Hindawi
Security and Communication Networks
Volume 2022, Article ID 7125491, 11 pages
https://doi.org/10.1155/2022/7125491

Research Article

Cryptanalysis and Enhancement of an Authenticated Key Agreement Protocol for Dew-Assisted IoT Systems

Yuqian Ma, Yongliu Ma, and Qingfeng Cheng

State Key Laboratory of Mathematical Engineering and Advanced Computing, Strategic Support Force Information Engineering University, Zhengzhou 450001, China

Correspondence should be addressed to Qingfeng Cheng; qingfengc2008@sina.com

Received 28 March 2022; Accepted 18 May 2022; Published 15 June 2022

Academic Editor: Qi Li

Real-time and high-efficient communication becomes a vital property for IoT-enabled equipment, since the application range of the Internet of Things has extended widely. At the same time, the centralized characterization of the cloud computing is gradually unable to meet the demand for both low latency and high computing efficiency. To resolve these issues, new computing paradigms have been introduced, such as edge, dew, and fog computing. Recently, Saurabh et al. introduced a mutual authentication protocol, which was claimed to resist various attacks without the requirement of a trusted server, for dew-assisted IoT devices. However, this paper will show that Saurabh et al.'s scheme lacks forward security and user anonymity. Then, a new authenticated key agreement (AKA) protocol, named e-SMDAS, will be put forward and formally proven secure under the eCK security model. Further, the analysis results of BAN logic and Scyther tool will also confirm the security of e-SMDAS. Finally, the comparative analysis of security features and computation efficiency between e-SMDAS and several recent schemes will be demonstrated at the end of this paper.

1. Introduction

Cloud computing, developing swiftly and violently, is gradually unable to satisfy the growing needs in the Internet. Flavio et al. [1] introduced the idea of fog computing. However, with the rapid development of the Internet, fog computing alone could not satisfy the quality of cloud-assisted services. Some other computing paradigms were proposed to meet the growing demand for high-quality cloud services. Tian et al. [2] recently proposed a framework for blockchain-assisted edge services in the Industrial Internet of Things (IIoT). The paradigm of dew computing was put forward by Wang [3, 4] to fully make use of on-premises devices and cloud services. Defined as an on-premises device software-hardware organization paradigm in the cloud computing environment, the dew computing, in which dew servers are independent of cloud servers when offline and collaborative with cloud servers when online, provides the functionality of high information processing and low latency communication. The system architecture of cloud-fog-dew computing is demonstrated in Figure 1.

To build a secure and flexible dew computing paradigm, many security features need to be considered. Besides the basic mutual authentication and session key confirmation features, protocols in this paradigm also require forward security which confirms the leakage of long-term secrets will not influence the session keys. Since communications between servers are closely related to users' privacy, anonymity and untraceability are also vital.

To achieve secure communication in the network driven by fog computing, Hameed et al. [5] proposed a scheme claiming that it could achieve mutual authentication, low consumption, and high efficiency in smart home case. In 2021, Liu et al. [6] proposed a distributed access control system based on the decentralized conception of fog computing and blockchain technology. A similar idea was also thought about by Shukla et al. [7], adopting a signature-based encryption algorithm to maximize the strength of fog computing and blockchain.

The application field of the Internet of Things (IoT) has extended largely in recent years. Aiming at protecting

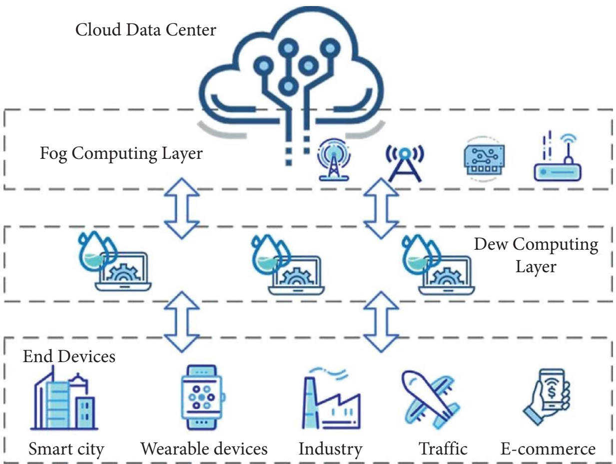

Figure 1: Fog computing system architecture.

the secrecy, integrity, and anonymity of IoT-assisted end devices, Singh and Chaurasiya [8] discussed a possible mutual authentication scheme for the vulnerable fog nodes. A combination of elliptic curve Diffie–Hellman ephemeral key exchange algorithm and preshared key was analyzed by Amanlou et al. [9] to achieve credible communication between the fog gateways and devices located in IoT.

Our contributions in this paper mainly consist of the following four points.

(i) We analyze an authenticated key agreement (AKA) protocol designed for a dew-assisted system by Saurabh et al. [10], referred to as SMDAS protocol below, and point out that their scheme lacks forward security and user anonymity.

(ii) Upon the analysis, we design a new AKA protocol, called e-SMDAS protocol below, remedying SMDAS protocol to achieve the mutual authentication, session key establishment, forward security, user anonymity, and other security features.

(iii) The security of our protocol is formally proven under the eCK security model and also confirmed using the Scyther tool and BAN logic.

(iv) Finally, results of comparison between the enhanced protocol and several recent schemes demonstrate the advantages of our protocol in the aspects of security features and communication efficiency.

The arrangement of this paper is as follows. Related works are first introduced in Section 2. In Section 3, we present some preliminaries used in the analysis of the proposed protocol. After reviewing the process of SMDAS protocol in Section 4, we analyze the security flaws of SMDAS protocol in Section 5. Our newly proposed protocol is described explicitly in Section 6; its formal security proof and security analysis using Scyther tool and BAN logic are provided in Section 7. Comparisons between the proposed protocol and SMDAS protocol are demonstrated in Section 8. Finally, in Section 9, the conclusion is highlighted.

2. Related Work

So far, anonymity and privacy-preserving are vital security features required urgently not only in dew computing paradigm but also in many other applications. To sum up the applications, several relative schemes [11–16] are listed in Table 1. They have been paid much attention to because of the decentralized feature of dew-assisted paradigm [17].

Recently, a lightweight anonymity client authentication scheme was proposed by Gaikwad et al. [18] adopting chaotic hash function. Moreover, Masud et al. [19] proposed a lightweight and physically secure mutual authentication and secret key establishment protocol preserving privacy for COVID-19 patients' care in the Internet of Medical Things. Their protocol used physical unclonable functions to make the network devices distinguish the legitimacy of doctors before acquiring a session key. Xiong et al. [20] proposed a three-party data privacy-preserving mechanism with game theory and machine learning technology. Tian et al. [21] proposed a graph clustering method to protect data privacy sharing in the Social Internet of Things (SIoT).

Besides, forward security is one of the main concerns for AKA protocols. In 2015, Chaudhry et al. [22] proposed a remote user authentication scheme. Regrettably, Ravanbakhsh et al. [23] claimed that Chaudhry et al.'s scheme was unable to achieve perfect forward security and proposed an authenticated communication scheme for Voice over Internet Protocol (VoIP). Later, Nikooghadam and Amintoosi [24] proved that Ravanbakhsh et al.'s scheme did not provide perfect forward security and put forward a two-factor AKA scheme with perfect forward security.

