PLOS ONE

RESEARCH ARTICLE

A secure remote user authentication scheme for 6LoWPAN-based Internet of Things

Ghulam Abbas[1,2☯], **Muhammad Tanveer**[2☯], **Ziaul Haq Abbas**[3☯], **Muhammad Waqas**[4☯], **Thar Baker**[5☯], **Dhiya Al-Jumeily OBE**[6☯*]

1 Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Topi, Pakistan, **2** Telecommunications and Networking Research Center, GIK Institute of Engineering Sciences and Technology, Topi, Pakistan, **3** Faculty of Electrical Engineering, GIK Institute of Engineering Sciences and Technology, Topi, Pakistan, **4** Engineering Research Center of Intelligent Perception and Autonomous Control, Faculty of Information Technology, Beijing University of Technology, Beijing, China, **5** Department of Computer Science, University of Sharjah, Sharjah, United Arab Emirates, **6** School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, United Kingdom

☯ These authors contributed equally to this work.
* d.aljumeily@ljmu.ac.uk

Citation: Abbas G, Tanveer M, Abbas ZH, Waqas M, Baker T, Al-Jumeily OBE D (2021) A secure remote user authentication scheme for 6LoWPAN-based Internet of Things. PLoS ONE 16(11): e0258279. https://doi.org/10.1371/journal.pone.0258279

Editor: Pandi Vijayakumar, University College of Engineering Tindivanam, INDIA

Received: June 30, 2021

Accepted: September 22, 2021

Published: November 8, 2021

Data Availability Statement: Minimal data set underlying the results described in this paper can be found https://github.com/TanveerPhD/Minimal-data/blob/main/Data.ods.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

One of the significant challenges in the Internet of Things (IoT) is the provisioning of guaranteed security and privacy, considering the fact that IoT devices are resource-limited. Oftentimes, in IoT applications, remote users need to obtain real-time data, with guaranteed security and privacy, from resource-limited network nodes through the public Internet. For this purpose, the users need to establish a secure link with the network nodes. Though the IPv6 over low-power wireless personal area networks (6LoWPAN) adaptation layer standard offers IPv6 compatibility for resource-limited wireless networks, the fundamental 6LoWPAN structure ignores security and privacy characteristics. Thus, there is a pressing need to design a resource-efficient authenticated key exchange (AKE) scheme for ensuring secure communication in 6LoWPAN-based resource-limited networks. This paper proposes a resource-efficient secure remote user authentication scheme for 6LoWPAN-based IoT networks, called SRUA-IoT. SRUA-IoT achieves the authentication of remote users and enables the users and network entities to establish private session keys between themselves for indecipherable communication. To this end, SRUA-IoT uses a secure hash algorithm, exclusive-OR operation, and symmetric encryption primitive. We prove through informal security analysis that SRUA-IoT is secured against a variety of malicious attacks. We also prove the security strength of SRUA-IoT through formal security analysis conducted by employing the random oracle model. Additionally, we prove through Scyther-based validation that SRUA-IoT is resilient against various attacks. Likewise, we demonstrate that SRUA-IoT reduces the computational cost of the nodes and communication overheads of the network.

1 Introduction

Low-power wireless personal area networks (LoWPANs) have found numerous applications in the Internet of Things (IoT) [1]. LoWPAN devices are amenable with IEEE 802.15.4 and are constricted in power, communication, data rate, and storage resources [2]. IEEE 802.15.4-enabled LoWPAN devices are deployed in various real-world applications, such as home automation, healthcare systems, security surveillance, smart grids, and industrial motoring. To provide Internet connectivity to a large number of devices deployed in a particular IoT environment, the IPv6 protocol is considered the most accordant solution because of its larger address space to render a unique IP address to each sensor node. By using IPv6 addressing, sensor nodes can transmit sensed information to other devices or to a central location through the public Internet.

To support large-scale connectivity for IoT, the Internet Engineering Task Force has designed IPv6-over-LoWPAN (6LoWPAN) adaptation layer to render packet fragmentation, reassembly, and encapsulation features for IEEE 802.15.4-based LoWPAN networks [3, 4]. Since LoWPAN devices collect information and send to a designated location via the public Internet, it is imperative for LoWPAN applications to provide security and privacy. However, the basic 6LoWPAN design does not provide security and privacy features to preclude an unauthorized network entity from procuring the collected information and to prevent illegitimate users from accessing the 6LoWPAN network resources [5–9].

6LoWPANs encounter the same security attacks as the traditional networks. These include denial-of-service (DoS), replay, user/server impersonation (UI/SI), man-in-the-middle (MITM), identity guessing (IG), user anonymity (UA), user/device impersonation (UI/DI), stolen smart card/device (SSC/SSD), and ephemeral secret leakage (ESL) attacks. However, due to the resource-constricted nature of 6LoWPANs and the inadequacy of organized network architectures, securing 6LoWPAN becomes more challenging [10]. Authentication, availability, integrity, data freshness, and confidentiality are imperative security provisions in 6LoWPANs. Confidentiality guarantees secure data transmission between authorized users and servers. Authentication and key establishment (AKE) is the mechanism to identify devices' and users' legitimacy in 6LoWPANs [11] and to set up a secret session key (SK) for encrypted communication. Therefore, a lightweight AKE mechanism becomes imperative for securing the network [12–20].

