $R_{SM} \times G_P$
$StoreToken_{SM}$
Generates $R_{AG} = R_{AG} \times P$
Select = T_{AG}
Select = $Token_{AG}$
Calculates = SK_{AG}
Then respond with message E { M2= $Token_{AG}||R_{AG}||T_{AG}||SK_{AG}$}
For the send (AG, { $Token_{AG}||R_{AG}||T_{AG}||SK_{AG}$ query the SM oracle simulate
Compare $T_{AG} \leq T_{SM}$
Calculates $SK_{AG} = SK_{SM}$
Select = $Token_{AG}$
Select = T_{SM}
Then respond with message E { M3 =$SK_{SM}||Token_{AG}||T_{SM}$}
For the send (SM, { $SK_{SM}||Token_{AG}||T_{SM}$}) query the AG oracle simulate
Verify $SK_{AG} = SK_{SM}$
Verify = $T_{SM} \leq T_{AG}$
Check = $Token_{AG}$= $Token_{AG}$
Send:M4 = ACK_massage to SM
For execute (SM and AG) using the following send queries first retain {$Token_{SM}$ } ← TA in secure channel then {M1, M2,}← send (SM, start), and {M1, M2,}← send (AG, start) then return them to A
For a Session-Reveal(O) inquiry, re-back the session key if O has actually generated the session key and both O and its partner have not been asked by a test query or return null.
For a Session key-*Reveal*(O) inquiry, if O= SM return a_i; else; if O= AG return.
For a *Corrupt* (O) inquiry, if O= SM return SM*r*; else, if O= AG return AG*r*
For a *Test* (O) queries, firstly get the session key of O from the session (O) test; then, tossed a coin, if $c = 1$, return value of the session key, or else, return generated value with the same length to A

4.5. Security Analysis B (Formal)

Formal assessment 2 was adopted based on schemes [9,14,29] to propose a protocol of mutual authentication and forward secrecy between a smart meter and an aggregator point. This section critically verified the proposed protocol used by Avispa, which is a kind of role-based tool that measures the security of a cryptographic method automatically like public key infrastructure, encryption, decryption, and signature generation, hash function while symmetric encryption and decryption handled within the system. It supports unlimited simulation of session keys and message basements and is effective in determining protocol accuracy. Avispa is a set of applications that enable automatic verification with different backsides of Internet security protocols and applications.

Some examples include the On-the-Fly Model Checker (OFMC) and the Constraint-Logic-based Attack Searcher (Cl-AtSe). Avispa scripting is done entirely in a role-based language known as high level protocol specification language (HLPSL). According to principles of Avispa, specifications are based on the protocol that all states whether it is acceptable in terms of active and passive attackers with the assistance of the given proposed protocol. Further, this suggested protocol provides forward secrecy and authentication for the SK1, SK2 during smart meter and aggregator point session.

To validate the proposed protocol, Avispa was implemented usingSpan-Ubuntu10.10 light into OracleVM virtual box version 6.1.18 which is open source application developed by University of Genova from FET project. Span is a core application for written scripting according to setup protocol (refer to Table 2) for superior perception, which is performed

in HLPSL, while the Dolve–Yao model has been considered an intruder model [39]. The security of this protocol is discussed under the role of the user agent (SM) and for another user (AG) as dummies named A and B, respectively. There are four roles in Span to justify the testing protocol, namely user agent, smart meter, and aggregator point. In this role section, both parties contain their parameters to transfer the value between them to verify the purpose in Span. Then, the state section describes which parameters pass by using SND () and RCV () functions within the two parties to establish the session. After validating a specific function and formulating it, the composition event section declares the number of identical parameters this event occupied to a common parameter, which are examined prior to each other in order to initiate a session. Moreover, the results section responsible for analysing the proposed protocol by two model checkers are known as (OFMC) and (Cl-AtSe). Lastly, after evaluating those sections, Span will declare the status as safe or unsafe. Notable in the case for LIKA, it produced a safe status from active and passive attackers, while it evaluated the number of sessions and piles operation handled by A, B to strongly manage authentication between SK1, SK2 and forward secrecy. Figure 3 shows that the proposed protocol is safe for attack scenarios.

4.6. Security Features Comparison

In this section, the security features offered by the built authentication and key agreement protocol are evaluated based on the current solution.

Table 3 demonstrates the notation of the security comparison based on [40], the names of the attacks in forming from A1 to A13. Additionally, $\surd$ is presented for able to protect the following protocols, beside $\times$ used for unable to fix the attack scenario.