Recently, Saurabh et al. [10] introduced a mutual AKA protocol for the dew-assisted devices. They applied bilinear parings to achieve the mutual authentication and establishment of secure session keys. Formal analysis was presented by the use of AVISPA and the theory of security reduction. However, in this paper, we analyze the security of this protocol and show that it lacks forward security and user anonymity.

3. Preliminaries and Security Model

In this section, we concisely introduce the mathematical definitions and security model used next.

3.1. Mathematical Hard Problems

(i) *Elliptic Curve Discrete Logarithm (ECDL) Problem*: Given an elliptic curve E_p, an additive cyclic group G based on E_p, a generator P of G, and an element $Q = aP$ from G, it is hard to extract $a \in Z_p^*$ from Q and P.

(ii) *Elliptic Curve Computational Diffie–Hellman (ECCDH) Problem*: Given an elliptic curve E_p, an additive cyclic group G based on E_p, and a generator P of G, considering the elements $S = aP$ and $T = bP$ from G, it is hard to compute $U = abP$.

Table 1: The summary of schemes set in IoT systems.

Scheme	Settings applied in	Limitations
[11]	Wireless sensor networks	Vulnerable to insider attack
[12]		Vulnerable to secret key leakage and forgery attack
[13]	Telecare medicine information systems	Vulnerable to reflection attack
[14]		Vulnerable to replay attack
[15]		Vulnerable to offline password attack
[16]		Vulnerable to impersonation attack and users' identity leakage

3.2. Security Model. LaMacchia et al. [25] proposed the eCK security model in 2007. In this model, each entity owns two secrets, a long-term key x and an ephemeral key r. Assume two entities are A and B; their long-term keys are x_A, x_B; and their ephemeral keys are r_A, r_B, respectively. Besides, each session under the eCK security model has its own identity, denoted as $\mathrm{SID}^i_{A,B}$ if this session's owner is entity A. Then, the abilities of adversary, denoted as $\mathscr{A}$, can be defined through the queries below:

(i) $\mathrm{Send}(A, M)$: Through this query, $\mathscr{A}$ can send message M to entity A and get the corresponding message according to the protocol.

(ii) $\mathrm{Reveal}(\mathrm{SID}^i_{A,B})$: Through this query, $\mathscr{A}$ can acquire the session key of $\mathrm{SID}^i_{A,B}$ if session $\mathrm{SID}^i_{A,B}$ has been completed. Otherwise, $\mathscr{A}$ will get nothing.

(iii) $\mathrm{Ephemeral}(\mathrm{SID}^i_{A,B})$: Through this query, $\mathscr{A}$ can obtain the ephemeral key of the session $\mathrm{SID}^i_{A,B}$.

(iv) $\mathrm{Longterm}(A)$: Through this query, $\mathscr{A}$ can obtain the long-term key of entity A.

(v) $\mathrm{Test}(\mathrm{SID}^i_{A,B})$: If $\mathscr{A}$ launches this query, session $\mathrm{SID}^i_{A,B}$ will randomly choose b from $\{0, 1\}$. If $b = 0$, $\mathrm{SID}^i_{A,B}$ will choose a random number from the set of keys and send it back to $\mathscr{A}$. If $b = 1$, $\mathrm{SID}^i_{A,B}$ will send the real session key back to $\mathscr{A}$.

To define a secure protocol in the eCK security model, a definition of freshness should be presented first since a secure game through $\mathrm{Test}(\mathrm{SID}^i_{A,B})$ is querying toward a fresh session.

Definition 1. A session with identity $\mathrm{SID}^i_{A,B}$ in the eCK model at entity A whose intended partner denoted as B is fresh if the following items are satisfied:

(i) The session has not been asked for a Reveal query.

(ii) If a matching session exists with session identity $\mathrm{SID}^j_{B,A}$, then

  (i) not both $\mathrm{Ephemeral}(\mathrm{SID}^i_{A,B})$ and $\mathrm{Longterm}(A)$ queries have been asked for;

  (ii) not both $\mathrm{Ephemeral}(\mathrm{SID}^j_{B,A})$ and $\mathrm{Longterm}(B)$ queries have been asked for.

(iii) If no partner exists, then

  (i) not both $\mathrm{Ephemeral}(\mathrm{SID}^i_{A,B})$ and $\mathrm{Longterm}(A)$ queries have been asked for;

  (ii) $\mathrm{Longterm}(B)$ queries have not been asked for.

Based on this definition, we present the definition of a secure session in the eCK security model.

Definition 2. The advantage of the adversary $\mathscr{A}$ in the secure game with AKA protocol Π is defined as $\mathrm{Adv}^{\mathrm{AKA}}_{\Pi}(\mathscr{A}) = \Pr[A\,\mathrm{wins}] - 1/2$.

If the matching session of Π computes the same session key and no efficient adversary $\mathscr{A}$ has more than a negligible advantage in winning the secure game, then the protocol Π is secure under the eCK security model.

4. Review of SMDAS Protocol

In this section, we review the registration and session key distribution phases of SMDAS protocol [10]. There are three types of entities participating in SMDAS protocol, namely, a sensor node SN_i, a dew server DS_j, and a cloud server S. Notations used in SMDAS protocol are listed in Table 2.

4.1. Registration Phase. Firstly, the cloud server S initializes this system according to the following steps.

(i) S selects an appropriate elliptic curve E over a finite field F_q and then selects G, a subgroup of E, whose order is n. P is a group generator of G.

(ii) S randomly chooses $s \in Z^*_n$ and calculates $X = sP$, $A = e(P, P)^s$.

(iii) Finally, S publishes the public parameters $\{E, G, A, n, P, X\}$ and keeps s as its own secret key securely.

4.2. Dew Server Registration Phase. Assume that there are m dew servers and each one is denoted as DS_j, $j \in \{1, 2, \ldots, m\}$. These servers select their own identities $\mathrm{ID}_{\mathrm{DS}_j}$. When a dew server registers to the cloud server, it sends its identity $\mathrm{ID}_{\mathrm{DS}_j}$ to S. After receiving DS_j's identity, S will compute $\mathrm{SID}_{\mathrm{DS}_j}$ for DS_j, where $\mathrm{SID}_{\mathrm{DS}_j} = s(X + P \cdot h(\mathrm{ID}_{\mathrm{DS}_j}))$.

4.3. Sensor Node Registration Phase. Every sensor node, denoted as SN_i, has its own identity $\mathrm{ID}_{\mathrm{SN}_i}$ and password $\mathrm{PW}_{\mathrm{SN}_i}$. When the sensor node needs to register to S, it firstly computes $\mathrm{SH}_1 = h(\mathrm{ID}_{\mathrm{SN}_i} \| \mathrm{PW}_{\mathrm{SN}_i})$ and sends message $\mathrm{ID}_{\mathrm{SN}_i}, H_1$ to S. Upon receiving the registration request from SN_i, S verifies $\mathrm{ID}_{\mathrm{SN}_i}$ to confirm SN_i is an unregistered node. Then, S computes $I = h(\mathrm{ID}_{\mathrm{SN}_i} \| s)$, $H_2 = I \oplus H_1$, $\mathrm{SID}_{\mathrm{SN}_i} = s(P + I)$. After computing, S stores SID_i and sends message $H_2, \mathrm{SID}_{\mathrm{SN}_i}$ to SN_i. When SN_i receives message from S, it computes $I = H_2 \oplus H_1$ and stores $\mathrm{SID}_{\mathrm{SN}_i}, I$.