1.1 Related work

An overview of the existing AKE schemes for 6LoWPAN-based IoT networks and their limitations is presented in S1 Table, which shows that no existing scheme can withstand all known attacks. Pandi *et al.* [42] propounded an authentication scheme for vehicular ad-hoc networks (VANETs) to enable the network entities to communicate securely. The scheme presented by Pandi *et al.* is efficient in terms of certificate computation while preserving privacy of the entities. Pandi *et al.* [43] presented an AKE scheme for IoT-based wireless body area networks (WBANs), which is computationally less expensive and ensures secure communication. Azees *et al.* [44] propounded an anonymous authentication scheme for WBANs, which is capable of resisting various covert security attacks while requiring fewer resources. Azees *et al.* [45] presented a blockchain based authentication scheme for VANETs, which is capable of resisting different security attacks and renders secure communication in VANETs.

The authors in [34] propounded a multi-factor AKE scheme for the IoT environment. The AKE scheme proffered in [34] uses a lightweight hash function along with advanced encryption standard (AES). However, the scheme is unable to restrain SSD, DoS, replay, and sensor node (SN) capture attacks. A Chinese remainder theorem-based authentication scheme is

presented in [23], which cannot resist replay attack and does not provide strong privacy. In addition, a signature and certificate-based computationally efficient authentication scheme for VANETs is presented in [46]. The authors in [47] propounded a resource-efficient AKE scheme for the IoT environment by utilizing hash function and XOR operation. However, the AKE scheme presented in [47] is prone to SSD, stolen verifier, UI, and UA attacks and is unable to ensure SN's anonymity. An AKE scheme is propounded in [48] for mobile networks. The scheme proposed in [48] is resource-efficient and is suitable for mobile networks. A cosine similarity-based AKE scheme for the IoT environment is proposed in [49]. Furthermore, to enable security and privacy in different IoT-based networks, various AKE schemes are reported in the exiting literature [19, 50–66].

Additionally, the security analysis of an eminent AKE scheme presented in [31] is given at S1 Appendix. We have thoroughly analyzed the scheme and demonstrate that it is unsafe against de-synchronization attack and does not provide a revocation phase (RP). In [31], gateway broadcasts the authentication message to all sensor nodes deployed in the network, and a user does not specify the sensor node from which it is going to procure the information. Thus, all the sensor nodes in the network process the received message, which causes an extra computational overhead for every node.

1.2 Research contributions

This paper presents a resource-efficient secure remote user authentication scheme for 6LoWPAN-based IoT networks (SRUA-IoT). The proposed scheme performs user authorization before procuring real-time data from sensors stationed in the 6LoWPAN-based IoT networks. The scheme employs a lightweight secure hash algorithm (SHA-160) and advanced encryption standard (AES-192) to accomplish the AKE process and makes the following contributions.

1. SRUA-IoT is an AES and hash function based remote user AKE scheme for 6LoWPAN-based IoT networks, which renders user revocation and password change phases. Besides, SRUA-IoT ensures the legitimacy of remote users (RUs) to access real-time information from a sensor node while ensuring the privacy and anonymity of RUs. An RU indicates to the gateway a particular sensor node for procuring real-time information, which reduces the unnecessary computational overhead.
2. SK's security in SRUA-IoT is corroborated using random oracle model (ROM). Informal security validation illustrates that SRUA-IoT is protected against SSC, de-synchronization, replay, and DoS attacks. In addition, Scyther tool analysis illustrates that the proposed scheme is protected.
3. We demonstrate that SRUA-IoT renders enhanced security functionalities aside from its low storage, computational, and communication costs, as compared to well-known AKE schemes.

1.3 Paper organization

The remainder of this paper is organized as follows. The system model is presented in Section 2. The proposed SRUA-IoT scheme is elaborated in Section 3. Security analysis is presented in Section 4. Performance evaluation of SRUA-IoT is detailed in Section 5. Finally, the paper is concluded in Section 6.

2 System model

The network model consists of a gateway GW, a registration center (RC), and remoter users ($RU_y|y = 1, 2, 3, \cdots, N$). In the SH environment, sensor nodes ($SN_x|x = 1, 2, 3, \cdots, n$) are deployed to monitor various processes. SN_x collect critical information and forward to the server stationed at RC. RC is responsible for the deployment of SN_x and implementing various access control policies in SH. Before procuring real-time information from SN_x, it is necessary for RU_y to register with RC. After registration, RU_y can access the network resources and the allocated SN_x. It is assumed that all network nodes are time synchronized.

The well-established Dolev Yao (DY) threat model [67] is employed, wherein an adversary $\mathcal{A}$ can intercept communications between two network entities communicating via a public channel. $\mathcal{A}$ can modify the intercepted messages or use the message for various malicious purposes. $\mathcal{A}$ can procure the secret credentials stored in a sensor node's memory. Furthermore, $\mathcal{A}$ can obtain RU_y's smart device SD_y and can extract secret credentials form SD_y to execute various security attacks.

RU_y needs to communicate with SN_x to securely procure the real-time information collected by SN_x. Therefore, an AKE scheme is imperative for secure and reliable communications between RU_y and SN_x. To achieve reliable and secure communication, the following section presents an RU_y AKE scheme, called SRUA-IoT.