Table 4 indicated that the [16] protocol covered A1, A2, A6, A11, although it does not withstand the rest of the security features. Similarly [40] protocol kept A1, A2, A4, A5, A11, A12 and others features were unable to manage in their methodology. Then [17] worked efficiently maintain A1 to A4 and A8, A12 but failed against A6, A7 and A9 to A12. Protocol [41] was able to protect A1 to A7, A12, A13 beside failing against A8 to A11. Then protocol [1] achieved A1, A2, A3, A4, A7, A9, A11, but was unable to protect A5, A6, A8, A10, A12, A13 [12]. According to the protocol [42] security analysis claimed that kept A1 to A6, A11 and protocol [43] preserve A1-A5, A8, A10, A12 but the rest of the features was not covered by the following work. After protocols [25] managed A1 to A6, A11, A13 along does not withstand the rest of the security features. Then the protocol [27] withstand A1 to A4 and A8 but unfortunately unable to protect the rest of the features. Similarly the protocol [9] performed almost the same with the proposed protocol, but unfortunately it does not secure against A7, A8, A13. Moreover, The [14] protocol managed A1, A2, A3, A4, A7, A11, and A12, but does not withstand A5, A6, A8, A9, A10. After that [15] tackled a number of multiples attack known as A1 to A7, A9, A10 before it failed against A8, A12, A13, this is nearly close to the proposed protocol where the proposed work provides larger security features that ensures three-way security confirmation, namely informal cryptographic parameter analyses and CK adversarial basis, and then formal assessment employed by Avispa to following protocol to withstand A1 to A13 security features.

Table 2. Avispa role specification.

```
role smart meter (A, B: agent,
m1: text,
%H is SHA512 hash function
%ECDH is elliptic curve DH function
H: hash_func.
EC: hash_func
Ka, Kb: public_key,
SND_B, RCV_B: channel (dy))
played_by the initiator smart meter
def=
local State: nat
ID, Qag, Tsm, Rsm: text,
H: hash_func
M1: text,
init State:= 0
transition
1. State = 0 /\ RCV_B(start) = | >
State':= 1 /\ M1:= new()
% Send the mutual authentication request
to AG
/\ ID':=(ID)
/\ Qag':= (dsm × G=Qsm)
/\ Tsm': = (Tsm × G)
/\ Rsm': = (Rsm × G)
/\ Snd_A(ID'.Qag'.Tsm'.Rsm'.H)
2. State = 0 /\ RCV_M2 (ID, Tsm, Rsm,
Sk1) = | >
State':= 1 /\ M2:= new()
% Received acknowledgement from AG
/\ ID':=(ID)
/\ Tsm': = (Tsm × G)
/\ Snd_M3 (M, ID, Tsm, SK2)
role session(A, B: agent,
Ka, Kb: public_key,
G: text,
H: hash_func)
def=
local ID, Tsm, Rsm, H: channel (dy)
composition
SM(ID, Tsm, Rsm, a, b, p, G, n, h)
/\ AG(ID, Tsm, Rsm, a, b, p, G, n, h)
end role

role environment()
def=
const sk1,sk2: protocol_id,
a, b: agent,
ka, kb, Qsm: public_key,
M1,M2,M3: text,
H: hash_func
Intruder_knowledge = {Qsm, Qag, a, b, G,
h, n, p}
composition
session(a, b, G, n, h, p, Qsm, Qag, Tsm,
Rsm, H)
/\ session(M, Tsm, ID, SK1, SK2)
end role
goal
%secrecy_of SK2
secrecy_of sec_a_SK,2 sec_b_SK1
% Smart meter authenticates on Ag sk1
authentication_on sk1 % Addresses M1,
M2, M3
%Aggregator point authenticates on sk2
authentication_on sk2 % Addresses M1,
M2, M3
end goal
environment()
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>
role aggregator point (B, A:agent,
m1: text,
%H is SHA512 hash function
%ECDH is elliptic curve DH function
H: hash_func.
EC: hash_func.
Kb, Ka: public_key,
SND_A, RCV_A: channel (dy))
played_by responder aggregator point
def=
local State: nat
ID, Qsm, Tsm, Rsm: text,
H: hash_func
M1: text,
init State:= 1
transition
% Mutual authentication phase
% Received the authentication request
1. State = 0 /\
RCV_M1(ID'.Qag'.Tsm'.Rsm'.H) = | >
State':= 1 /\ secret {ID, Tsm, Rsm, H}(A)
/\ secret {H, Tsm, Rsm, Tsm}(B)
%send the mutual authentication
acknowledgement to SM
/\ ID': = TA(ID)
/\ Qag':= dsm × G(Qag)
/\ Tsm':= Tsm × G
/\ Rag':= Rsm × Gp
/\ H':= H=h'
% Send M2 to SM
/\ Snd (ID, Tsm, Rsm, SKag)
2. State = 0 /\ RCV_M3 (M, ID, Tsm, Sk1)
= | >
State':= 1 /\ M3:= new()
% Received acknowledgement from SM
/\ ID':=(ID)
/\ Tsm': = (Tsm × G)
/\ Rsm': = (Rsm × Gp)
/\ SK2': = (SK1=SK2)
role session(A, B: agent,
Ka, Kb: public_key,
G: text,
H: hash_func)
def=
local ID, Tsm, Rsm, H: channel (dy)
composition
SM(ID, Tsm, Rsm, a, b, p, G, n, h)
/\ AG(ID, Tsm, Rsm, a, b, p, G, n, h)
end role
role environment()
def=
const sk1,sk2: protocol_id,
a, b: agent,
ka, kb, Qsm: public_key,
M1,M2,M3: text,
H: hash_func
intruder_knowledge = {Qsm, Qag, a, b, G,
h, n, p}
composition
session(a, b, G, n, h, p, Qsm, Qag, Tsm,
Rsm, H)
/\ session(M, Tsm, ID, SK1, SK2)
end role
goal
%secrecy_of SK1
secrecy_of sec_a_SK2, sec_b_SK1
% Aggregator point authenticates on Ag
sk2
authentication_on sk2 % Addresses M1,
M2, M3
% Smart meter authenticates on sk1
authentication_on sk1 % Addresses M1,
M2, M3
end goal
environment()
```

```
%OFMC
%Version of 2006/02/13
SUMMARY
     SAFE
DETAILS

BOUNDED_NUMBER_OF_SESSIONS
PROTOCOLS

/home/span/span/testsuite/plank.
if
GOAL
      As specified
BACKEND
      OFMC
COMMENTS
STATISTICS
       ParseTime:0.0s
       Search Time:1.70s
       VisitedNodes:542 nodes
       Depth:14 piles
```

```
SUMMARY
 SAFE
DETAILS

BOUNDED_NUMBER_OF_SESSIONS
PROTOCOLS

      TYPED_MODEL

/home/span/span/testsuite/plank.if
GOAL
      As specified

BACKEND
CL-AtSe

STATISTICS
Analysed:4states
Reachable:4states
Translation:0.02seconds
Computation:0.12 sec
```