Table 2: Notations applied in SMDAS protocol.

Parameters	Description
S	The cloud/fog server
SN_i	The sensor node i
DS_j	The dew server j
ID_{SN_i}, ID_{DS_j}	The identity of SN_i, DS_j, respectively
S	The secret key of S
PW_{SN_i}	The password of sensor node i
T_i, T_j	Timestamp generated by SN_i, DS_j, respectively
$h(\ )$	One-way hash function defined from Z_n^* to Z_n^*
$\mathscr{A}$	The adversary

4.4. Session Key Distribution Phase. After DS_j and SN_i register to S, they can establish a session with SID_{SN_i} and SID_{DS_j}. The detailed steps are described below.

(i) SN_i randomly chooses $r_u \in Z_n^*$ and computes the corresponding public key $R_u = r_u P$ and $Z = A^{r_u}$. Then, SN_i calculates the elements of message as follows: $M = R_u + (X + P \cdot h(ID_{DS_j}))$, $N = h(Z) \oplus ID_{SN_i}$, $Q = h(Z \| ID_{SN_i} \| X) \oplus R_u$, $S = SID_{SN_i} \oplus h(R_u \| ID_{SN_i} \| TS_i \| Z)$, $J = h(SID_{SN_i} SR_u NQID_{SN_i} TS_i)$. SN_i sends $M, N, Q, S\ J, TS_i$, where TS_i is the current timestamp.

(ii) DS_j computes Z', ID_{SN_i}', R_u', and SID_i'. According to these parameters, DS_j verifies whether J' equals J. DS_j randomly selects $y \in Z_n^*$ and computes the public key $Y = yP$. Then, DS_j calculates $T_j = h(ID_{SN_i}' \| ID_{DS_j} \| R_u' | Y | TS_j)$, $F = SID_{SN_i}' \oplus T_j$, $SK = h(SID_{SN_i}' \| T_j \| TS_j)$, $V_e = h(SK \| T_j \| F \| TS_j)$. DS_j sends message TS_j, V_e, F, where TS_j is the current timestamp and stores the session key SK.

(iii) SN_i computes T_j', SK', and V_e'. According to these parameters, SN_i verifies whether V_e' equals V_e. If it succeeds, SN_i accepts SK' as the session key.

5. Cryptanalysis of SMDAS Protocol

In this section, we present two security flaws of SMDAS protocol as the adversary $\mathscr{A}$ can acquire private key of SN_i and DS_j through $Extract(ID_{SN_i})$ and $Extract(ID_{DS_j})$, respectively, mentioned in [10].

5.1. Lack of Forward Security. In this subsection, we demonstrate if the private key of sensor node i is compromised; then, the session key will be easily recovered by the adversary $\mathscr{A}$:

(i) In the session key distribution phase, $\mathscr{A}$ eavesdrops the message from dew server to sensor node, TS_j, V_e, F.

(ii) $\mathscr{A}$ launches Extract query to the sensor node SN_i and acquires SN_i's private secret keys SID_{SN_i}.

(iii) After obtaining the parameters above, $\mathscr{A}$ can extract T_j' by $T_j' = F \oplus SID_{SN_i}$ and the session key according to the way generating $SK = h(SID_{SN_i} \| T_j' \| TS_j)$.

Thus, in this way, adversary $\mathscr{A}$ can recover the session key. It can be concluded that the steps described are in accordance with the definition of weak forward security.

5.2. Lack of User Anonymity. We point out an efficient method to prove that SMDAS protocol lacks user anonymity in this subsection by compromising the private key of dew server following the steps below.

(i) $\mathscr{A}$ first eavesdrops the message M, N, Q, S, J, TS_i.

(ii) Then, $\mathscr{A}$ launches $Extract(ID_{DS_j})$ to get the private key of DS_j, SID_{DS_j}.

(iii) In this way, $\mathscr{A}$ can compute $Z' = e(M, X)/e(SID_{DS_j}, P)$.

(iv) Finally, the adversary can derive the identity of SN_i as $ID_{SN_i} = N \oplus h(Z')$.

When the adversary implements the attack described above, $\mathscr{A}$ can easily get the identity of the sensor node. This means SMDAS protocol can hardly protect the anonymity of users.

6. e-SMDAS Protocol

In this section, we propose a new anonymity and secure mutual AKA protocol remedying the flaws of SMDAS protocol, which we call e-SMDAS protocol.

There are three main phases in the proposed protocol, namely, initialization phase, registration phase, and secure session key establishment phase. Particularly, the registration phase can be divided into two parts, the sensor node registration phase and the dew server registration phase. In Table 3, the notations applied in the proposed protocol are presented.

6.1. Initialization Phase. The cloud server, also the registration server, acts as the trusted authority. It first selects a suitable cyclic group G based on an elliptic curve E. The order of the group is the prime p and the generator of the group is P. Then, the server randomly selects $s \in Z_p^*$ as its master key while it computes its public key $X = sP$ accordingly and defines the three hash functions h_1, h_2, h_3. Finally, the server publishes the public parameters $\{E, G, P, X, p, h_1, h_2, h_3\}$ to initialize the system and keeps s secretly.

6.2. Registration Phase. Before sensor nodes and dew servers are put into usage, they must be registered in the cloud server first to acquire their long-term keys in the further communications. Both the sensor node registration phase and the dew server registration phase are described as follows.

6.2.1. Sensor Node Registration Phase. Before SN_i registers in the cloud server S, SN_i should first choose its identity ID_{SN_i} and password PW_{SN_i}. Then, SN_i can begin the registration phase as it first sends the registration request to the cloud server S.

Table 3: Notations applied in e-SMDAS protocol.

Parameters	Description	Parameters	Description
λ	The security parameter	e_{SN_i}, e_{DS_j}	The ephemeral private keys of SN_i, DS_j, respectively
S	The cloud/fog server	T_1, T_2	The timestamps
SN_i	The sensor node i	h_1	One-way hash function defined from $\{0,1\}^*$ to Z_p^*
DS_j	The dew server j	h_2	One-way hash function defined from $\{0,1\}^*$ to $\{0,1\}^l$, where l is the length of session key
S	The secret key of S	h_3	One-way hash function defined from $\{0,1\}^*$ to $\{0,1\}^{2\lambda}$
ID_{SN_i}, ID_{DS_j}	The identity of SN_i, DS_j, respectively	$\mathscr{A}$	The adversary

(i) SN_i first chooses its identity ID_{SN_i} and password PW_{SN_i}. It randomly selects l_{SN_i} in Z_p^* and computes $H_1 = h_1(ID_{SN_i}\|PW_{SN_i}\|l_{SN_i})$. Finally, SN_i sends message ID_{SN_i}, H_1 to S.

