Transient Session Key Derivation Protocol for Key Escrow Prevention in Public Key Infrastructure

Vincent Omollo Nyangaresi[1], Zaid Ameen Abduljabbar[2,3], Ismail Yaqub Maolood[4], Mustafa A. Al Sibahee[5,6], Junchao Ma[5(✉)], and Abdulla J. Y. Aldarwish[2]

[1] Faculty of Biological and Physical Sciences, Tom Mboya University, 40300 Homabay, Kenya
vnyangaresi@tmuc.ac.ke
[2] Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq
{zaid.ameen,abdullajas}@uobasrah.edu.iq
[3] Shenzhen Institute, Huazhong University of Science and Technology, Shenzhen 518118, China
[4] Department of Information and Communication Technology Center (ICTC) – System Information, Ministry of Higher Education and Scientific Research, Erbil, Iraq
ismail.maulood@mhe-krg.org
[5] College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China
{mustafa,majunchao}@sztu.edu.cn, mustafa.alsibahee@iuc.edu.iq
[6] Computer Technology Engineering Department, Iraq University College, Basrah, Iraq

Abstract. The Internet of Things (IoT) devices have been deployed to realize smart environments such as smart cities, smart homes, smart health and smart grids. In these domains, the IoT devices collect and forward high volumes of sensitive and private data. It is therefore important that security schemes be developed to protect the exchanged data. In this regard, a myriad of authentication protocols have been developed over the recent past. However, these schemes deploy cryptographic primitives that result in extremely high communication, storage and computation complexities. In addition, some of these protocols still have numerous security and privacy issues that render them unsuitable for deployment in an IoT environment. For instance, untraceability, anonymity, key escrow problems and attack vectors such as de-synchronization and forgery attacks are frequently ignored. In this paper, a transient session key derivation protocol is developed to address some of these security and efficiency challenges. The security analysis executed shows that this protocol offers untraceability, device anonymity and perfect forward key secrecy. In addition, it is robust against de-synchronization, known secret key leakage, eavesdropping and forgery attacks. In terms of operational efficiency, this protocol incurs the lowest computation and communication complexities.

Keywords: Authentication · De-synchronization · Key escrow · Security · Privacy

T. Pereira et al. (Eds.): IoECon 2022, LNICST 458, pp. 103–116, 2023.
https://doi.org/10.1007/978-3-031-25222-8_9

1 Introduction

In typical cyber physical systems, a number of smart devices are interconnected to form an Internet of Things (IoT). These smart devices collect a myriad of data from their surroundings and then forward the same to some central data servers where processing is accomplished. After some manipulations, transformations and analysis, the collected data can facilitate various activities [1]. For instance, IoT can be amalgamated with cloud computing to realize smart cities [2]. Apart from smart cities, IoT deployments have been utilized to improve healthcare, transportation, safety, efficiency and convenience [3]. Over the recent past, Device-to-Device (D2D) communication has been developed, in which IoT devices exchange messages directly amongst themselves. This helps to offload the servers [4] in addition to boosting reliability and transmission rates. As such, the IoT devices can still communicate even when the central cloud has failed.

Although D2D comes with many benefits, security and privacy of the devices as well as data present some challenges [5, 6]. This is attributed to the numerous vulnerabilities and risks occasioned by device heterogeneity. Consequently, the architectures and protocols utilized in these smart devices are different and incompatible. Since privacy and security implementations in these devices differ, interoperability is also difficult. Therefore, a single privacy and security technique may not be satisfy the requirements of all devices and applications. Ultimately, upholding security and privacy of exchanged data as well as data at rest presents some challenges [7]. Considering smart health IoT systems, sensitive and private information is being generated and transmitted. Compared with wired communication systems, wireless transmissions in IoT is susceptible to many threats and attacks such as eavesdropping, forgery and modifications. Although the interconnectivity among numerous devices brings forth user convenience, this inadvertently increases the attack surfaces. As such, security and privacy among the interconnected devices has become a significant issue [8].

In wireless networks, authentication is normally the first step towards security and privacy preservation [2, 9]. Indeed, the authors in [10] explain that secure authentication protocols help preserve privacy of the users as they interact with their smart devices. Unfortunately, the resource-constrained nature of most of the smart devices renders it difficult to implement strong security protocols. In this regard, this paper makes the following contributions:

- A protocol that directly authenticates IoT devices is developed, devoid of the involvement of any central controller. This potentially solves the key escrow problems in majority of the current authentication protocols.
- Random nonces and pseudonyms are incorporated in the derivation of security tokens. This serves to preserve user anonymity and thwart any traceability attacks.
- The exchanged messages are enciphered through symmetric keys to protect them against eavesdropping and modifications. Effectively, this upholds both confidentiality and integrity of the communication process.
- Extensive security analysis is carried out to show the robustness of the proposed protocol against conventional IoT attack vectors.
- Comparative performance evaluation is executed to demonstrate that the developed protocol incurs the least computation and communication costs.

The rest of this paper is structured as follows: Sect. 2 gives a presentation of the related work while Sect. 3 discusses the employed system model. On the other hand, Sect. 4 presents the results obtained and discusses them. Towards the end of this paper, Sect. 5 gives the conclusions of this paper and offers some future research directions.