Figure 3. Avispa results by using OFMC and CL-AtSe back-ends.

Table 3. Notation of the Security Features.

A1	Replay attack
A2	MITM (Man in the middle)
A3	Impersonation Attack
A4	Forward secrecy
A5	Support Anonymity
A6	Eavesdropping
A7	Massage authentication
A8	Session key security
A9	Key freshness
A10	Dos Attack
A11	Guessing attack
A12	Mutual Authentication
A13	Three-way security proof

Table 4. Comparison Security Features.

Protocols	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
[16]	√	√	×	×	×	√	×	×	×	×	√	×	×
[40]	√	√	×	√	√	×	×	×	×	×	√	√	×
[17]	√	√	√	√	√	×	×	√	×	×	×	√	×
[41]	√	√	√	√	√	√	√	×	×	×	×	√	√
[1]	√	√	√	√	×	×	×	×	×	√	√	×	×
[12]	√	√	√	√	×	×	×	×	×	√	×	×	×
[42]	√	√	√	√	√	√	×	×	×	×	√	×	×
[43]	√	√	√	√	√	×	×	√	×	√	×	√	×
[9]	√	√	√	√	√	√	×	×	√	√	√	√	×
[14]	√	√	√	√	×	×	×	√	√	×	√	√	×
[15]	√	√	√	√	√	√	√	×	√	√	√	×	×
[25]	√	√	√	√	√	√	×	×	×	×	√	×	√
[27]	√	√	√	√	×	×	×	√	×	×	×	×	×
Proposed	√	√	√	√	√	√	√	√	√	√	√	√	√

4.7. Performance Analysis

This section discusses the performance analysis of the proposed protocol against previous works.

I. Computation Time

In order to use the implementation model and platform scenario for smart meter to aggregator point for LIKA adopted from protocol [24], which are the SM and AG point, consider two devices (1) computer Asus Intel(R) Core i5-8265U, CPU@1.80 GHz and 4 GB RAM (2) Raspberry PI 4, type B terminal with RAM 4GB. The purpose of data pre-processing for [12,17,18,40,41] schemes computation time were taken from [9,15,27] portal. The computation time of the previous protocols along with the proposed protocol based on various cryptographic operations run time performed using python version 3.4 into crypto library with associating import time module function [44], which is open sources programming language developed by python software foundation 20 February 1991. Following cryptographic execution operation performed among SM to AG as follow: T_{PA}: Times for point addition ≅0.097 ms, T_{PM}: Times for point multiplication ≅0.141 ms, $T_{E/D}$: Time for encryption/decryption ≅2.270 ms. Moreover, hashlib was used for generating T_{HO}: Time of hash operation ≅0.024 ms, then T_{RN}: Time of random number generator ≅ 0.009 ms and Ti: Time stamp expenses are ≅0.048 ms, T_{BP} = Time of bilinear pairing ≅3.32 ms [1,17] then $T_{PK}{}^{E/D}$ = Time public key encryption/decryption = 3.85 ms [14,15], and $1T_{HMAC}$ = Time for keyed hash message authentication code ≅ 0.0046 ms [12,14,42].

Part of the production of access tokens from the trusted authority setting are accomplished through the use of OpenSSL that acts as TA, according to link and preservation ID and password of smart meter and aggregator point information utilized structured query languages and hypertext processors database system.

According to Table 5 the heading interpretation proposed protocol required cryptographic operation, 2 times for point multiplication = 0.141 × 2 ms + 1 time encryption/decryption = 2.270 × 2 ms + 1time of hash operation = 0.024 ms + 2 time random number generator + 0.009 × 2 ms + 3 time-stamp ≅0.048 × 3 ms. This brings the total time expenses = (5.319 ms) for session key generation between smart meter to aggregator point. Existing works refs. [16,17,40] protocol took 30.80, 25.28, 22.28 ms. While protocols [1,12,14,41–43] expenses were approximately 18.34, 26.96, 15.01, 22.39, 16.16 and 11.92 ms. Furthermore, refs. [9,15,25,27] took 11.91, 17.80, 20.98, 17.86 ms respectively. According to the data link in Figure 4, the yielded value shows that the proposed protocol needed fewer computation times than existing protocols.