(ii) After receiving ID_{SN_i}, H_1 from SN_i, S first checks if this identity has ever been registered. If it has not, then the server computes $L_{SN_i} = (sH_1)P$. After finishing computation, S sends message L_{SN_i} back to SN_i.

(iii) After getting L_{SN_i} from S, SN_i stores $\{L_{SN_i}, H_1\}$ as its long-term key securely and deletes l_{SN_i} timely.

6.2.2. Dew Server Registration Phase. Just as the sensor node registration phase, the dew server DS_j first registers in the cloud server S. DS_j operates the following steps for registration:

(i) DS_j randomly selects l_{DS_j} in Z_p^* and computes $P_{DS_j} = l_{DS_j}P$. Then, it sends its identity ID_{DS_j} and P_{DS_j} to S in a secure channel.

(ii) After receiving the message from DS_j, S first checks whether the ID_{DS_j} has been registered. If it has not, S generates the long-term key for the dew server. S computes $H_2 = h_1(ID_{DS_j}\|P_{DS_j})$, $L_{DS_j} = (sH_2)P$ and sends L_{DS_j} to DS_j.

(iii) On receiving the message from S, DS_j stores $\{L_{DS_j}, l_{DS_j}\}$ securely and publishes P_{DS_j}.

6.3. Secure Session Establishment Phase. After registering in the cloud server, both the sensor node SN_i and the dew server DS_j get their long-term keys. Then, they can establish their session key through the following steps, also illustrated in Figure 2.

(i) SN_i randomly chooses $e_{SN_i} \in Z_p^*$ and computes the corresponding public key $E_{SN_i} = e_{SN_i}P$. Then, SN_i computes $C_1 = h_1(L_{SN_i}\|ID_{SN_i})$, $A = (ID_{SN_i}\|L_{SN_i}) \oplus h_3(e_{SN_i}P_{DS_j}\|T_1)$. SN_i sends message $M_1 = A, E_{SN_i}, T_1$ to DS_j as the request for service, where T_1 is the present timestamp.

(ii) On receiving message from SN_i, DS_j first checks the freshness of the timestamp T_1. Then, it computes $E_{SN_i}l_{DS_j}$ and $ID_{SN_i}\|L_{SN_i} = A \oplus h_3(E_{SN_i}l_{DS_j}\|T_1)$. If it succeeds, DS_j can obtain $ID_{SN_i}\|L_{SN_i}$, by utilizing which it can compute C_1'. DS_j randomly selects $e_{DS_j} \in Z_p^*$ and computes $E_{DS_j} = e_{DS_j}P$, $T_{DS} = h_3(e_{DS_j}E_{SN_i}\|T_2)$ as well as the session key $SK_{DtS} = h_2(C_1'|T_{DS}|T_2)$. Finally, DS_j computes $C_2 = E_{DS_j} \oplus L_{SN_i} \oplus ID_{SN_i}$, $B = h_1(SK_{DtS}\|T_2)$ and sends message $M_2 = B, C_2, T_2$.

(iii) After receiving the message from DS_j, SN_i computes $E_{DS_j} = C_2 \oplus L_{SN_i} \oplus ID_{SN_i}$, $T_{SN} = h_3(e_{SN_i}E_{DS_j}\|T_2)$ and the session key $SK_{StD} = h_2(C_1|T_{SN}|T_2)$. Finally, it verifies whether the equality $B = h_1(SK_{StD}\|T_2)$ is right.

Hence, both the sensor node SN_i and the dew server DS_j get the same session key:

$$T_{DS} = h_3\left(e_{DS_j}E_{SN_i}\|T_2\right) = h_3\left(e_{DS_j}e_{SN_i}P\|T_2\right) = h_3\left(e_{SN_i}E_{DS_j}\|T_2\right) = T_{SN}. \tag{1}$$

In this way, if the dew server is the right potential partner, it can correctly calculate C_1'. SN_i and DS_j can obtain the same session key apparently according to the equality bellow:

$$SK_{DtS} = h_2\left(C_1'|T_{DS}|T_2\right) = h_2\left(C_1|T_{SN}|T_2\right) = SK_{StD}. \tag{2}$$

7. Security Proof

This section provides the proof of the security of e-SMDAS protocol by three methods. Firstly, we prove the proposed protocol security under the eCK security model. Then, we present a further security attribute analysis using the Scyther tool. Finally, by using BAN logic, we deduce the final security goals.

7.1. Security Theorem. We have proven the correctness of the proposed protocol above; in this subsection, we will prove the security of e-SMDAS protocol.

Theorem 1. *Let $\mathscr{A}$ be a probabilistic polynomial time adversary against the proposed protocol Π with a time bound t, making at most q_s. Send queries q_{h_1}, q_{h_2}, q_{h_3} random oracle queries. Then,*

$$\mathrm{Adv}_{\Pi}(\mathscr{A}) \le \frac{q_s}{2^{\lambda-2}} + \frac{q_s}{2^{2\lambda-2}} + \frac{2^{\lambda}\cdot q_{h_1}^2 + q_{h_3}^2}{2^{2\lambda}} + \frac{q_{h_2}^2}{2^l} + 2q_{h_2}q_s^2\mathrm{Adv}^{\mathrm{ECCDH}}(\mathscr{S}), \tag{3}$$

where $Adv^{ECCDH}(\mathscr{S})$ means the success probability of solving an instance of ECCDH problem by an algorithm $\mathscr{S}$.