3 The proposed SRUA-IoT scheme

SRUA-IoT seeks to ensure reliable and secure access to 6LoWPAN network resources. The scheme first verifies the authenticity of RU_y and then establishes a secret SK for encrypted communication by employing a lightweight hash function and AES-192 during the AKE process. SHA is an irreversible function, which means that it is impossible to derive the input from the output of SHA-160. Moreover, SHA-160 is a collision resistance function, which means that the output of SHA-160 can never be the same for different inputs. AES-192 is used as the encryption and decryption scheme in SRUA-IoT. SRUA-IoT is composed of seven phases, which are presented in the following subsections. S2 Table lists the notations used in this paper.

3.1 Sensor node deployment phase

RC assigns various secret credentials to SN_x before its deployment in the 6LoWPAN network. Moreover, RC selects a GW's secret Key (GK) of 512 bits and a unique identity ID_G. Both GK and ID_G are known only to GW. RC executes the following steps to accomplish the sensor node deployment (SND) phase.

3.1.1 Step SND-1. RC picks a unique ID_{SN_x} and PID_{SN_x} each of size 80 bits. Moreover, RC selects a random number R_x and computes a temporary secret (TS) for SN_x as $A_e = H(GK \| R_x \| ID_G)$, and $TS_{SN_x} = A_e^a \oplus A_e^b$, where A_e^a and A_e^b are two chunks of A_e, each of size 80 bits.

3.1.2 Step SND-2. RC stores the credentials $\{ID_{SN_x}, PID_{SN_x}, TS_{SN_x}\}$ in the memory of SN_x before its deployment.

3.2 Remote user registration phase

It is imperative for RC to register RU_y before providing access to the 6LoWPAN network resources. RC assigns different secret credentials and a list of SN_x to RU_y. RC executes the following steps to perform the remote user registration (RUR) phase.

3.2.1 Step RUR-1. RU_y selects a distinct identity ID_{RU_y} and computes $HID_y = H(ID_{RU_y})$. Moreover, RU_y contrives a registration message $ME_1^r : \{HID_y\}$ and dispatches ME_1^r to RC via a protected channel.

3.2.2 Step RUR-2. RC selects a distinct pseudonym PID^x for RU_y and calculates $A_q = H(GK \parallel ID_G)$, and $A_x = H(HID \parallel ID_G \parallel GK)$. RC determines a TS credential for RU_y by dividing A_x into two equal parts, namely, A_x^a and A_x^b, each of size 80 bits, and computes $TS_{RU_y} = A_x^a \oplus A_x^b$. Moreover, RC computes the revocation parameter (ReP) as $B_x = A_q \oplus HID_y$ and $RP_{RU_y} = B_x^a \oplus B_x^b$, where B_x^a and B_x^b are two chunks of B_x. Besides, RC assigns a list of SN_x to be accessed by RU_y. Furthermore, RC computes encryption key as $EK = (A_q \parallel [A_q]^{32})$, where $[A_q]^{32}$ are the first 32 bits of A_q (to make the size of EK 192 bits). In addition, RC derives $CT_{RU_y} = E_{EK}\{TS_{RU_y}, PID_{SN_x}, TS_{SN_x}\}$ by using AES-192, and stores a list of credentials $\{PID^x, RP_{RU_y}, CT_{RU_y}\}$ in GW's memory. Finally, RC fabricates a message $ME_2^r : \{PID^x, TS_{RU_y}, PID_{SN_x}\}$ and sends ME_2^r to RU_y securely.

3.2.3 Step RUR-3. After procuring ME_2^r from RC, RU_y supplies its ID_{RU_y}, password PS_{RU_y} and B_{RU_y} at the interface of smart device SD_y and computes $(\beta_k, Rp) = Gen(B_{RU_y})$ by using fuzzy extractor (FE). FE consist of two functions. The first one is $Gen(.)$, which is a probabilistic function that takes bio-metric information B_{RU_y} of RU_y and produces two output parameters, namely, secret bio-metric key β_k and reproduction parameter Rp. The second function of FE is $Rep(.)$, which is a deterministic function that takes Rp and B_{RU_y} to reproduce β_k. Moreover, SD_y calculates $Z_x = H(PID^x \parallel TS_{RU_y} \parallel PID_{SN_x})$, $Z_y = H(ID_{RU_y} \parallel PS_{RU_y} \parallel \beta_k)$, and encryption key $EK_y = (Z_y \parallel [Z_y]^{32})$, where $[Z_y]^{32}$ are the first 32 bits of Z_y to create EK_y of size 192 bits. Furthermore, SD_y calculates $CT_{lo} = E_{EK_y}\{PID^x, TS_{RU_y}, CT_{RU_y}\}$ by using AES-192. In addition, SD_y computes authentication parameter as $Auth_y = H(ID_{RU_y} \parallel PS_{RU_y} \parallel \beta_k \parallel Z_x)$.

3.2.4 Step RUR-4. Finally, SD_y stores the list of credentials $\{CT_{lo}, Auth_y, Rp, Gen(.), Rep(.), Et\}$ in its memory and deletes all other parameters.

3.3 RU AKE phase

To access and communicate with the deployed 6LoWPAN based devices, it is necessary for RU_y to register itself with RC. RC allocates a list of secret credentials and devices to RU_y at the time of registration. After authorizing RU_y's legitimacy, RC allows RU_y to access the specified devices deployed in the network. After getting authenticated by RC, RU_y and SN_x set up an SK for reliable and secure communication. The following steps elaborate RU AKE phase (RAP).