2 Related Work

The proliferation of attacks in IoT environment has seen numerous security protocols being developed to address these issues. Although majority of these schemes have fairly addressed security and privacy issues, majority of them have high computation and communication overheads [11, 12]. In addition, some security challenges still exist in majority of these protocols. For instance, the protocol in [13] has security issues as discussed by [14]. In addition, this scheme is vulnerable to Denial of Service (DoS) attacks and is unable to detect incorrect password login attempts. To curb these issues, lightweight and provably secure protocols have been introduced in [15] and [16]. To prevent black-hole attacks, trust-based and two-tier based protocols have been developed in [11] and [17] respectively. However, these two schemes solve privacy and authenticity issues at the expense of increased complexities. To boost security without compromising performance, Elliptic Curve Cryptography (ECC) based security approaches are presented in [18] and [19]. On the other hand, a blind anonymous authentication scheme is introduced in [20] too prevent eavesdropping attacks. However, since these three protocols are basically identity-based, they face key escrow challenges [21]. Although an efficient and secure scheme developed in [22] may address this problem, it inadvertently compromises user privacy during identity authentication. Therefore, a scheme in [23] has been developed to effectively deal with this challenge through biometrics based authentication.

De-synchronization attacks are other serious security issues that require immediate solution. For instance, although the protocol in [24] offers mutual authentication, it cannot withstand de-synchronization attacks. Therefore, the protocol in [25] has been designed to prevent these attacks. On the other hand, blockchain technology has also been deployed to offer security in IoT environment. For example, based on blockchain technology, a security preserving scheme is presented in [25]. Unfortunately, this technique has high complexity and communication overheads [26]. Similarly, the blockchain based privacy preserving protocol in [27] has high computation and communication overheads. On the other hand, the ECC and pseudonym based authentication protocol in [28] is lightweight and hence can address performance issues in both [27] and [29]. Similarly, the rating-based authentication protocol in [30] has lower computation costs and hence can address the performance issues in [27] and [29]. Using hash chains and blind signatures, an anonymous authentication scheme is developed in [31]. Although this scheme upholds sender privacy, it cannot trace malicious senders within the network. Similarly, a blind group signature based anonymous authentication protocol is presented in [32]. Although this scheme offers conditional anonymity, it incurs high communication and computation costs.

On its part, the protocol in [33] cannot withstand impersonation and key compromise attacks. In addition, it does not uphold user anonymity. Although the bilinear pairing

based protocol in [34] offers mutual authentication, these pairing operations are time consuming [35]. Similarly, the scheme in [17] offers protection against black-hole attacks at the expense of very high complexity and communication costs. On the other hand, due to the centralized registration for secret cookie data creation, the scheme in [36] presents a single point of failure. To prevent exposure attacks, a key agreement protocol is introduced in [37]. However, a Physically Unclonable Functions (PUF) based scheme is introduced in [38] to thwart physical attacks. Unfortunately, PUFs challenge-response pairs are sometimes inconsistent and hence PUF based schemes have stability issues [39, 40]. According to [7], many privacy preserving protocols have been developed for IoT devices. However, most of these protocols have high computation complexities in addition to numerous security weaknesses.

3 System Model

The communicating entities involved in the proposed protocol include two IoT devices, which are labeled DV_1 and DV_2 as shown in Fig. 1.

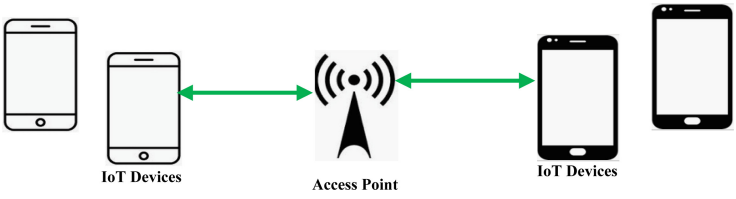

Fig. 1. Network model

As shown in Fig. 1, the IoT devices in this network model communicate directly through a wireless access point. These devices have the same specifications and can therefore be regarded as peers. Table 1 presents the notations used in this protocol.

The three phases that characterize the proposed protocol include the handshake, authentication and the key agreement. The sub-sections below describes these phases in some greater details.

3.1 Handshake Phase

Before the two devices can start exchanging packets, they engage in a handshake in which they generate initial security tokens. This is a 3 step process as described below.

Step 1: DV1 stochastically generates its identity ID1 and some high entropy random nonce R1 which are then forwarded to DV2 in message {ID1, R1} over some secure channels.

Step 2: On receiving message {ID1, R1}, DV2 randomly generates its identity ID2 before deploying its master key MSK to generate cipher $C1 = EMSK((ID1 \oplus h(ID2||MSK))||(R1 \oplus ID1))$. Next, it derives its pseudo-identity $PID = h((ID1|| ID2) \oplus MSK \oplus R1)$ which it stores in its memory. Lastly, it transmits {C1} to DV1 over secure channels.

Table 1. Notations

Notation	Description
R_i	Random number *i*
DV_1 & DV_2	Device 1 and device 2
MSK	DV_2 master key
E_k	Encipher using key *k*
ID_1 & ID_2	Unique identity of DV_1 & DV_2
PID	DV_2 pseudo-identity
D_k	Decipher using key *k*
SK	Symmetric key
S_{SK}	Session key
h (.)	One-way hashing function
$\|\|$	Concatenation operation
$\oplus$	XOR operation

Step 3: After getting {C1} from DV2, DV1 stores parameter set {C1, ID1, R1} in its memory.

3.2 Authentication and Key Negotiation Phase

In this phase, the two devices utiize the security tokens obtained during the handshake phase to verify the authenticity of each other. This is accomplished in 7 steps that are elaborated below.