Table 5. Computations Time of the Previous/Proposed Protocols.

Serial Number	Protocols	Cryptographic Operation	Time (ms)	Compares*
1	[16]	$4T_{ME} + 4T_{PKED} + 2T_{HO}$	≈30.80	−25.48
2	[40]	$10T_{PM} + 2T_{ESED} + 5T_{HO}$	≈22.28	−16.96
3	[17]	$5T_{PM} + T_B + 2^{ECC}{}_{PA} + 10_H$	≈25.28	−19.96
4	[41]	$6T_{ME} + 4T_{ESED} + 4T_{PKED} + 2T_{HMAC}$	≈22.39	−17.07
5	[1]	$2^{ECC}{}_{PM} + 2^{ECC}{}_{PA} + 10_H$	≈18.34	−13.02
6	[42]	$1T_{PM} + T_{HMAC} + T_H$	≈16.16	−10.84
7	[43]	$3Tm + 3Ta + 3T_H$	≈11.92	−6.60
8	[12]	$6T_{ME} + 4T_{ESED} + 2T_{HO} + 2T_{HMAC}$	≈26.96	−21.64
9	[14]	$4T_{ME} + 4T_{PKED} + 2T_{HMAC}$	≈15.01	−9.69
10	[9]	$4Tm + 8^{Th}$	≈11.91	−6.59
11	[15]	$T_{PA} + T_{ESED} + T_{HMAC} + T_{ME} + T_{PKED} + T_{PM} + T_{HO}$	≈17.80	−12.48
12	[25]	5Tsm + 3Th+5Tsm + 3Th	≈20.98	−15.66
13	[27]	$2^{ECC}{}_{PM} + 3T_B + 3T_h$	≈17.86	−12.54
14	Proposed	$2T_{PA+}\ 2T_{PM+}\ 2T_{E/D+}\ 2T_{PRNG+}\ 3T_{TS+}\ 1T_{TOKEN} + 2_{HO}$	≈5.319	0

Compare* mean = proposed protocol minus (-) computation time from given protocols.

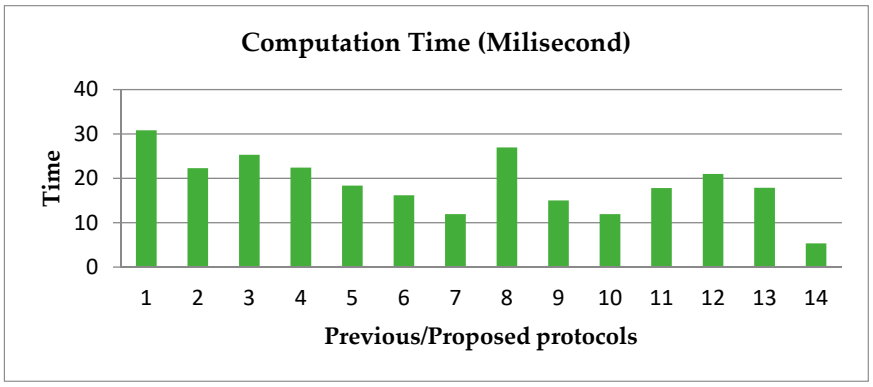


Figure 4. Comparison Computation Time among Previous Protocols.

- The computation cost proposed protocol $2T_{PA}+ 2T_{PM}+ 2T_{E/D}+ 2T_{PRNG}+ 3T_{TS}+ 2_{TOEKN}$ is 5.319 ms.
- The computation cost of the protocol [16] $4T_{ME} + 4T_{PKED} + 2T_{HO}$ is 28.80 ms which is 82.73%higher than the proposed protocol.
- The computation cost of the protocol [40] $10T_{PM} + 2T_{ESED} + 5T_{HO}$ which is 76.12%higher than the proposed protocol.
- The computation cost of the protocol [17] $5T_{PM} + TB+ 2^{ECC}{}_{PA} + 10_H$ which is 78.95% higher than the proposed protocol.
- The computation cost of the protocol [41] $6T_{ME} + 4T_{ESED}+ 4T_{PKED} + 2T_{HMAC}$ which is 76.24% higher than the proposed protocol.
- The computation cost of the protocol [1] $2^{ECC}{}_{PM} + 2^{ECC}{}_{PA} + 10_H$ which is 70.99% higher than the proposed protocol.
- The computation cost of the protocol [42] $1T_{PM} + T_{HMAC} + T_H$ which is 27.91% higher than the proposed protocol
- The computation cost of the protocol [43] 3Tm + 3Ta + 3Th which is 55.37% higher than the proposed protocol
- The computation cost of the protocol [12] $6T_{ME} + 4T_{ESED} + 2T_{HO} + 2T_{HMAC}$ which is 80.27% higher than the proposed protocol.
- The computation cost of the protocol [14] $4T_{ME} + 4T_{PKED} + 2T_{HMAC}$ which is 64.56% higher than the proposed protocol.
- The computation cost of the protocol [9] $4Tm + 8^{Th}$ which is 55.34% higher than the proposed protocol.
- The computation cost of the protocol [15] $T_{PA} + T_{ESED} + T_{HMAC} + T_{ME} + T_{PKED} + T_{PM} + T_{HO}$ which is 70.11% higher than the proposed protocol.
- The computation cost of the protocol [25] 5Tsm + 3Th+5Tsm + 3Thwhich is 74.64% higher than the proposed protocol
- The computation cost of the protocol [27] $2^{ECC}{}_{PM}+ 3T_B+3T_h$ which is 70.21% higher than the proposed protocol