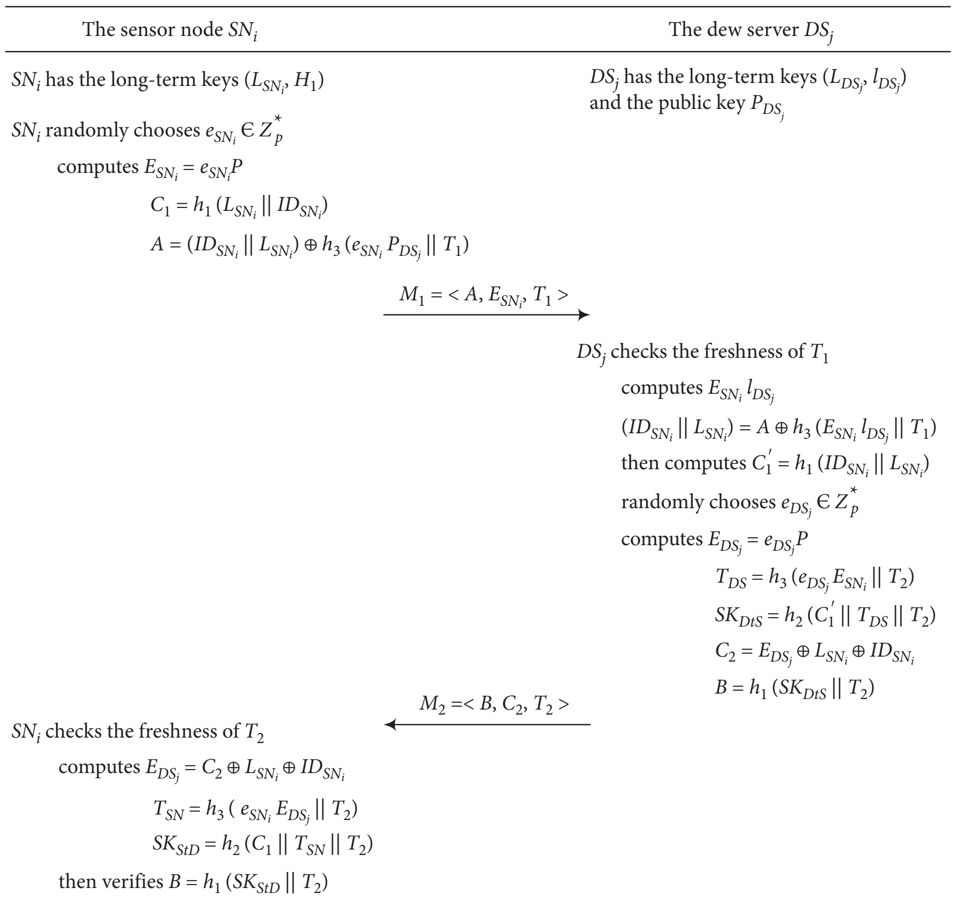

Figure 2: Secure session establishment phase of e-SMDAS protocol.

Proof of Theorem 1: Next, we will prove the security of the proposed protocol through defining a sequence of hybrid experiments where $\mathscr{A}$ correctly guesses the random bit b in the Test query. Specifically, each experiment has a definition of Succ_i to illustrate the advantage.

(i) **Experiment 0**: This experiment simulates the situation of the attacks against the real protocols in the random oracle model. According to the definition, there exists $\mathrm{Adv}(\mathscr{A}) = 2\Pr[\mathrm{Succ}_0] - 1$, which means the origin advantage of adversary.

(ii) **Experiment 1**: In this experiment, $\mathscr{S}$ simulates the random oracles h_1, h_2, and h_3 by keeping hash lists L_{h_1}, L_{h_2}, L_{h_3} as follows:

   (i) If there exists a record of message M as (M, H) in the list L_{h_1}, it returns H. Otherwise, it selects an element H, adds the record (M, H) to the list L_{h_1}, and then returns H.

   (ii) If there exists a record of message M as (M, K) in the list L_{h_2}, it returns K. Otherwise, it selects an element K in the key set, adds the record (M, K) to the list L_{h_2}, and then returns K.

   (iii) If there exists a record of message M as (M, J) in the list L_{h_3}, it returns J. Otherwise, it selects an element J in the key set, adds the record (M, J) to the list L_{h_3}, and then returns J.

The Send, Reveal, Longterm, Ephemeral, and Test queries are also simulated as the real attack. Thus, this experiment is same as the real experiment, which means that the equation $\Pr[\mathrm{Succ}_1] = \Pr[\mathrm{Succ}_0]$ holds.

(i) **Experiment 2**: In this experiment, we simulate all oracles the same as **Experiment 1** except that a collision occurs in the output of the oracle h_1 or the session transcripts. According to the birthday paradox, the probability of collisions in the output of the oracle h_1 is at most $q_{h_1}^2/2^{\lambda+1}$, where q_{h_1} is the maximum times of queries to h_1. The same deduction can be applied to h_2 and h_3. Therefore, the successful probability of **Experiment 2** satisfies $\Pr[\mathrm{Succ}_2] - \Pr[\mathrm{Succ}_1] \leq q_{h_1}^2/2^{\lambda+1} + q_{h_3}^2/2^{2\cdot\lambda+1} + q_{h_2}^2/2^{l+1}$.

(ii) **Experiment 3**: In this experiment, the protocol will not halt except that $\mathscr{A}$ successfully guesses C_1 or T_{DS} (T_{SN}) without querying h_1 or h_3. Therefore, there exists $\Pr[\mathrm{Succ}_3] - \Pr[\mathrm{Succ}_2] \leq 2 \cdot q_s/2^{\lambda} + 2 \cdot q_s/2^{2\cdot\lambda}$.

(iii) **Experiment 4**: In this experiment, we only consider the situation where $\mathscr{A}$ exactly chooses a random session as the test session. Besides, the computation of the test session key is modified to select a random key from the key set. Consequently, the difference between **Experiment 3** and **Experiment 4** is in the event when $\mathcal{A}$ queries the tuple $(C_1|T_{SN}|T_2)$ or $(C_1|T_{DS}|T_2)$ to h_2 in the test session. To describe this difference, the following four cases may be considered:

(i) Longterm(SN_i) and Longterm(DS_j) are queried, from which $\mathcal{A}$ can obtain the long-term key L_{SN_i} of SN_i and l_{DS_j}, L_{DS_j} of DS_j. To calculate the session key, either e_{SN_i} or e_{DS_j} is required.

(ii) Longterm(SN_i) and Ephemeral(DS_j) are queried, from which $\mathcal{A}$ can obtain the long-term key L_{SN_i} of SN_i and e_{DS_j} of DS_j. To calculate the session key, e_{SN_i} is required.

(iii) Ephemeral(SN_i) and Longterm(DS_j) are queried, from which $\mathcal{A}$ can obtain the long-term key e_{SN_i} of SN_i and l_{DS_j}, L_{DS_j} of DS_j. To calculate the session key, e_{DS_j} is required.

(iv) Ephemeral(SN_i) and Ephemeral(DS_j) are queried, from which $\mathcal{A}$ can obtain the long-term key e_{SN_i} of SN_i and e_{DS_j} of DS_j. To calculate the session key, L_{SN_i} and l_{DS_j} are required.

If any of these four cases happens, then referring to the method proposed in [26], we can construct an algorithm $\mathcal{S}$ to solve an instance of ECCDH problem, and there exists

$$\Pr[\mathrm{Succ}_4] - \Pr[\mathrm{Succ}_3] \leq q_{h_2} q_s^2 \mathrm{Adv}^{\mathrm{ECCDH}}(\mathcal{S}). \quad (4)$$

Besides, in **Experiment 4**, to guess the bit b in the Test query is random, and other sessions do not matter. Therefore, there exists $\Pr[\mathrm{Succ}_4] = 1/2$. □

7.2. Scyther Security Analysis. Besides proving the security of the proposed model formally, we also use Scyther tool to show the proposed protocol is secure against various attacks. The setting used is presented in Figure 3 to achieve highly strong security, including perfect forward security, resistance to session key reveal attack, and resistance to ephemeral key leakage attack.

The result of analysis is demonstrated in Figure 4. According to Figure 4, we can clearly infer that under the setting predefined, the session key is secure against various attacks.

7.3. BAN Logic Formalized Security Proof. In this subsection, we provide another method to analyze the security of e-SMDAS protocol.

Next, we will prove that the proposed protocol can achieve the mutual authentication and two participants can obtain the same session key. We first present the security goals using BAN logic followed. We simplify the sensor node SN_i as N, and the dew server DS_j as D.

(i) $\mathbf{G}_1 N$ believes $(N \overset{SK}{\leftrightarrow} D)$.
(ii) $\mathbf{G}_2 D$ believes $(N \overset{SK}{\leftrightarrow} D)$.
(iii) $\mathbf{G}_3 N$ believes $(D$ believes $(N \overset{SK}{\leftrightarrow} D))$.
(iv) $\mathbf{G}_4 D$ believes $(N$ believes $(N \overset{SK}{\leftrightarrow} D))$.