Step 1: Device DV_1 chooses high entropy random nonce R_2 before deriving security parameter $C_2 = R_2 \oplus h(ID_1 \| R_1)$. Next, DV_1 constructs message $M_1 = \{C_1, C_2\}$ and sends it over to DV_2 over public channels.
Step 2: Upon getting message M_1, DV_2 decrypts C_1 using its master key MSK by executing $D_1 = D_{MSK}(C_1)$ so as to obtain ${ID_1}^* \oplus h({ID_2}^* \| MSK^*)$ and ${R_1}^* \oplus {ID_1}^*$. Next, it uses the decrypted message ${ID_1}^* \oplus h({ID_2}^* \| MSK^*)$, its own identity ID_2 and master key MSK to compute device DV_1 identity, ${ID_1}^{New} = {ID_1}^* \oplus h({ID_2}^* \| MSK^*) \oplus h(ID_2 \| MSK)$. Upon successful derivation of DV_1's identity, DV_2 proceeds to compute ${R_1}^{New} = {R_1}^* \oplus {ID_1}^* \oplus {ID_1}^{New}$. Finally, DV_2 computes $PID^* = h((ID_1^{New} \| ID_2) \oplus MSK \oplus R_1^{New})$. It then checks if $PID^* \stackrel{?}{=} PID$ such that the authentication request is rejected if the two values are not identical. Otherwise, DV_2 shifts to other subsequent steps.
Step 3: Using the received C_2 in message M_1, DV_2 computes ${R_2}^* = C_2 \oplus h(ID_1^{New} \| R_1^{New})$. This is followed by the updating of C_1 with ${C_1}^*$ by substituting random nonce ${R_1}^*$ with ${R_2}^*$ in ${C_1}^* = E_{MSK}((ID_1^{New} \oplus h(ID_2 \| MSK)) \| ({R_2}^* \oplus ID_1^{New}))$.
Step 4: Device DV_2 chooses high entropy random nonce R_3 and derives a new symmetric key $SK = h(ID_1^{New} \oplus R_1^{New} \oplus {R_2}^*)$. Using the just computed encryption key, DV_2 generates authentication message $AM_1 = E_{SK}$

$((h((ID_1{}^{New} \oplus R_2{}^*)||R_1{}^{New}) \oplus R_3)||h(ID_1{}^{New}||R_1{}^{New}||R_2{}^*)||C_1{}^*)$. Lastly, DV_2 derive session key $S_{SK} = h(ID_1{}^{New}||R_1{}^{New}||R_2{}^*||R_3)$ that is sent in message $M_2 = \{S_{SK}\}$ over to device DV_1 using public channels.

Step 5: On receiving message M_2, device DV_1 deploys its identity ID_1 and high entropy random nonces R_1 and R_2 to derive symmetric key $SK^* = h(ID_1 \oplus R_1 \oplus R_2)$. It then utilizes this key to decrypt message AM_1 as $D_2 = D_{SK*}(AM_1)$ to obtain $h((ID_1{}^{New*} \oplus R_2{}^{New})||R_1{}^{New*}) \oplus R_3{}^*)$, $h(ID_1{}^{New*}||R_1{}^{New*}||R_2{}^{New})$, and $C_1{}^{New}$. Next, DV_1 checks if the decrypted value $h(ID_1{}^{New*}||R_1{}^{New*}||R_2{}^{New})$ is equivalent to the computed value $h(ID_1||R_1||R_2)$. On condition that the two values are dissimilar, the authentication session is terminated. Otherwise, DV_1 computes $R_3{}^{New} = h((ID_1{}^{New*} \oplus R_2{}^{New})||R_1{}^{New*}) \oplus R_3{}^* \oplus h(ID_1 \oplus R_2)||R_1)$ that it uses to derive session key $S_{SK}{}^* = h(ID_1||R_1||R_2||R_3{}^{New})$. Lastly, DV_1 computes authentication message $AM_2 = h(S_{SK}{}^*||R_3{}^{New})$ that is then sent in $M_3 = \{AM_2\}$ to DV_2 over public channels as shown in Fig 2.

Step 6: After receiving message M_3 from DV_1, device DV_2 deploys session key S_{SK} and random nonce R_3 to check if $AM_2 \stackrel{?}{=} h(S_{SK}||R_3)$such that the authentication session is terminated if this equation does not hold. Otherwise, it sets S_{SK} as the shared session key with devise DV_1.

Thereafter, it computes $PID^{New} = h((ID1^{New}||ID2) \oplus MSK \oplus R2)$ and substitutes PID with PID^{New} to be utilized during the next authentication phase. Finally, it composes acknowledgment message $AM3 = h((R2^* \oplus R3)||R1^{New})$ that is forwarded in $M4 = \{AM3\}$ to device DV1.

Step 7: Upon receiving message M_4, device DV_1 validates acknowledgment message AM_3 by confirming if $AM_3 \stackrel{?}{=} h((R_2 \oplus R_3{}^{New})||R_1)$.. Provided that these two values are equivalent, device DV_1 accepts session key $S_{SK}{}^*$ and substitutes parameter set $\{R_1, C_1\}$ with $\{R_2, C_1{}^{New}\}$ in its memory. However, if AM_3 verification fails or device DV_1 fails to get this acknowledgment message within the stipulated time, the session is terminated. Essentially, this implies that device DV_1 commences a fresh session.