II. Communication Bits

The communication bits cost of the proposed protocol is presented in Table 6 and compared with against existing protocols. This calculation was dependent on the number of messages sent and received by the smart meter and aggregator point, the number of bits connected with each transmission process during the session key generation in terms of which is performed in the key generation, registration and authentication phases. The size of the parameter constructed into the proposed protocol is carried on random 128 bits, ECC encryption and decryption 320 bits, time-stamps taken 64 bits with access token that consist of 64 bits and hash function 128 bits. The protocols communication costs transmission is determined by the number of messages 1, 2 and 3 transmitted bits associated with each handshaking process for session key generation. According to Table 6 value schemes [1,9,16] utilized three message expense are 1440, 1250 and 4416 bits. Consequently [17,40] protocols

used four message that took 2897 and 3158. Protocols [12,14,15,25,27,41–43] forwarded two massage needed 1752, 1984, 1152, 1120, 1465, 1792, 832 and 544 bits respectively. Finally, the proposed protocol employed three messages transmitted between the smart meter and aggregator point for entire session key generation, which is cost 1056 bits. As per displayed in Table 6 the proposed methodology efficiency is depicted in Figure 5. We conclude that, the proposed protocol requires less communication bits than other related existing schemes.

Table 6. Comparison Communication Bits among Previous Protocols.

Serial Number	Protocols	Forward Massage	Communication Cost (bits)	Compares*
1	[16]	3	≈4416	−3360
2	[40]	4	≈3158	−2102
3	[17]	4	≈2897	−1841
4	[41]	2	≈1792	−736
5	[1]	3	≈1440	−384
6	[42]	2	≈832	+224
7	[43]	2	≈544	+512
8	[12]	2	≈2752	−1696
9	[14]	2	≈1984	−928
10	[9]	3	≈1250	−194
11	[15]	2	≈1152	−96
12	[25]	2	≈1120	−64
13	[27]	2	≈1465	−409
14	Proposed	3	≈1056	0

Compare* mean = proposed protocol minus (-) communication bits from given protocols.

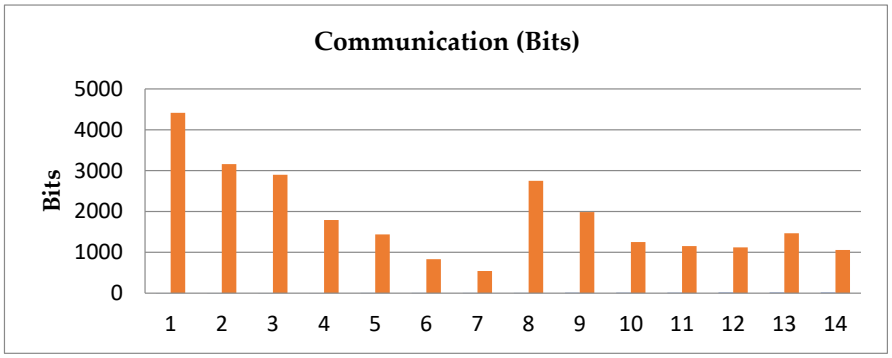


Figure 5. Comparison Communication Bits among Previous Protocols.

- The communication cost of the proposed protocol used 3 message which is required 1056 bits
- The Protocol [16] utilized 4416 bits, which is approximately 76.08% higher than the proposed protocol
- The Protocol [40] utilized 3158bits, which is approximately 66.56% higher than the proposed protocol
- The Protocol [17] utilized 2897 bits, which is approximately 63.54% higher than the proposed protocol
- The Protocol [41] utilized 1792 bits, which is approximately 41.07% higher than the proposed protocol
- The Protocol [1] utilized 1440 bits, which is approximately 26.66% higher than the proposed protocol
- The Protocol [42] utilized 832 bits, which is approximately 21.22% lower than the proposed protocol
- The Protocol [43] utilized 544 bits, Which is approximately 48.48% lower than the proposed protocol

- The Protocol [12] utilized 2752 bits, which is approximately 61.62% higher than the proposed protocol
- The Protocol [14] utilized 1984 bits, which is approximately 46.77% higher than the proposed protocol
- The Protocol [15] utilized 1152 bits, which is approximately 8.33% higher than the proposed protocol
- The Protocol [25] utilized 1120 bits, which is approximately 5.71% higher than the proposed protocol
- The Protocol [27] utilized 1465 bits, which is approximately 27.91% higher than the proposed protocol

III. Performance Comparison

In this section, we compare the systems performance of prior schemes to the proposed protocol.