3.3.1 Step RAP-1. SD_y receives the secret credentials PS_{RU_y}, ID_{RU_y}, and B_{RU_y}, and computes $\beta_k = Rep(B_{RU_y}, Rp)$ and $Z_y = H(ID_{RU_y} \parallel PS_{RU_y} \parallel \beta_k)$. In addition, SD_y computes the decryption key DK_{lo} as $DK_{lo} = (Z_y \parallel [Z_y]^{32})$, where $[Z_y]^{32}$ are the first 32 bits of Z_y to make DK_{lo} of size 192 bits. Moreover, SD_y computes $PT_{lo} = D_{DK_{lo}}\{CT_{lo}\}$, where CT_{lo} is the ciphertext stored in SD_y, and retrieves $PT_{lo} = \{PID^x, TS_{RU_y}, PID_{SN_x}\}$. Furthermore, SD_y calculates $Z_x^{lo} = H(PID^x \parallel TS_{RU_y} \parallel PID_{SN_x})$, and authentication parameter $Auth_{lo} = H(ID_{RU_y} \parallel PS_{RU_y} \parallel \beta_k \parallel Z_x)$. Finally, SD_y checks $Auth_y = Auth_{lo}$ to perform local authentication. If the condition holds, SD_y continues the AKE process.

3.3.2 Step RAP-2. After performing the local authentication, SD_y chooses T_x of size 32 bits, and R_1 of size 80 bits. SD_y calculates $G_1 = (R_1 \parallel PID_{SN_x}) \oplus H(TS_{RU_y} \parallel T_x)$ and $Auth_{a1} = H(PID^x \parallel PID_{SN_x} \parallel R_1 \parallel TS_{RU_y})$. Furthermore, SD_y contrives a message ME_a: $\{T_x, PID^x, G_1, Auth_{a1}\}$ and dispatches it to GW via an open communication channel.

3.3.3 Step RAP-3. Upon procuring ME_a from SD_y, GW verifies the validity of timestamp by validating the condition $TD_x \geq |Tr - T_x|$, where TD_x is maximum tolerable packet time delay, Tr is the receiving time of ME_a, and T_x is fabrication time of ME_a. If ME_a receives at the GW within the maximum allowed time delay limit, GW considers ME_a to be a licit and fresh message and continues the AKE phase. GW picks PID^x from the received ME_a and looks up PID^x in GW's memory. If found, GW extracts the list of credentials $\{PID^x, RP_{RU_y}, CT_{RU_y}\}$ related to PID^x. In addition, GW calculates DK as $M_1 = H(GK \| ID_G)$ and $DK = (M_1 \| [M_1]^{32})$. Moreover, GW computes $PT_1 = D_{DK}\{CT_{RU_y}\}$ by using AES-192, and procures secret credentials $\{TS_{RU_y}, (PID_{SN_x}, TS_{SN_x})\}$ from PT_1. Furthermore, GW obtains R_1 and PID_{SN_x} by computing $(R_1 \| PID_{SN_x}) = G_1 \oplus H(TS_{RU_y} \| T_x)$. To validate the authenticity of ME_a, GW calculates $Auth_{a2} = H(PID^x \| PID_{SN_x} \| R_1 \| TS_{RU_y})$ and verifies the condition $Auth_{a1} = Auth_{a2}$. If the condition holds, GW continues the execution of the AKE process.

3.3.4 Step RAP-4. After validating the authenticity of ME_a, GW picks a timestamp T_y and random number R_2, and computes $W_1 = H(R_1 \| TS_{RU_y} \| PID^x)$, where W_1 is obtained using hash of the parameters, including R_1, TS_{RU_y}, and PID^x. GW calculates the update parameter (UP) as $UP = W_1^a \oplus W_1^b$, where W_1^a and W_1^b are obtained by dividing W_1 into two equal chunks of 80 bits each. Besides, GW computes $PID^{x+1} = UP \oplus PID^x$ and stores both PID^x and PID^{x+1} in its memory to avoid the de-synchronization attack. Moreover, GW calculates $W_2 = H(TS_{SN_x} \| PID_{SN_x} \| T_y)$, $G_2 = W_1 \oplus W_2$, $G_3 = (R_2, R_1) \oplus W_2$, and $Auth_{a3} = H(W_1 \| R_2 \| R_1 \| TS_{SN_x} \| PID_{SN_x} \| T_y)$. Finally, GW creates a message ME_b: $\{T_y, G_2, G_3, Auth_{a3}\}$ and sends it to SN_x via the public channel.

3.3.5 Step RAP-5. After procuring ME_b from GW, SN_x verifies the condition $TD_x \geq |Tr - T_y|$. If the condition holds, SN_x computes $W_3 = H(TS_{SN_x} \| PID_{SN_x} \| T_y)$, $W_1 = G_2 \oplus W_3$, and $(R_2, R_1) = G_3 \oplus W_3$. Moreover, SN_x calculates $Auth_{a4} = H(W_1 \| R_2 \| R_1 \| TS_{SN_x} \| PID_{SN_x} \| T_y)$. Furthermore, SN_x determines the integrity of ME_b by validating the condition $Auth_{a3} = Auth_{a4}$. If the condition holds, SN_x picks a timestamp T_z and a random number R_2, and computes $G_4 = H(R_1 \| R_2 \| R_3) \oplus W_1$. For securing communication with RU_y, SN_x calculates $SK_x = H(H(R_1 \| R_2 \| R_3) \| W_1 \| T_z \| PID_{SN_x})$. In addition, SN_x computes $Auth_{a5} = H(H(R_1 \| R_2 \| R_3) \| R_1 \| T_z \| SK_x)$. Finally, SN_x calculates a message ME_c: $\{T_z, G_4, Auth_{a5}\}$ and sends it to RU_y via the public channel.