4 Results and Discussion

In this section, the proposed protocol is evaluated using security and performance metrics. To accomplish security analysis, different lemmas are formulated and proofed. On the other hand, performance evaluation is accomplished using computation and communication complexities.

4.1 Security Analysis

In this section, it is shown that the proposed protocol offers untraceability and anonymity. In addition, it resists attacks such as de-synchronization and known secret key leakage attacks.

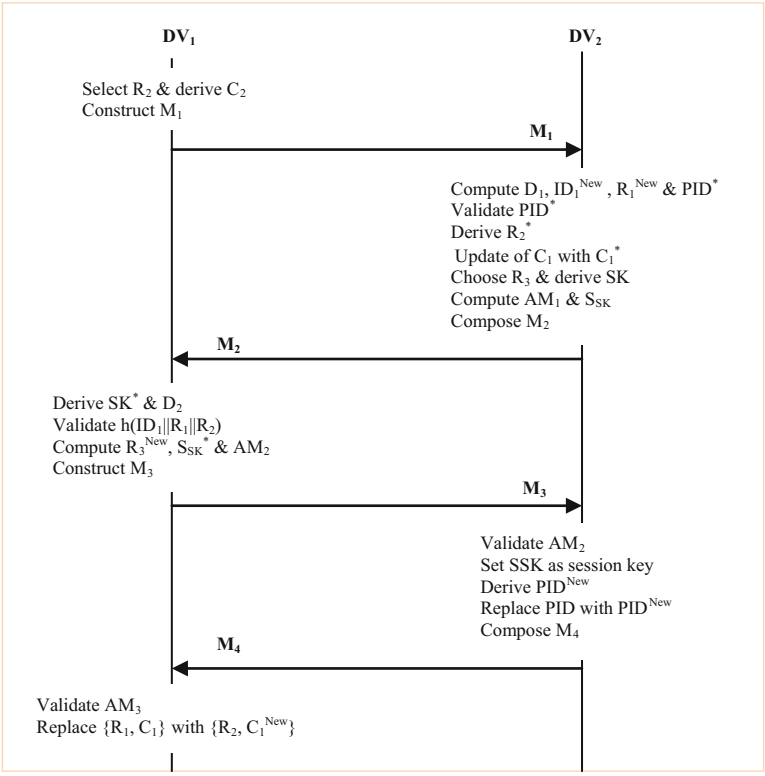

Fig. 2. Authentication and Key Negotiation

Lemma 1: The proposed protocol offers device untraceability.

Proof: In this protocol, an attacker is unable to track the origin of the exchanged messages. This is achieved by employing transcient security parameters during the authentication and key negotiation phase. For instance, during the authentication process, messages $M_1 = \{C_1, C_2\}$, $\{AM_1, AM_2\}$, and AM_3 are transient in nature. Here, $C_1 = E_{MSK}((ID_1 \oplus h(ID_2||MSK))||(R_1 \oplus ID_1))$, $C_2 = R_2 \oplus h(ID_1||R_1)$, $AM_1 = E_{SK}((h((ID_1{}^{New} \oplus R_2{}^*)||R_1{}^{New}) \oplus R_3)||h(ID_1{}^{New}||R_1{}^{New}||R_2{}^*)||C_1{}^*)$, $AM_2 = h(S_{SK}{}^*||R_3{}^{New})$, and $AM_3 = h((R_2{}^* \oplus R_3)||R_1{}^{New})$. Evidently, all these parameters incorporate high entropy random nonce R_i which are stochastically generated at the devices. Consequently, all the exchanged messages are session-specific. In addition, towards the end of each authentication and key negotiation phase, parameter set $\{R_1, C_1\}$ are replaced with refreshed values $\{R_2, C_1{}^{New}\}$. Suppose that an adversary captures authentication messages belonging to either device DV_1 or device DV_2. In the proposed protocol, the intercepted messages cannot permit the attacker to distinguish whether two sessions belong to the same device and hence tracking a particular device is infeasible.

Lemma 2: De-synchronization attacks are effectively thwarted in the proposed protocol.

Proof: In the proposed scheme, after device DV_2 verifies message AM_2, it transmits an acknowledgement message AM_3 to device DV_1. Here, the authenticity of AM_3 is verified by checking if $AM_3 \stackrel{?}{=} h((R_2 \oplus R_3^{New}) || R_1)$.. After successful verification process, session key $S_{SK}^{*} = h(ID_1||R_1||R_2||R_3^{New})$ is accepted as the shared session key with device DV_2. However, if message AM_2 or acknowledgement AM_3 cannot be delivered to device DV_1, S_{SK}^{*} is deleted and a new session is initiated. The same will take place when acknowledgement message AM_3 is delayed. In this new session, device DV_1 and DV_2 re-negotiate the session key and hence this attack is prevented.