Comparative performance comparison depend on features F1: Mutual authentication; F2: Forward secrecy; F3: Anonymity; F4: Three-way security proof (CK, Avispa/ProVerif and Informal); F5: Computation time= compare*; F6: Communication bits = compare*; F7: Lightweight Computation time Yes = 15>, No = <15; F8: Lightweight Communication bits Yes = 1500>, No =< 1500; NA: Not available; Hence, Yes is specify to reserve and No is unable to reserve following protocol. Since this comparison conducted from Tables 4–6, then we get the Table 7. Based on the table proposed protocol is ahead from entire competitive scheme. The proposed protocol retains greater number system features than existing work. Additionally, it demonstrated that it covers both security and performance aspects within the protocol. Besides, as per table interpretation previous schemes unable to integrate full of features into their schema.

Table 7. System Performance Comparison.

	2011 [16]	**2012 [40]**	**2014 [17]**	**2016 [41]**	**2018 [1]**	**2018 [12]**	**2019 [42]**	**2019 [43]**	**2019 [14]**	**2020 [9]**	**2020 [15]**	**2021 [25]**	**2021 [28]**	**Proposed**
F1	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
F2	Yes	No	No	NA	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes
F3	No	No	No	NA	Yes	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes
F4	No	No	No	No	No	Yes	No	No	Yes	No	Yes	Yes	Yes	Yes
F5	−25.48	−16.96	−19.96	−17.07	−13.02	−9.69	−1084	−6.60	−9.69	−6.59	−12.48	−15.66	−12.54	0
F6	−3360	−2102	−1841	−736	−384	−928	+224	+512	−928	−194	−96	−64	−409	0
F7	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
F8	No	No	No	No	Yes	No	Yes	Yes	No	Yes	No	Yes	No	Yes

5. Conclusions

Smart grids are a perfect fit for our rapidly connecting world and assure security and performance within the following field of utmost important to governments, businesses, academia, and so on. To address existing security and performance constraint, this paper proposed a lightweight identity based key agreement protocol that leverages on the elliptic curve Diffie–Hellman cryptography and used obey parameters for smart meter to aggregator points premises in advanced metering infrastructure. The proposed protocol combines security and performance features halt inside the same protocol. Security assessments were evaluated both in formal and informal aspects. In order to ensure security functionality, the proposed protocol was thoroughly tested by the CK adversary model and cryptographic parameters, which returned strong evidence to maintain forward secrecy and anonymity in term of preserving the mutual authentication. Furthermore, random oracle judgment declared that the session key does not consider any unauthorized user for interruption during session key generation. Consequently, the Avispa tool confirmed the designed key agreement approach is safe from multiple attacks scenarios. Further, performance analyses illustrated proposed protocol substantially reduced computation time and bit overheads. The execution depends on undergoing a three phase mechanism for the entire parameter computation operation expenses 5.319ms. Moreover, communication progression involved 3-way handshaking which took 1056 bits. Surprisingly, the outcome of the assessments

indicated that the overheads cost 1/3 of the previous scheme for session established among smart meter to aggregator point.

Despite this, a security evaluation ensures that reciprocal authentication is maintained within minimum time and bits expenses compare than other protocols. However, the protocol for smart meter and aggregator point side overheads are mostly similar. Besides, a smart meter contains limited resources, power and processing capabilities, the suggestion of this study though proposed work noteworthy, decreased overall time and bits operation. Besides, it will be diminished more in the smart meter end. In the near future, a smart grid context will be able to comply with the proposed protocol in a broad manner. With one recommendation, the mutual identity is balanced for vice versa devices, sensor, machine, and platform in smart grid of AMI. Finally, it is hoped that hierarchical architecture of key agreement protocols should improve the smart grid platform, robustness dynamism and secure.

Author Contributions: Conceptualization, and methodology, writing—and editing implement by M.M.H.; validation, formal analysis, supervision, project administration investigation under N.A.M.A.; review, visualization, and guideline N.F.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: Great thanks for financial support (Geran Putra IPM) from Universiti Putra Malaysia to publish this journal.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Special thanks from authors to Universiti Putra Malaysia for providing the working environment for me to perform this research, supervisor Noor Afiza Mohd Ariffin and committee member PM Fazlida Mohd Sani.

Conflicts of Interest: The authors declare no conflict of interest.

Notation

Symbols	Description
E	Elliptic curve
G	Generator point of E
P	Prime number
a, b	Coefficients of the elliptic curve
Q_{SM}	Public key of smart meter
Q_{AG}	Public key of aggregator point
d_{SM}	Secret key of smart meter
d_{AG}	Secret key of aggregator point
ID_{SM}, $ID_{AG,}$	Identity of SM and AG
T_{SM}, $T_{AG,}$	Time-stamp
(.h)	Output of hash value
R_{NUM}	Pseudo random number
SK_{SM}, SK_{AG}	Session key of SM, AG
$Token_{SM,}$ $Token_{AG}$	Token of the smart meter and aggregator point
A	Adversary
B_M	Real-time billing information
M1, M2, M3	Transmitted massage of SM and AG
SM, AG	Smart meter, Aggregator point
TA	Trusted authority
AMI, SG	Advanced metering infrastructure, Smart grid