3.3.6 Step RAP-6. RU_y considers the received ME_c fresh if the condition $TD_z \geq |Tr - T_z|$ holds. If ME_c is valid, RU_y calculates $W_4 = H(R_1 \| TS_{RU_y} \| PID^x)$, and $H(R_1 \| R_2 \| R_3) = G_4$ W_4. For encrypted communication with SN_x, RU_y computes $SK_y = H(H(R_1 \| R_2 \| R_3) \| W_4 \| T_z \| PID_{SN_x})$. Furthermore, RU_y computes $Auth_{a6} = H(H(R_1 \| R_2 \| R_3) \| R_1 \| T_z \| SK_y)$ and checks $Auth_{a5} = Auth_{a6}$. If the equation holds, RU_y considers ME_c as a valid message. Finally, RU_y computes $UP = W_4^a \oplus W_4^b$ and updates PID^x by calculating $PID^{x+1} = PID^x \oplus UP_1$. RU_y keeps both PID^{x+1} and PID^x in its memory to ensure resistance against de-synchronization attack. The user AKE phase of SRUA-IoT is summarized in S1 Fig.

3.4 Password change phase

In SRUA-IoT, an authorized user RU_y can change its password and update bio-metric information without involving RC. RU_y needs to perform the following steps to execute the password change phase (PCP).

3.4.1 Step PCP-1. RU_y provides its secret credentials, namely, $ID^o_{RU_y}$, $PS^o_{RU_y}$, and $B^o_{RU_y}$ as inputs at the interface of SD_y. After procuring the inputs, SD_y computes the bio-metric key $\beta^o_k = Rep(B^o_{RU_y}, Rp^o)$. Moreover, SD_y derives the decryption Key DK^o_{lo} by computing $Z^o_y = H(ID^o_{RU_y} \| PS^o_{RU_y} \| \beta^o_k)$, and $DK^o_{lo} = (Z^o_y \| [Z^o_y]^{32})$. By using AES-192 decryption algorithm, SD_y calculates $PT^o_{lo} = D_{DK^o_{lo}}\{CT^o_{lo}\}$, where $PT^o_{lo} = \{PID^x, TS_{RU_y}, PID_{SN_x}\}$. Furthermore, SD_y computes $Z^o_x = H(PID^x \| TS_{RU_y} \| PID_{SN_x})$, $Auth^o_{lo} = H(ID^o_{RU_y} \| PS^o_{RU_y} \| \beta^o_k \| Z^o_x)$, and verifies if the condition $Auth^o_{lo}$=$Auth_{lo}$ holds. If it holds, SD_y notifies RU_y to enter a new password $PS^n_{RU_y}$ and update bio-metric information $B^n_{RU_y}$. Otherwise, SD_y halts the AKE process.

3.4.2 Step PCP-2. Upon procuring $PS^n_{RU_y}$ and $B^n_{RU_y}$ from RU_y, SD_y determines a new bio-metric key β^n by computing $(\beta^n, Rp^n) = Gen(B^n_{RU_y})$. Moreover, SD_y computes the encryption key EK^n_{lo} as $Z^n_y = H(ID^o_{RU_y} \| PS^n_{RU_y} \| \beta^n_k)$, $EK^n_{lo} = (Z^n_y \| [Z^n_y]^{32})$, where $[Z^n_y]^{32}$ are the first 32 bits of Z^n_y. Furthermore, SD_y calculates new plaintext PT^n_{lo} by deriving $PT^n_{lo} = \{PID^x, TS_{RU_y}, PID_{SN_x}\}$. In addition, SD_y computes $Z^n_x = H(PID^x \| TS_{RU_y} \| PID_{SN_x})$, and $Auth^n_{lo} = H(ID^o_{RU_y} \| PS^n_{RU_y} \| \beta^n_k \| Z^n_x)$. Finally, by utilizing AES-192 encryption algorithm, SD_y calculates $CT^n_{lo} = E_{EK^n_{lo}}\{PT^n_{lo}\}$, replaces $\{CT^n_{lo}, Auth^n_y, Rp^n, Gen(.), Rep(.), Et^n\}$ with $\{CT_{lo}$, $Auth_y$, Rp, $Gen(.)$, $Rep(.)$, $Et\}$ in SD_y's memory, and deletes all other credentials in its memory. S2 Fig summarizes PCP.

3.5 Revocation phase

If a legitimate RU_y loses its SD_y, RU_y can obtain a new SD^{new}_y from RC. To obtain SD^{new}_y, it is necessary for RU_y to remember its ID_{RU_y}. For proper RP, it is necessary to remove the previous data from GW's memory. Most AKE schemes do not delete the old data from the memory of GW or server. RU_y needs to perform the succeeding steps to procure a new SC.