Lemma 3: *The proposed protocol upholds device anonymity.*

Proof: In this protocol, device DV_1 identity ID_1 is masked in message $M_1 = \{C_1, C_2\}$, $AM_1 = E_{SK}((h((ID_1^{New} \oplus R_2^{*})||R_1^{New}) \oplus R_3)||h(ID_1^{New}||R_1^{New}||R_2^{*})||C_1^{*})$ and $AM_2 = h(S_{SK}^{*}||R_3^{New})$, where $S_{SK}^{*} = h(ID_1||R_1||R_2||R_3^{New})$. Suppose that an attacker attempts to extract device DV_1 identity ID_1 from message $C_1 = E_{MSK}((ID_1 \oplus h(ID_2||MSK))||(R_1 \oplus ID_1))$. However, this requires knowledge of master key MSK for device DV_2 to decrypt C_1, as well as random nonce R_1 and identity ID_2 for device DV_2. As such, devoid of $\{MSK, ID_2\}$ or $\{MSK, R_1\}$, the attacker is unable to derive ID_1 from C_1. In addition, ID_1 is protected by a one-way hash function together with high entropy random numbers $\{R_1, R_2\}$ in message C_2, where $C_2 = R_2 \oplus h(ID_1||R_1)$. As such, an adversary is unable to extract ID_1 from the intercepted C_2. For the case of authentication message $AM_1 = E_{SK}((h((ID_1^{New} \oplus R_2^{*})||R_1^{New}) \oplus R_3)||h(ID_1^{New}||R_1^{New}||R_2^{*})||C_1^{*})$, identity ID_1 is protected using symmetric key SK and random nonces $\{R_1, R_2\}$. Once again, an attacker is unable to extract ID_1 from AM_1. Similarly, ID_1 is protected using one-way hash function together with random nonces $\{R_1, R_2, R_3\}$ in message $AM_2 = h(S_{SK}^{*}||R_3^{New})$, where $S_{SK}^{*} = h(ID_1||R_1||R_2||R_3^{New})$. Consequently, the proposed protocol offers anonymity during the authentication and key negotiation phase.

Lemma 4: Known secret key leakage attacks are thwarted in this protocol.

Proof: During the authentication and key negotiattion process, session key S_{SK} is derived, where $S_{SK} = h(ID_1^{New}||R_1^{New}||R_2^{*}||R_3)$. This session key incorporates device DV_1 identity ID_1 and three random nonces $\{R_1, R_2, R_3\}$ protected via one-way hashing operation. Here, R_1 and R_2 are generated by device DV_1 in different scenarios, while random nonce R_3 is generated at device DV_2. This random and independent generation of random nonces $\{R_1, R_2, R_3\}$ implies that the derived session key is unique for each authentication instant. As such, an adversary cannot compute the session key for subsequent authentication session using the captured current session key.

Lemma 5: The proposed protocol offers protection against eavesdropping attacks.

Proof: Suppose that an attacker wants to intercept secret parameters such as master key MSK and device real identities ID_1 and ID_2. To achieve this, messages $M_1 = \{C_1, C_2\}$, $M_2 = \{S_{SK}\}$, $M_3 = \{AM_2\}$ and $M_4 = \{AM_3\}$ must be captured. Here, $C_1 = E_{MSK}((ID_1 \oplus h(ID_2||MSK))||(R_1 \oplus ID_1))$, $C_2 = R_2 \oplus h(ID_1||R_1)$, $S_{SK} = h(ID_1^{New}||R_1^{New}||R_2^{*}||R_3)$,

$AM_2 = h(S_{SK}{}^{*}||R_3{}^{New})$ and $AM_3 = h((R_2{}^{*} \oplus R_3)||R_1{}^{New})$. However, identity ID_1 is encrypted in C_1 and hashed in both C_2 and S_{SK}. On the other hand, identity ID_2 is hashed and enciphered in C_1. Similarly, master key MSK is hashed and encrypted in C_1. Since it is computationally infeasible to reverse the one-way hash function, and an attacker has no access to the secret master key, the decryption of security parameter C_1 is infeasible and hence eavesdropping attack flops.

Lemma 6: Forgery attacks are prevented in the proposed protocol.

Proof: The assumption made here is that an attacker wants to fool both devices DV_1 and DV_2 through message fabrication. To achieve this, bogus messages $C_1 = E_{MSK}((ID_1 \oplus h(ID_2||MSK))||(R_1 \oplus ID_1))$, $AM_1 = E_{SK}((h((ID_1{}^{New} \oplus R_2{}^{*})||R_1{}^{New}) \oplus R_3)||h(ID_1{}^{New}||R_1{}^{New}||R_2{}^{*})||C_1{}^{*})$, $C_2 = R_2 \oplus h(ID_1||R_1)$, $AM_2 = h(S_{SK}{}^{*}||R_3{}^{New})$ and $AM_3 = h((R_2{}^{*} \oplus R_3)||R_1{}^{New})$ need to be constructed. However, the construction of these messages require encryption key SK and master key MSK. In addition, the attacker needs to determine the hashing function deployed for the construction of these messages. However, since all these parameters are unavailable to the adversary, this attack fails.

Lemma 7: The proposed protocol offers perfect forward key secrecy.

Proof: Suppose that an attacker is interested in deriving the session key $S_{SK}{}^{New}$ for the subsequent authentication session. To accomplish this, message M_2 is intercepted, where $M_2 = \{S_{SK}\}$. Here, session key $S_{SK} = h\ (ID_1{}^{New}||R_1{}^{New}||R_2{}^{*}||R_3)$. Evidently, this requires correct generation of random nonces $R_1{}^{New}$, $R_2{}^{*}$ and R_3, as well as device DV_1 identity $ID_1{}^{New}$. Since these three nonces are randomly generated, it is infeasible to accurately generate all of them. In addition, by Lemma 5, an adversary cannot determine device identities ID_1 and ID_2. As such, perfect key secrecy is upheld.

4.2 Performance Evaluation

In this sub-section, the computation overheads, communication costs and functionality of the proposed protocol are presented. Thereafter, comparisons are also made with other related schemes so as to provide the basis for the appraisal of the proposed protocol.