Then, we formalize the original messages into the idealized ones as follows:

(i) $\mathbf{M}_1 N \longrightarrow D$: $\{N, L_N, T_1\}_{e_N \cdot P_D}$, $\overset{K_2}{\longrightarrow} N, T_1$.
(ii) $\mathbf{M}_2 D \longrightarrow N$: $T_{2K_{ND}}$, $\overset{K_2}{\longrightarrow} D_{L_N}, T_2$.

Thirdly, we make the initial assumptions.

(i) $\mathbf{A}_1 D$ believes (fresh (T_1)).
(ii) $\mathbf{A}_2 N$ believes (fresh (T_2)).
(iii) $\mathbf{A}_3 D$ believes $(N \overset{E_N}{\Leftrightarrow} E_N D)$.
(iv) $\mathbf{A}_4 N$ believes $(D \overset{E_D}{\Leftrightarrow} E_D N)$.
(v) $\mathbf{A}_5 D$ believes $(N$ controls $(N \overset{K_{ND}}{\leftrightarrow} D))$.
(vi) $\mathbf{A}_6 N$ believes $(D$ controls $(N \overset{K_{ND}}{\leftrightarrow} D))$.

Finally, following the idealized messages, we utilize the predefined notations, rules, and assumptions to deduce the goals of the proposed protocol. The proof process is presented as follows:

(i) From $\mathbf{M}_1$, we can derive the formula $\mathbf{F}_1$ as follows:

(1) $\mathbf{F}_1 D$ sees $(\{N, L_N, T_1\}_{e_N \cdot P_D}, \overset{K_2}{\longrightarrow} N, T_1)$.

(ii) According to $\mathbf{R}_4$ and $\mathbf{F}_1$, we can deduce the formula $\mathbf{F}_2 \sim \mathbf{F}_4$ as follows:

(1) $\mathbf{F}_3 D$ sees (T_1).
(2) $\mathbf{F}_4 D$ sees $(\overset{K_2}{\longrightarrow} N)$.

(iii) According to $\mathbf{R}_1$, $\mathbf{A}_3$, and $\mathbf{F}_2$, we can deduce the formula $\mathbf{F}_5$ as follows:

(1) $\mathbf{F}_5 D$ believes $(N$ said $(N, K_N, T_1))$.

(iv) According to $\mathbf{F}_5$, $\mathbf{A}_1$, and $\mathbf{R}_2$, we can deduce the formula $\mathbf{F}_6 \sim \mathbf{F}_8$ as follows:

(1) $\mathbf{F}_6 D$ believes $(N$ believes $(N, K_N, T_1))$.
(2) $\mathbf{F}_7 D$ believes $(N$ believes $(N, K_N))$.
(3) $\mathbf{F}_8 D$ believes $(N$ believes $(C_1))$.

(v) From $\mathbf{M}_2$, we can derive the formula $\mathbf{F}_9$ below:

(1) $\mathbf{F}_9 N$ sees $(T_{2_{K_{ND}}}, \overset{K_1}{\longrightarrow} N, N_{E_D}, T_2)$.

(vi) According to $\mathbf{R}_4$, $\mathbf{A}_2$ and $\mathbf{F}_9$, we can deduce the formula $\mathbf{F}_{10}$, $\mathbf{F}_{11}$, and $\mathbf{F}_{12}$:

(1) $\mathbf{F}_{10} N$ sees $(T_{2K_{ND}})$.
(2) $\mathbf{F}_{11} N$ sees $(\overset{K_1}{\longrightarrow} N, N_{E_D})$.
(3) $\mathbf{F}_{12} N$ believes (fresh $0 \overset{K_1}{\longrightarrow} N, N_{E_D}$).

(vii) According to $\mathbf{F}_{11}$, $\mathbf{A}_4$, and $\mathbf{R}_1$, we can deduce the formula $\mathbf{F}_{13}$:

(1) $\mathbf{F}_{14} N$ believes $(D$ said $(\overset{K_1}{\longrightarrow} N, N))$.

(viii) According to $\mathbf{F}_{13}$, $\mathbf{F}_{12}$, and $\mathbf{R}_2$, we can deduce the formula $\mathbf{F}_{14} \sim \mathbf{F}_{15}$:

(1) $\mathbf{F}_{14} N$ believes $(D$ believes $(L_N, N))$.
(2) $\mathbf{F}_{15} N$ believes $(D$ believes $(C_1))$.

(ix) Since $K_{ND} = h_2(C_1 || h_3(e_N e_D P \| T_2) || T_2)$, we can deduce the formula $\mathbf{F}_{16}$ according to $\mathbf{F}_{15}$, $\mathbf{A}_2$, and $\mathbf{A}_4$, which is also $\mathbf{G}_3$:

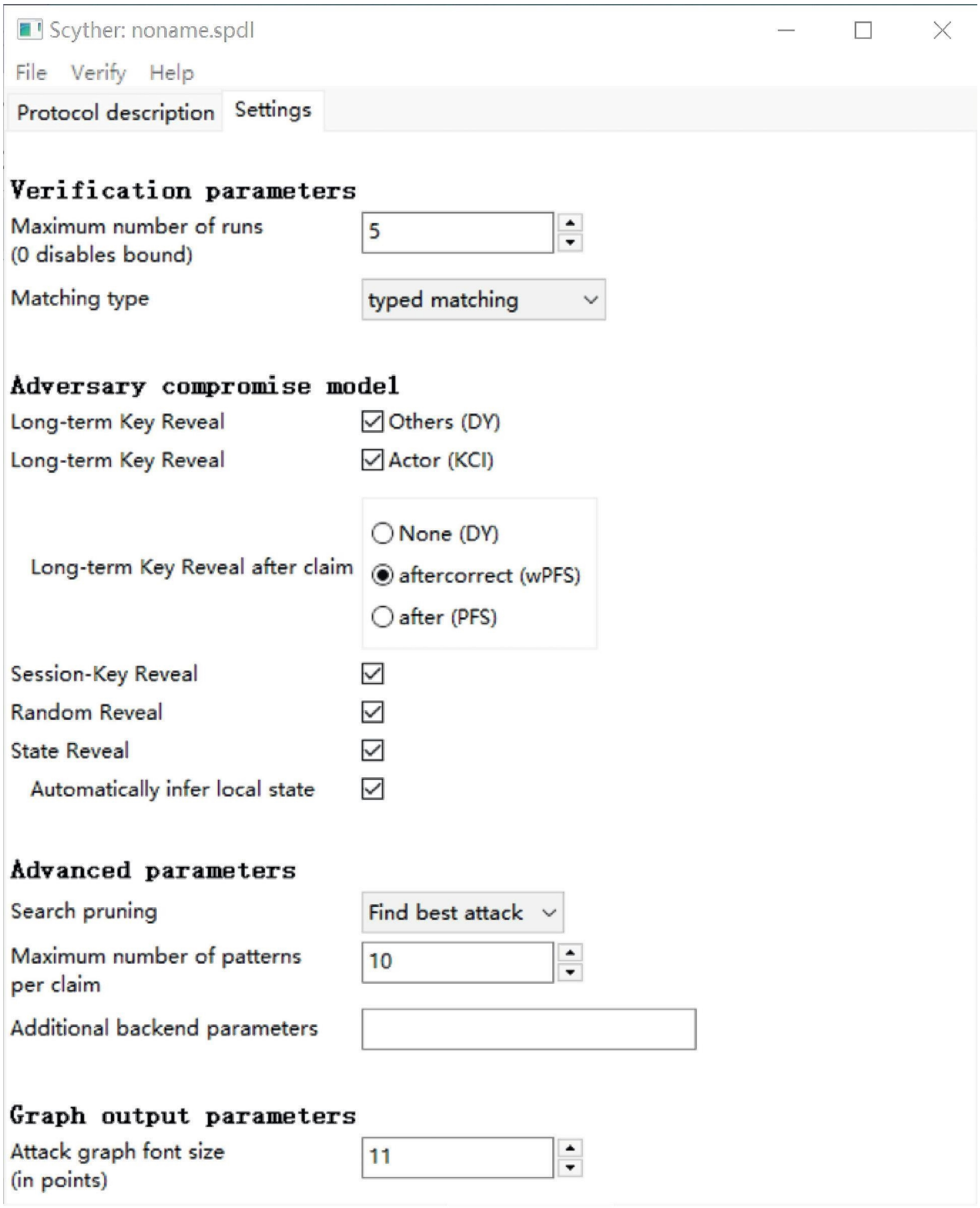

FIGURE 3: The setting of Scyther.

(1) $\mathbf{F}_{16} N$ believes $(D$ believes $(N \overset{SK}{\leftrightarrow} D))$.

(x) Since $K_{ND} = h_2(C_1 h_3 \| (e_N e_D P \| T_2) \| T_2)$, we can deduce the formula $\mathbf{F}_{17}$ according to $\mathbf{F}_8$, $\mathbf{A}_1$, and $\mathbf{A}_3$, which is also $\mathbf{G}_4$:

(1) $\mathbf{F}_{17} D$ believes $(N$ believes $(N \overset{SK}{\leftrightarrow} D))$.

(xi) According to $\mathbf{F}_{16}$, $\mathbf{A}_5$, and $\mathbf{R}_3$, we deduce the formula $\mathbf{F}_{18}$, which is also $\mathbf{G}_1$:

(1) $\mathbf{F}_{18} N$ believes $(N \overset{SK}{\leftrightarrow} D)$.

(xii) Similarly, according to $\mathbf{F}_{17}$, $\mathbf{A}_6$, and $\mathbf{R}_3$, we deduce the formula $\mathbf{F}_{19}$, which is also $\mathbf{G}_2$:

(1) $\mathbf{F}_{19} D$ believes $(N \overset{SK}{\leftrightarrow} D)$.

According to $\mathbf{F}_{16}$ to $\mathbf{F}_{19}$, the secure goals $\mathbf{G}_1$ to $\mathbf{G}_4$ of e-SMDAS protocol are achieved. The sensor node SN_i and the dew server DS_j can achieve the mutual authentication and the same session key securely.

8. Performance Analysis

In this section, we present the performance analysis of e-SMDAS protocol, compared with several recent works, namely, SMDAS [10], He et al.'s scheme [27], and Ying et

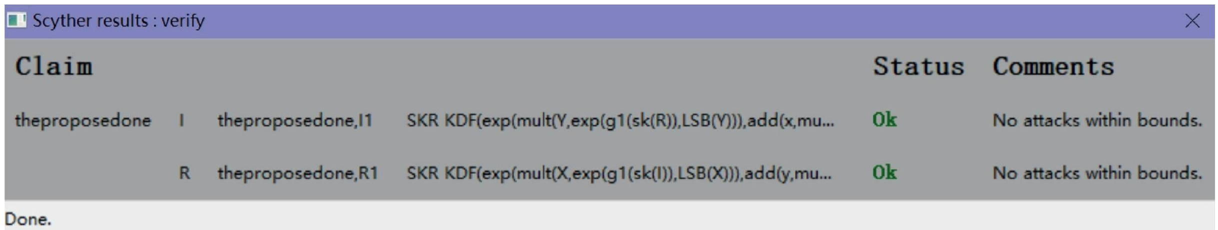

Figure 4: The analysis result by Scyther tool.

Table 4: Comparison of security features with SMDAS protocol.

Security features	Scheme			
	[27]	[28]	SMDAS	e-SMDAS
Mutual authentication	No	Yes	Yes	Yes
Session key agreement	Yes	Yes	Yes	Yes
Replay attack resistance	Yes	No	Yes	Yes
User impersonation attack resistance	No	No	Yes	Yes
User anonymity	No	Yes	No	Yes
Forward security	Yes	Yes	No	Yes

Table 5: Comparison of the communication efficiency.

Scheme	Communication efficiency	
	Computation cost	Communication cost (bits)
SMDAS	$10T_h + T_b + T_{pa}$	1184
He et al. [27]	$4T_e + 5T_{pa} + 2T_b + 5T_h$	3296
Ying et al. [28]	$7T_e + 10T_h + 4T_{enc/dec}$	3840
e-SMDAS	$10T_h + 2T_e$	1024

Note. T_{pa}: point addition in elliptic curve group; T_e: exponentiation operation in cyclic group; T_h: hash function; T_b: bilinear map; T_{eb}: exponentiation operation over bilinear pairing; T_{ma}: modular addition in cyclic group; $T_{enc/dec}$: encryption/decryption operation.

al.'s scheme [28], from the aspects of security features and computational efficiency.

Table 4 demonstrates the result of security feature comparison with several similar works. According to the work of [29, 30], the comparative result of security features is clear. It is shown in the table that the proposed protocol remedies the flaws of SMDAS protocol. As Table 4 shows, the e-SMDAS protocol can resist replay attack as well as user impersonation attack and satisfy the secure requirements for anonymity and forward security. Generally, our e-SMDAS protocol performs better than the previous one.

Before presenting the analysis, we first denote the notations used in the estimation of the computation efficiency. To be concise, the meanings of T_{pa}, T_e, T_h, T_b, T_{eb}, T_{ma}, and $T_{enc/dec}$ are time of performing a point addition in elliptic curve group, time of performing an exponentiation operation in cyclic group, time of performing a hash function, time of performing a bilinear map, time of performing an exponentiation over bilinear pairing, time of performing a modular addition in cyclic group, and time of performing an encryption or decryption operation, respectively. Besides, the time of XOR can be negligible. In Table 5, we compare the computation efficiency of the related works with that of ours. For the sensor node in the proposed protocol, the computation cost is $5T_h + T_e$. On the other hand, dew server operates at the cost of $5T_h + T_e$.

To compare the efficiency of communication precisely, we simulate the schemes under the following assumptions. The output length of hash function is 160 bits while that of symmetric encryption tool is 1024 bits. The size of timestamp is 32 bits, while the output of elliptic curve is 160 bit. The comparative result is demonstrated in both Table 5 and Figure 5, in which e-SMDAS appears to be more efficient.