3.5.1 Step RP-1. Upon getting ID_{RU_y}, SD_y computes $HID_y = H(ID_{RU_y})$, constructs a message $ME^{rov}_1 : \{HID_y\}$, and forwards ME^{rov}_1 to RC. After getting ME^{rov}_1 from RU_y, RC computes $B = H(GK \| ID_G) \oplus HID_y$, $RP_{RU_y} = B^a \oplus B^b$, and verifies if RP_{RU_y} exists in its memory. If found, RC removes RP_{RU_y} related record and informs RU_y for new registration by sending $ME^{rov}_1 : \{registration \quad request\}$ to RU_y.

3.5.2 Step RP-2. Upon getting the new registration request, RU_y picks new $PS^{new}_{RU_y}$, $ID^{new}_{RU_y}$, and computes $HID^{new} = H(ID^{new}_{RU_y})$. SD_y constructs a message $ME^{rov}_3 : \{HID^{new}_y\}$ and sends to RC.

3.5.3 Step RP-3. RC picks a new pseudonym PID^x_{new} for RU_y and computes $A^{new}_q = H(GK \| ID_G)$. To issue a new SD^{new}_y to RU_y, RC computes the same computation as accomplished in Step RUR-2 of Section 3.2. Finally, RC contrives a message $ME^{rov}_4 : \{PID^x_{new}, TS^{new}_{RU_y}, PID^{new}_{SN_x}\}$ and sends ME^{rov}_4 to RU_y via a reliable channel.

3.5.4 Step RP-4. After receiving ME^{rov}_4 from RC, SD_y executes the same computation as excuted in Step RUR-3 of Section 3.2 Finally, SD_y stores a new list of parameters $\{CT^{new}$, $Auth^{new}_y$, $Gen(.)$, $Rep(.)$, Rp^{new}, $Et^{new}\}$ in SD_y's memory. Moreover, RC stores a list of credentials $\{PID^x_{new}, RP^{new}_{RU_y}, CT^{new}_{RU_y}\}$ in GW's memory. The revocation phase is summarized in S3 Fig.

3.6 New SN deployment phase

RC can deploy a new SN (NSN) by performing the following steps.

3.6.1 Step NSN-1. RC picks a distinct $ID^n_{SN_x}$ and $PID^n_{SN_x}$ for NSN SN^n_x. In addition, RC picks R^n_x and computes a new temporary secret $TS^n_{SN_x}$ for SN^n_x by calculating $A^n_e = H(GK \parallel R^n_x \parallel ID_G)$, and $TS^n_{SN_x} = A^{n-a}_e \oplus A^{n-b}_e$, where A^{n-a}_e and A^{n-b}_e are two chunks of A^n_e, each of size 80 bits.

3.6.2 Step NSN-2. Finally, RC stores the credentials $\{ID^n_{SN_x}, PID^n_{SN_x}, TS^n_{SN_x}\}$ in SN^n_x's memory before its deployment.

4 Security analysis

In this section informal security analysis of SRUA-IoT is carried out to shows its resistance against various security attacks. The security of SK is validated by utilizing the well-known ROM. Scyther based security analysis is performed to validate SRUA-IoT's resistance against replay and MITM attacks.

4.1 Informal security analysis

This subsection illustrates that the proposed scheme is protected against various attacks, namely, replay, MITM, UI, offline PG, PI, and impersonation attacks.

Proposition 1 *SRUA-IoT is resistant to replay attack.*

proof 4.1 *There are three messages exchanged during the execution of the AKE phase, namely, ME_a: $\{T_x, PID^x, G_1, Auth_{a1}\}$, ME_b: $\{T_y, G_2, G_3, Auth_{a3}\}$, and ME_c: $\{T_z, G_4, Auth_{a5}\}$. These messages are constructed by incorporating latest timestamps T_x, T_y, and T_z. The freshness of each timestamp is verified by validating the conditions $TD_x \geq |Tr - T_x|$, $TD_x \geq |Tr - T_y|$, and $TD_x \geq |Tr - T_z|$ for each message ME_a, ME_b, and ME_c, respectively. If these conditions do not hold, GW, SN_x, and RU_y will detect the replay attack and the receiving network entity will discard the received message. Therefore, SRUA-IoT is resistant to replay attack.*

Proposition 2 *SRUA-IoT is protected against DoS attack.*

proof 4.2 *In SRUA-IoT, RU_y uses its secret credentials to pass the local authentication, for which SD_y needs to calculate $Auth_{lo} = H(ID_{RU_y} \parallel PS_{RU_y} \parallel \beta_k \parallel Z_x)$ and check the condition $Auth_y = Auth_{lo}$. Local verification will be successful if the condition holds. After local verification, SD_y sends the AKE request to GW. Otherwise, SD_y terminates the AKE process and prevents RU_y from sending a large number of AKE requests to GW. Hence, SRUA-IoT is protected against DoS attack.*

Proposition 3 *SRUA-IoT ensures untraceability and anonymity of RU_y.*

proof 4.3 *In SRUA-IoT, during the registration and the AKE phase, only pseudo identities are used, which do not provide any information about ID_{RU_y}. For each new AKE session, RU_y utilizes the updated PID^{x+1}, and fresh random numbers R_1, R_2, and R_3. During the AKE process, the communicated messages are different for each session. Therefore, $\mathcal{A}$ cannot correlate the captured message from two different AKE sessions. Thus, SRUA-IoT renders the anonymity and untraceability of RU_x and SN_x.*