Computation Costs: A typical authentication and key negotiation phase involves operations such as one-way hashing (T_H), symmetric encryption (T_{SE}), ECC point multiplication (T_M), symmetric decryption (T_{SD}) and ECC point addition (T_A). In the proposed protocol, $7T_H$ and $1T_{SD}$ operations are executed on device DV_1 while $9T_H$, $1T_{SD}$ and $2T_{SE}$ operations are carried out on device DV_2. As such, the total computation cost in this scheme is $16T_H$, $2T_{SD}$ and $2T_{SE}$. Based on the values in [41], single T_H, T_{SE} / T_{SD}, T_M and T_A execution times are 0.0023 ms, 0.0046 ms, 2.226 ms and 0.0288 ms respectively. As such, the total computation cost of the proposed protocol is 0.0552 ms as shown in Table 2.

Table 2. Computation costs

Scheme	Costs (ms)
[16]	0.8
[24]	0.88
[25]	0.64
[33]	8.98
Proposed	0.0552

It is evident from Table 2 that the scheme in [33] incurs the highest computation costs followed by the protocols in [16, 24] and [25] respectively. On the other hand, the proposed protocol has the lowest computation overheads. Owing to the processing limitations of D2D devices, the proposed protocol is the most ideal for these devices.

Communication Costs: Communication Costs: During the authentication and key agreement process, messages M_1, M_2, M_3 and M_4 are exchanged between DV_1 and DV_2. Here, $M_1 = \{C_1, C_2\}$, $M_2 = \{S_{SK}\}$, $M_3 = \{AM_2\}$, $M_4 = \{AM_3\}$, $C_1 = E_{MSK}((ID_1 \oplus h(ID_2||MSK))||(R_1 \oplus ID_1))$, $C_2 = R_2 \oplus h(ID_1||R_1)$, $S_{SK} = h(ID_1^{New}||R_1^{New}||R_2^{*}||R_3)$, $AM_2 = h(S_{SK}^{*}||R_3^{New})$ and $AM_3 = h((R_2^{*} \oplus R_3)||R_1^{New})$. Using the values in [41], the output sizes of hash function (SHA-1), timestamps, identity, symmetric encryption / decryption and ECC point are 160 bits, 32 bits, 160 bits, 128 bits and 160 bits respectively. As such, $C_1 = 128$ bits, $C_2 = 160$ bits; and hence M_1 is 288 bits long. On the other hand, $S_{SK} = AM_2 = AM_3 = 160$ bits; therefore $M_2 = M_3 = M_4 = 160$ bits in length. Consequently, the total communication cost of the proposed protocol is 768 bits as shown in Table 3.

Table 3. Communication costs

Scheme	Costs (bits)
[16]	2624
[24]	2640
[25]	2856
[33]	2016
Proposed	768

As shown in Fig. 3, the scheme in [25] has the highest communication costs followed by the schemes in [16, 24] and [33] respectively. On the other hand, the proposed protocol has the least communication cost of only 768 bits.

Since most of the D2D devices are limited in terms of battery power and communication abilities, the proposed protocol places the least constrain on these devices. As such, it is the most suitable for deployment in this environment.

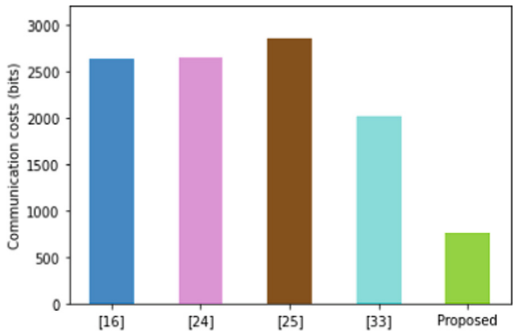

Fig. 3. Communication costs comparisons

To investigate the effect of device density on the computation costs and end-to-end latencies, a computer runing on Windows 10 Pro 64-bit with 2.4 GHz Intel Core i5-4210U CPU and 4 GB of RAM is deployed to implement the proposed protocol. Java pairing based cryptography and Bouncy Castle cryptographic libraries are used. Figure 4 shows the variation of communication costs as a function of the number of IoT devices and number of messages sent.

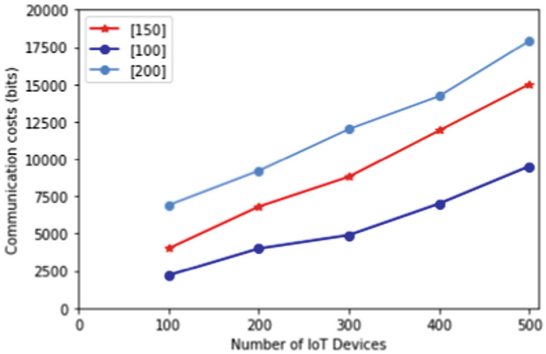

Fig. 4. Communication costs variations

As shown in Fig. 4, for a given number of IoT devices, the higher the messages transmitted the higher the communication costs and vice versa. In general, as the number of IoT devices increases so does the communication costs. Figure 5 shows the variations of end to end latencies as a function of the number of messages exchanged.

It is evident from Fig. 5 that as the number of messages increases, the values of end to end latencies increase. This is attributed to the increased processing that has to be executed at the IoT devices when the numbers of messages increase.