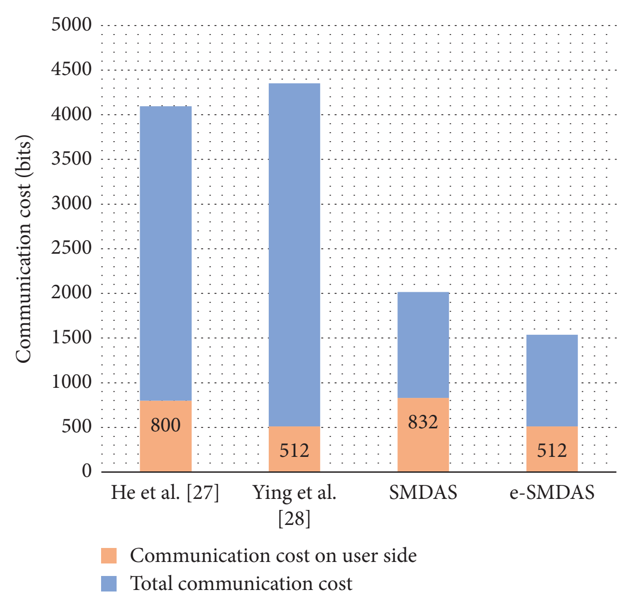

Figure 5: The comparison of the communication cost.

9. Conclusion

The dew-assisted IoT framework is an essential approach developing rapidly in the communication systems, which can provide high efficiency and low latency. In this paper, we first analyze SMDAS protocol showing that this protocol lacks forward security and user anonymity. Then, based on ECCDH problem, we propose an enhancement of the original one, called e-SMDAS protocol. We present the formal security proof of the proposed protocol. Moreover, the test of security by the usage of formalization tool Scyther and BAN logic shows that e-SMDAS can satisfy more security features than the former protocol. Furthermore, the performance analysis is presented at last showing that the enhancement does not affect the running time and computation efficiency.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant no. 61872449).

References

[1] B. Flavio, M. Rodolfo, Z. Jiang, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the conference: The First Edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13–16, dblp, New York, NY USA, 17 August 2012.

[2] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[3] Y. Wang, "Cloud-dew architecture," *International Journal of Cloud Computing*, vol. 4, no. 3, pp. 199–210, 2015.

[4] Y. Wang, "Definition and categorization of dew computing," *Open Journal of Cloud Computing*, vol. 3, no. 1, pp. 1–7, 2016.

[5] K. Hameed, S. Garg, M. B. Amin, and B. Kang, "A formally verified blockchain-based decentralised authentication scheme for the internet of things," *The Journal of Supercomputing*, vol. 77, no. 12, pp. 14461–14501, 2021.

[6] Y. Liu, J. Zhang, and J. Zhan, "Privacy protection for fog computing and the internet of things data based on blockchain," *Cluster Computing*, vol. 24, no. 2, pp. 1331–1345, 2021.

[7] S. Shukla, S. Thakur, S. Hussain, J. G. Breslin, and S. M. Jameel, "Identification and authentication in healthcare Internet-of-Things using integrated fog computing based blockchain model," *Internet of things*, vol. 15, Article ID 100422, 2021.

[8] S. Singh and V. K. Chaurasiya, "Mutual authentication scheme of IoT devices in fog computing environment," *Cluster Computing*, vol. 24, no. 3, pp. 1643–1657, 2021.

[9] S. Amanlou, M. K. Hasan, and K. A. A. Bakar, "Lightweight and secure authentication scheme for IoT network based on publish-subscribe fog computing model," *Computer Networks*, vol. 199, no. 2021, Article ID 108465, 2021.

[10] R. Saurabh, S. O. Mohammad, M. Dheerendra, A. Mishra, and Y. S Rao, "Efficient design of an authenticated key agreement protocol for dew-assisted IoT systems," *The Journal of Supercomputing*, vol. 78, pp. 3696–3714, 2022.

[11] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018.

[12] R. Amin and G. P. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, vol. 36, pp. 58–80, 2016.

[13] A. K. Awasthi and K. Srivastava, "A biometric authentication scheme for telecare medicine information systems with nonce," *Journal of Medical Systems*, vol. 37, no. 5, p. 9964, 2013.

[14] Z. Tan, "A user anonymity preserving three-factor authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 3, p. 16, 2014.

[15] H. Arshad and M. Nikooghadam, "Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 12, p. 136, 2014.

[16] Y. Lu, L. Li, H. Peng, and Y. Yang, "An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem," *Journal of Medical Systems*, vol. 39, no. 3, p. 32, 2015.

[17] L. Han, Q. Xie, and W. Liu, "An improved biometric based authentication scheme with user anonymity using elliptic curve cryptosystem," *International Journal on Network Security*, vol. 19, no. 3, pp. 469–478, 2017.

[18] V. P. Gaikwad, J. V. Tembhurne, C. Meshram, and C.-C. Lee, "Provably secure lightweight client authentication scheme with anonymity for TMIS using chaotic hash function," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 8281–8304, 2021.

[19] M. Masud, G. S. Gaba, S. Alqahtani et al., "A lightweight and robust secure key establishment protocol for internet of medical things in COVID-19 patients care," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15694–15703, 2021.

[20] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.

[21] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, "Achieving graph clustering privacy preservation based on structure entropy in social IoT," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2761–2777, 2022.

[22] S. A. Chaudhry, M. S. Farash, H. Naqvi, S. Kumari, and M. K. Khan, "An enhanced privacy preserving remote user authentication scheme with provable security," *Security and Communication Networks*, vol. 8, no. 18, pp. 3782–3795, 2015.

[23] N. Ravanbakhsh, M. Mohammadi, and M. Nikooghadam, "Perfect forward secrecy in VoIP networks through design a lightweight and secure authenticated communication scheme," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 11129–11153, 2019.

[24] M. Nikooghadam and H. Amintoosi, "Perfect forward secrecy via an ECC-based authentication scheme for SIP in VoIP," *The Journal of Supercomputing*, vol. 76, no. 4, pp. 3086–3104, 2020.

[25] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Proceedings of the conference: ProvSec2007 - Provable Security*, pp. 1–16, Springer, Wollongong, Australia, 1 November 2007.

[26] H. Krawczyk, "HMQV: a high-performance secure Diffie-Hellman protocol," *Advances in Cryptology - CRYPTO 2005*, vol. 3621, pp. 546–566, 2005.

[27] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud

computing services," *IEEE Systems Journal*, vol. 12, pp. 1621–1631, 2016.

[28] B. Ying and A. Nayak, "Efficient authentication protocol for secure vehicle communications," in *Proceedings of the conference: 2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pp. 1–5, IEEE, Seoul, Korea (South), 18 May 2014.

[29] C. Chen, B. Xiang, Y. Liu, and K. Wang, "A secure authentication protocol for internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[30] S. Nagaraju, S. K. V. Jayakumar, and C. S. Priya, "An effective mutual authentication scheme for provisioning reliable cloud computing services," in *Proceedings of the conference: 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 314–321, IEEE, Greater Noida, India, 19 February 2021.