Proposition 4 *SRUA-IoT is protected against MITM attack.*

proof 4.4 *In SRUA-IoT, there are three messages exchanged, i.e., ME_a, ME_b, and ME_c. Let $\mathcal{A}$ captures the the message ME_a: $\{T_x, PID^x, G_1, Auth_{a1}\}$, which is transmitted by RU_y, and tries to update the message content by selecting a random number R^a_1 and timestamp T^a_x. For this, $\mathcal{A}$ needs to compute G^a_1 and $Auth^a_{a1}$ to pretend that ME^a_a is from a legitimate RU_y. However, $\mathcal{A}$ cannot compute valid G_1 and $Auth_{a1}$ without knowing the secret credentials, namely, TS_{RU_y}, and PIS_{SN_x}, which are known only to RU_y. We can illustrate the same conditions for ME_b, and ME_c. Hence, SRUA-IoT is protected against MITM attack.*

Proposition 5 *SRUA-IoT is immune to offline PG and SSC attacks.*

proof 4.5 *In this case, $\mathcal{A}$ can execute various attacks by procuring sensitive information stored on the stolen/lost smart card or device. Let $\mathcal{A}$ obtains lost/stolen SD_y of RU_y and, by using power analysis attack, can procure the information, such as $\{CT_{lo}, Auth_y, Rp, Gen(.), Rep(.), Et\}$ stored in the memory of SD_y. From the obtained information, $\mathcal{A}$ cannot retrieve secret credentials, which are used during the AKE process. Therefore, SRUA-IoT is protected against SSC attack. To update the password of RU_y, $\mathcal{A}$ picks a random identity, password and bio-metric information to compute $\beta_k^a = Rep(B_{RU_y}^a, Rp)$, $Z_y^a = H(ID_{RU_y}^a \parallel PS_{RU_y}^a \parallel \beta_k^a)$, $DK_{lo}^a = (Z^a y \parallel [Z_y^a]^{32})$, and $PT_{lo}^a = D_{DK_{lo}^a}\{CT_{lo}\}$, retrieve $PT_{lo}^a = \{PID^x, PID_{SN_x}^a, TS_{RU_y}^a\}$, calculate $Z_x^a = H(PID^x \parallel TS_{RU_y}^a \parallel PID_{SN_x}^a)$, $Auth_a = H(ID_{RU_y}^a \parallel PS_{RU_y}^a \parallel \beta_k^a \parallel Z_x^a)$, and check $Auth_y^a \stackrel{?}{=} Auth_{lo}$. However, without knowing the secret credentials of RU_y, such as ID_{RU_y}, PS_{RU_y}, and B_{RU_y}, it is not possible for $\mathcal{A}$ to perform valid commutation as mentioned above. Therefore, SRUA-IoT is immune to offline PG attack.*

Proposition 6 *SRUA-IoT is secure against impersonation attack.*

proof 4.6 *SRUA-IoT considers the following three types of impersonation attacks.*

1. *UI attack: Let $\mathcal{A}$ tries to generate an AKE request message $ME_a^a : \{T_x^a, PID^x, G_1^a, A^a uth_{a1}\}$ by selecting T_x^a, and R_1. However, to send an AKE request to RC, $\mathcal{A}$ needs to known both the secret credentials, i.e., TS_{RU_y} and PID_{SN_x}, which are known only to RU_y. Moreover, TS_{RU_y} and PID_{SN_x} are stored in SD_y's memory in the encrypted form. Therefore, SRUA-IoT is secure against UI attack.*

2. *RC impersonation attack: In this case, $\mathcal{A}$ picks R_2^a, T_y^a, and contrives a message $ME_b^a : \{T_y^a, G_2^a, G_3^a, Auth_{a3}^a\}$ to pretend that this messages is from a legitimate RC. However, to generate ME_b^a, $\mathcal{A}$ needs to know the secret parameters, such as TS_{SN_x} and PID_{SN_x}, which are stored in encrypted form. Therefore, without knowing these parameters, $\mathcal{A}$ cannot fabricate a false massage to make SN_x believe that the message is created by a legal RC. Hence, SRUA-IoT is secure against RC impersonation attack.*

3. *SN_x impersonation attack: $\mathcal{A}$ can generate a fake message $ME_c^a : \{T_z^a, G_4^a, Auth_{a5}^a\}$ and send it to RU_y to make RU_y believe that the message is from a legal SN_x. However, to generate a valid ME_c, $\mathcal{A}$ needs to know W_1, R_1, R_2, R_3, and TS_{SN_x}. Without the knowledge of these secret credentials, it is impractical for $\mathcal{A}$ to create a licit message ME_c. Hence, SRUA-IoT is secure against SN_x impersonation attack.*

Proposition 7 *SRUA-IoT is resilient against SN_x capture attack.*

proof 4.7 *In 6LoWPANs, SN_x are deployed in unattended environment. $\mathcal{A}$ can capture an SN_x and can procure the sensitive information stored in the memory of SN_x. Since all the deployed SN_x contain distinct secret information, therefore, by capturing an SN_x $\mathcal{A}$ cannot breach the security of the entire 6LoWPAN. Hence, SRUA-IoT is resilient against SN_x capture attack.*

Proposition 8 *SRUA-IoT is immune to de-synchronization attack.*

proof 4.8 *If the network entities are updating pseudonyms during the execution of the AKE process, $\mathcal{A}$ can establish de-synchronization attack by dropping the captured message. In SRUA-IoT, GW and RU_y update PID^x to PID^{x+1} to accomplish anonymous communication. However, to avoid the de-synchronization attack, both GW and RU_y keep PID^x and PID^{x+1} in their memory. If $\mathcal{A}$ halts the AKE process by dropping the authentication messages, RU_y can use old PID^x for the AKE process. Therefore, SRUA-IoT is immune to de-synchronization attack.*