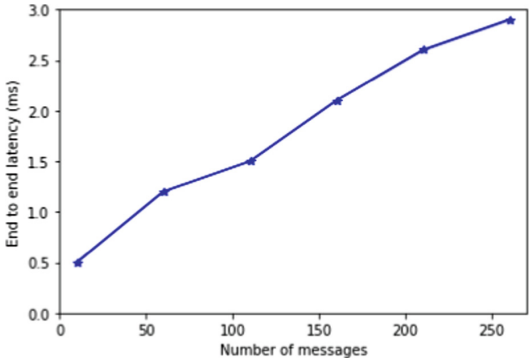

Fig. 5. End to end latency variations

5 Conclusion and Future Work

The IoT communication has found applications in a number of application domains to offer convenience and efficiency to the users. However, there are many efficiency, security and performance issues that remain unresolved in this communication environment. To this end, many security protocols have been developed in an effort to address these issues. However, the attainment of perfect security and privacy protection at low complexities still remain a mirage. On the other hand, the proposed protocol has been demonstrated to have the least communication and computational complexities. In addition, it has been shown to have resilience against many IoT attacks. As such, the proposed protocol can potentially address insecurity issues in application domains such as smart cities, smart homes and smart health. Future research directions may involve the formal verification of the security features provided by this protocol.

Acknowledgement. This work is supported by Natural Science Foundation of Top Talent of SZTU (grant No. GDRC202135).

References

1. Hao, Y., Helo, P.: The role of wearable devices in meeting the needs of cloud manufacturing: a case study. Robot. Comput.-Integr. Manuf. **45**, 168–179 (2017)
2. Dang, T.K., Pham, C.D., Nguyen, T.L.: A pragmatic elliptic curve cryptography-based extension for energy-efficient device-to-device communications in smart cities. Sustain. Cities Soc. **56**, 102097 (2020)
3. Habibzadeh, H., Nussbaum, B.H., Anjomshoa, F., Kantarci, B., Soyata, T.: A survey on cybersecurity, data privacy, and policy issues in cyber-physical system deployments in smart cities. Sustain. Cities Soc. **50**, 101660 (2019)
4. Dang, T.K., Tran, K.T.: The meeting of acquaintances: a cost-efficient authentication scheme for light-weight objects with transient trust level and plurality approach. Secur. Commun. Net. **2019**(8123259), 1–18 (2019)

5. Zhang, Y., Cheng, K., Khan, F., Alturki, R., Khan, R., Rehman, A.U.: A mutual authentication scheme for establishing secure device-to-device communication sessions in the edge-enabled smart cities. J. Inf. Secur. Appl. **58**, 102683 (2021)
6. Nyangaresi, V.O., Morsy, M.A.: Towards privacy preservation in internet of drones. In: 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI), 306–311. IEEE (2021)
7. Li, J., Zhang, W., Dabra, V., Choo, K.K.R., Kumari, S., Hogrefe, D.: AEP-PPA: an anonymous, efficient and provably-secure privacy-preserving authentication protocol for mobile services in smart cities. J. Netw. Comput. Appl. **134**, 52–61 (2019)
8. Xia, X., Ji, S., Vijayakumar, P., Shen, J., Rodrigues, J.J.: An efficient anonymous authentication and key agreement scheme with privacy-preserving for smart cities. Int. J. Distrib. Sens. Netw. **17**(6), 15501477211026804 (2021)
9. Nyangaresi, V.O.: Hardware assisted protocol for attacks prevention in ad hoc networks. In: Miraz, M.H., Southall, G., Ali, M., Ware, A., Soomro, S. (eds.) iCETiC 2021. LNICSSITE, vol. 395, pp. 3–20. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90016-8_1
10. Malik, M.N., Azam, M.A., Ehatisham-Ul-Haq, M., Ejaz, W., Khalid, A.: ADLAuth: passive authentication based on activity of daily living using heterogeneous sensing in smart cities. Sensors **19**(11), 2466 (2019)
11. Yaseen, Q.M., Aldwairi, M.: An enhanced AODV protocol for avoiding black holes in MANET. Procedia Comput. Sci. **134**, 371–376 (2018)
12. Shen, J., Chang, S., Shen, J., Liu, Q., Sun, X.: A lightweight multi-layer authentication protocol for wireless body area networks. Futur. Gener. Comput. Syst. **78**, 956–963 (2018)
13. Gope, P., Hwang, T.: An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks. J. Netw. Comput. Appl. **62**, 1–8 (2016)
14. Li, X., Peng, J., Niu, J., Wu, F., Liao, J., Choo, K.K.R.: A robust and energy efficient authentication protocol for industrial internet of things. IEEE Internet Things J. **5**(3), 1606–1615 (2017)
15. Khemissa, H., Tandjaoui, D., Bouzefrane, S.: An ultra-lightweight authentication scheme for heterogeneous wireless sensor networks in the context of internet of things. In: Bouzefrane, S., Banerjee, S., Sailhan, F., Boumerdassi, S., Renault, E. (eds.) MSPN 2017. LNCS, vol. 10566, pp. 49–62. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67807-8_4
16. Alzahrani, B.A., Irshad, A., Albeshri, A., Alsubhi, K.: A provably secure and lightweight patient-healthcare authentication protocol in wireless body area networks. Wireless Pers. Commun. **117**(1), 47–69 (2021). https://doi.org/10.1007/s11277-020-07237-x
17. Yasin, A., Abu, Z.M.: Detecting and isolating black-hole attacks in MANET using timer based baited technique. Wirel. Commun. Mob. Comput. **2018**, 1–10 (2018)
18. Chaudhry, S.A., Naqvi, H., Sher, M., Farash, M.S., Hassan, M.U.: An improved and provably secure privacy preserving authentication protocol for SIP. Peer-to-Peer Netw. Appl. **10**(1), 1–15 (2015). https://doi.org/10.1007/s12083-015-0400-9
19. Zhong, H., Huang, B., Cui, J., Xu, Y., Liu, L.: Conditional privacy-preserving authentication using registration list in vehicular ad hoc networks. IEEE Access **6**, 2241–2250 (2018)
20. Vasco, M., Pozo, A., Soriente, C.: A key for John Doe: modeling and designing anonymous password authenticated key exchange protocols. IEEE Trans. Dependable Secure Comput. **18**(3), 1336–1353 (2021)
21. Nyangaresi, V.O.: Provably secure protocol for 5G HetNets. In: 2021 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS), 17–22. IEEE (2021)
22. Wei, J., Phuong, T., Yang, G.: An efficient privacy preserving message authentication scheme for internet of-things. IEEE Trans. Industr. Inf. **17**(1), 617–626 (2021)