References

1. Abbasinezhad-Mood, D.; Nikooghadam, M. An Anonymous ECC-Based Self-Certified Key Distribution Scheme for the Smart Grid. *IEEE Trans. Ind. Electron.* **2018**, *65*, 7996–8004. [CrossRef]
2. Perez-Guzman, R.E.; Salgueiro-Sicilia, Y.; Rivera, M. Communication systems and security issues in smart microgrids. In Proceedings of the 2017 IEEE Southern Power Electronics Conference (SPEC), Puerto Varas, Chile, 4–7 December 2017; pp. 1–6. [CrossRef]
3. Benmalek, M.; Challal, Y.; Derhab, A. Authentication for Smart Grid AMI Systems: Threat Models, Solutions, and Challenges. In Proceedings of the 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2019, Naples, Italy, 12 June 2019; pp. 208–213. [CrossRef]
4. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT Privacy and Security: Challenges and Solutions. *Appl. Sci.* **2020**, *10*, 4102. [CrossRef]
5. Gope, P. PMAKE: Privacy-aware multi-factor authenticated key establishment scheme for Advance Metering Infrastructure in smart grid. *Comput. Commun.* **2020**, *152*, 338–344. [CrossRef]
6. Chen, Y.; Martínez, J.-F.; Castillejo, P.; López, L. An Anonymous Authentication and Key Establish Scheme for Smart Grid: FAuth. *Energies* **2017**, *10*, 1354. [CrossRef]
7. Ariffin, N.A.M.; Rahim, F.A.; Asmawi, A.; Ibrahim, Z.-A. Vulnerabilities detection using attack recognition technique in multi-factor authentication. *Telkomnika* **2020**, *18*, 1998. [CrossRef]
8. Park, J.H.; Kim, M.; Kwon, D. Security Weakness in the Smart Grid Key Distribution Scheme Proposed by Xia and Wang. *IEEE Trans. Smart Grid* **2013**, *4*, 1613–1614. [CrossRef]
9. Garg, S.; Kaur, K.; Kaddoum, G.; Rodrigues, J.J.P.C.; Guizani, M. Secure and Lightweight Authentication Scheme for Smart Metering Infrastructure in Smart Grid. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3548–3557. [CrossRef]
10. Kumar, N.; Kaur, K.; Misra, S.C.; Iqbal, R. An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud. *Peer-Peer Netw. Appl.* **2016**, *9*, 824–840. [CrossRef]
11. Pires, F.M.; Quinonez, L.L.; Mendes, L.D.S. A Cloud-Based System Architecture for Advanced Metering in Smart Cities. In Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference IEMCON 2019, Vancouver, BC, Canada, 17–19 October 2019. [CrossRef]
12. Li, W.; Li, R.; Wu, K.; Cheng, R.; Su, L.; Cui, W. Design and Implementation of an SM2-Based Security Authentication Scheme with the Key Agreement for Smart Grid Communications. *IEEE Access* **2018**, *6*, 71194–71207. [CrossRef]
13. Wang, W.; Lu, Z. Cyber Security in the Smart Grid: Survey and Challenges. *Comput. Netw.* **2013**, *57*, 1344–1371. [CrossRef]
14. Wu, F.; Xu, L.; Li, X.; Kumari, S.; Karuppiah, M.; Obaidat, M.S. A Lightweight and Provably Secure Key Agreement System for a Smart Grid with Elliptic Curve Cryptography. *IEEE Syst. J.* **2019**, *13*, 2830–2838. [CrossRef]
15. Khan, A.A.; Kumar, V.; Ahmad, M.; Rana, S.; Mishra, D. PALK: Password-based anonymous lightweight key agreement framework for smart grid. *Int. J. Electr. Power Energy Syst.* **2020**, *121*, 106121. [CrossRef]
16. Wu, D.; Zhou, C. Fault-Tolerant and Scalable Key Management for Smart Grid. *IEEE Trans. Smart Grid* **2011**, *2*, 375–381. [CrossRef]
17. Fouda, M.; Fadlullah, Z.; Kato, N.; Lu, R.; Shen, X.S. A Lightweight Message Authentication Scheme for Smart Grid Communications. *IEEE Trans. Smart Grid* **2011**, *2*, 675–685. [CrossRef]
18. Nicanfar, H.; Jokar, P.; Beznosov, K.; Leung, V.C.M. Efficient Authentication and Key Management Mechanisms for Smart Grid Communications. *IEEE Syst. J.* **2014**, *8*, 629–640. [CrossRef]
19. Yan, L.; Chang, Y.; Zhang, S. A lightweight authentication and key agreement scheme for smart grid. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1–7. [CrossRef]
20. Tsai, J.-L.; Lo, N.-W. Secure Anonymous Key Distribution Scheme for Smart Grid. *IEEE Trans. Smart Grid* **2016**, *7*, 906–914. [CrossRef]
21. Braeken, A.; Kumar, P.; Martin, A. Efficient and Provably Secure Key Agreement for Modern Smart Metering Communications. *Energies* **2018**, *11*, 2662. [CrossRef]
22. Khan, A.A.; Kumar, V.; Ahmad, M. An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach. *J. King Saud Univ.-Comput. Inf. Sci.* **2019**, *34*, 698–705. [CrossRef]
23. Chaudhry, S.A. Correcting "PALK: Password-based anonymous lightweight key agreement framework for smart grid". *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106529. [CrossRef]
24. Farooq, S.M.; Hussain, S.M.S.; Ustun, T.S.; Iqbal, A. Using ID-Based Authentication and Key Agreement Mechanism for Securing Communication in Advanced Metering Infrastructure. *IEEE Access* **2020**, *8*, 210503–210512. [CrossRef]
25. Baruah, B.; Dhal, S. An Authenticated Key Agreement Scheme for Secure Communication in Smart Grid. In Proceedings of the 2021 International Conference on COMmunication Systems & NETworkS, COMSNETS 2021, Bengaluru, India, 5–9 January 2021; Volume 2061, pp. 447–455. [CrossRef]
26. Deng, L.; Gao, R. Certificateless two-party authenticated key agreement scheme for smart grid. *Inf. Sci.* **2021**, *543*, 143–156. [CrossRef]
27. Gupta, D.S.; Islam, S.K.H.; Obaidat, M.S.; Vijayakumar, P.; Kumar, N.; Park, Y. A Provably Secure and Lightweight Identity-Based Two-Party Authenticated Key Agreement Protocol for IIoT Environments. *IEEE Syst. J.* **2021**, *15*, 1732–1741. [CrossRef]