23. Blasco, J., Peris-Lopez, P.: On the feasibility of low-cost wearable sensors for multi-modal biometric verification. Sensors **18**(9), 2782 (2018)
24. Ibrahim, M.H., Kumari, S., Das, A.K., Wazid, M., Odelu, V.: Secure anonymous mutual authentication for star two-tier wireless body area networks. Comput. Methods Programs Biomed. **135**, 37–50 (2016)
25. Li, X., Ibrahim, M.H., Kumari, S., Sangaiah, A.K., Gupta, V., Choo, K.K.R.: Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. Comput. Netw. **129**, 429–443 (2017)
26. Nyangaresi, V.O., Abduljabbar, Z.A., Al Sibahee, M.A., Abduljaleel, I.Q., Abood, E.W.: Towards security and privacy preservation in 5G networks. In: 2021 29th Telecommunications Forum (TELFOR), pp. 1–4. IEEE (2021)
27. Shen, M., Tang, X., Zhu, L., Du, X., Guizani, M.: Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities. IEEE Internet Things J. **6**(5), 7702–7712 (2019)
28. Reddy, A.G., Suresh, D., Phaneendra, K., Shin, J.S., Odelu, V.: Provably secure pseudo-identity based device authentication for smart cities environment. Sustain. Cities Soc. **41**, 878–885 (2018)
29. Singh, P., Nayyar, A., Kaur, A., Ghosh, U.: Blockchain and fog based architecture for internet of everything in smart cities. Future Internet **12**(4), 61 (2020)
30. Tran, K.K., Pham, M.K., Dang, T.K.: A light-weight tightening authentication scheme for the objects' encounters in the meetings. In: Dang, Tran Khanh, Küng, Josef, Wagner, Roland, Thoai, Nam, Takizawa, Makoto (eds.) FDSE 2018. LNCS, vol. 11251, pp. 83–102. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03192-3_8
31. Dimitriou, T., Karame, G.O.: Enabling anonymous authorization and rewarding in the smart grid. IEEE Trans. Dependable Secure Comput. **14**(5), 565–572 (2017)
32. Kong, W., Shen, J., Vijayakumar, P., Cho, Y., Chang, V.: A practical group blind signature scheme for privacy protection in smart grid. J. Parallel Distrib. Comput. **136**, 29–39 (2020)
33. Mandal, S., Mohanty, S., Majhi, B.: Cryptanalysis and enhancement of an anonymous self-certified key exchange protocol. Wireless Pers. Commun. **99**(2), 863–891 (2017). https://doi.org/10.1007/s11277-017-5156-5
34. He, D., Kumar, N., Khan, M.K., Wang, L., Shen, J.: Efficient privacy-aware authentication scheme for mobile cloud computing services. IEEE Syst. J. **12**(2), 1621–1631 (2018)
35. Nyangaresi, V.O., et al.: Provably secure session key agreement protocol for unmanned aerial vehicles packet exchanges. In: 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), pp. 1–6. IEEE (2021)
36. Wang, K.H., Chen, C.M., Fang, W., Wu, T.Y.: A secure authentication scheme for internet of things. Pervasive Mob. Comput. **42**, 15–26 (2017)
37. Wu, L., Wang, J., Choo, K.K.R., He, D.: Secure key agreement and key protection for mobile device user authentication. IEEE Trans. Inf. Forensics Secur. **14**(2), 319–330 (2019)
38. Boyapally, H., et al.: Safe is the new smart: PUF-based authentication for load modification-resistant smart meters. IEEE Trans. Dependable Secure Comput. **19**(1), 663–680 (2022)
39. Suzuki, M., Ueno, R., Homma, N., Aoki, T.: Efficient fuzzy extractors based on ternary debiasing method for biased physically unclonable functions. IEEE Trans. Circ. Syst. **66**(2), 616–629 (2019)
40. Nyangaresi, V.O., Petrovic, N.: Efficient PUF based authentication protocol for internet of drones. In: 2021 International Telecommunications Conference (ITC-Egypt), pp. 1–4. IEEE (2021)
41. Alzahrani, B.A., Chaudhry, S.A., Barnawi, A., Al-Barakati, A., Shon, T.: An anonymous device to device authentication protocol using ECC and self certified public keys usable in Internet of Things based autonomous devices. Electronics **9**(3), 520 (2020)