28. Panchbhai, M.M.; Ghodeswar, U.S. Implementation of point addition & point doubling for Elliptic Curve. In Proceedings of the 2015 International Conference on Communications and Signal Processing ICCSP 2015, Chengdu, China, 10–11 October 2015; pp. 746–749. [CrossRef]
29. Yu, S.; Park, K.; Lee, J.; Park, Y.; Park, Y.; Lee, S.; Chung, B. Privacy-Preserving Lightweight Authentication Protocol for Demand Response Management in Smart Grid Environment. *Appl. Sci.* **2020**, *10*, 1758. [CrossRef]
30. Odelu, V.; Das, A.K.; Wazid, M.; Conti, M. Provably Secure Authenticated Key Agreement Scheme for Smart Grid. *IEEE Trans. Smart Grid* **2018**, *9*, 1900–1910. [CrossRef]
31. Hasan, M.M.; Ariffin, N.A.M.; Sani, N.F.M. A review of cryptographic impact in cybersecurity on smart grid: Threat, challenges and countermeasures. *J. Theor. Appl. Inf. Technol.* **2021**, *99*, 2458–2472.
32. Pandey, R.K.; Ieee, S.M. Cyber Security Threats-Smart Grid Infrastructure. In Proceedings of the 2016 National Power Systems Conference (NPSC), Bhubaneswar, India, 19–21 December 2016.
33. Moghadam, M.F.; Nikooghadam, M.; Mohajerzadeh, A.H.; Movali, B. A lightweight key management protocol for secure communication in smart grids. *Electr. Power Syst. Res.* **2020**, *178*, 106024. [CrossRef]
34. Alohali, B.; Merabti, M.; Kifayat, K. A Cloud of Things (CoT) Based Security for Home Area Network (HAN) in the Smart Grid. In Proceedings of the 2014 8th International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST 2014, Oxford, UK, 10–12 September 2014; pp. 326–330. [CrossRef]
35. Anzalchi, A.; Sarwat, A. A survey on security assessment of metering infrastructure in Smart Grid systems. In Proceedings of the SoutheastCon 2015, Fort Lauderdale, FL, USA, 9–12 April 2015. [CrossRef]
36. Hankerson, D.; Menezes, J.A.; Vanstone, S. Elliptic Curve Arithmetic. In *Guide to Elliptic Curve Cryptography*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 75–152. [CrossRef]
37. Blanchet, B.; Smyth, B.; Cheval, V. ProVerif 1.91: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. 2016. Available online: https://bblanche.gitlabpages.inria.fr/proverif/manual.pdf (accessed on 3 March 2022).
38. Canetti, R.; Krawczyk, H. Analysis of key-exchange protocols and their use for building secure channels. In Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, 6–10 May 2001; pp. 453–474.
39. Genet, T. SPAN+AVISPA for Verifying Cryptographic Protocols. Available online: http://people.irisa.fr/Thomas.Genet/span/present_span.pdf (accessed on 15 March 2022).
40. Xia, J.; Wang, Y. Secure Key Distribution for the Smart Grid. *IEEE Trans. Smart Grid* **2012**, *3*, 1437–1443. [CrossRef]
41. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Shon, T.; Ahmad, H.F. A lightweight message authentication scheme for Smart Grid communications in power sector. *Comput. Electr. Eng.* **2016**, *52*, 114–124. [CrossRef]
42. Kumar, P.; Gurtov, A.; Sain, M.; Martin, A.; Ha, P.H. Lightweight Authentication and Key Agreement for Smart Metering in Smart Energy Networks. *IEEE Trans. Smart Grid* **2019**, *10*, 4349–4359. [CrossRef]
43. Garg, S.; Kaur, K.; Kaddoum, G.; Gagnon, F.; Ahmed, S.H.; Jayakody, D.N.K. LiSA: A Lightweight and Secure Authentication Mechanism for Smart Metering Infrastructure. In Proceedings of the 2019 IEEE Global Communications Conference GLOBECOM 2019, Hawaii, HI, USA, 9–13 December 2019. [CrossRef]
44. Deebak, B.; Al-Turjman, F. A smart lightweight privacy preservation scheme for IoT-based UAV communication systems. *Comput. Commun.* **2020**, *162*, 102–117. [CrossRef]