Proposition 9 *SRUA-IoT is resistant to ESL attack.*

proof 4.9 *Proof In SRUA-IoT, both RU_y and SN_x compute SK as* $SK_{x,y} = H(H(R_1 \parallel R_2 \parallel R_3) \parallel H(R_1 \parallel TS_{RU_y} \parallel PID^x) \parallel T_z \parallel PID_{SN_x})$. *It is obvious that the calculated SK is the concoction of ephemeral (short term) parameters R_1, R_2 and R_3, and long term credential, TS_{RU_y}, PID_{SN_x}, and PID^x. $\mathcal{A}$ needs to compromise both ephemeral and long term credentials to reveal SK. Therefore, SRUA-IoT is resistant to ESL attack.*

Proposition 10 *SRUA-IoT ensures PFS.*

proof 4.10 *From the discussion in Proposition 9, it is clear that SK is the concatenation of fresh ephemeral and long term secret credentials. If $\mathcal{A}$ compromises SK of the previous AKE process but cannot compromise SK of the new AKE processes, then SRUA-IoT renders the PFS feature.*

Proposition 11 *SRUA-IoT ensures secure MA.*

proof 4.11 *In SRUA-IoT, RU_y achieves validation on RC after verifying the condition $Auth_{a1} = Auth_{a2}$. For this condition to hold, the knowledge of credentials GK, ID_G, and TS_{RU_y} is required. To verify the condition at SN_x $Auth_{a3} = Auth_{a4}$, the knowledge of TS_{SN_x} and PID_{SN_x} is necessary. SN_x achieves authentication on SD_{RU_y} by validating the condition $Auth_{a5} = Auth_{a6}$. Therefore, RU_y, SN_x, and GW mutually validate each other to achieve secure mutual authentication.*

4.2 SK security validation using random oracle model

We employ ROM to corroborate SK's security in SRUA-IoT. In ROM, $\mathcal{A}$ consociates with kth instance of a participating entity EN^k, which is involved in executing SRUA-IoT. It can be a legitimate RU_y, GW or SN_x. Therefore, $EN^k_{RU_y}$, EN^k_{GW}, and $EN^k_{SN_x}$ are k_1^{th}, k_2^{th}, and k_3^{th} instances of RU_y, GW, and SN_x, respectively. To simulate real attacks, ROM considers various queries, namely, *Send*, *Test*, *Reveal*, *CorruptSD*, and *Execute*. A description of these queries is presented in S3 Table. Furthermore, SHA is modeled as a random oracle HR ($|HR|$ specifies the rage space of SHA output) and it is available for all SRUA-IoT executing entities including $\mathcal{A}$. By using the queries presented in S3 Table, the security of SK is proved in Theorem 4.12.

Theorem 4.12 *Suppose a polynomial-time $\mathcal{A}$ is running against the proposed SRUA-IoT in time T_i. If QR_h denotes the hash quires, $|HR|$ specifies the range space of SHA output, SQ_s indicates the send queries, lbk defines the length of β_k key, and $|PD|$ refers to the password dictionary, the approximated advantage of $\mathcal{A}$ in breaching the security of SRUA-IoT for procuring SK between RU_y and SN_x can be defined as*

$$AD_{\mathcal{A}}^{SRUA-IoT}(T_i) \leq \frac{QR_h^2}{|HR|} + \frac{SQ_s}{2^{lbk-1}|PD|}. \quad (1)$$

proof 4.13 *To prove this theorem, we consider the following four games ($GM_x|x = 0, 1, 2, 3$).*

4.2.1 GM_0. *A real security attack is accomplished by $\mathcal{A}$ against SRUA-IoT in GM_0. $\mathcal{A}$ picks c bits at GM_0. Therefore, we can procure*

$$AD^{\mathcal{A}}_{SRUA-IoT}(T_i) = |2.AD^{\mathcal{A},GM_0}_{SRUA-IoT} - 1|. \quad (2)$$

4.2.2 GM_1. *In GM_1, $\mathcal{A}$ effectuates an eavesdropping attack and captures all the exchanged messages ME_a:{T_x, PID^x, G_1, $Auth_{a1}$}, ME_b:{T_y, G_2, G_3, $Auth_{a3}$}, and ME_c:{T_z, G_4, $Auth_{a5}$} during the AKE process of SRUA-IoT by utilizing the execute query defined in S3 Table. To procure SK, $\mathcal{A}$ executes the Reveal and Test queries and checks if the return key is a random string or real key at the completion of GM_1. The constructed SK between RU_y and SN_x is* $SK_{x,y} = H(H(R_1 \parallel R_2 \parallel R_3) \parallel H(R_1 \parallel TS_{RU_y} \parallel PID^x) \parallel T_z \parallel PID_{SN_x})$. *$\mathcal{A}$ needs to know all the*

long-term secrets and other ephemeral numbers, which are known only to RU_y, SN_x, and RC. Hence by executing the eavesdropping attack, the chance of $\mathcal{A}$ to win the game will not be enhanced. Therefore, it is evident that

$$AD^{\mathcal{A},GM_1}_{SRUA-IoT} = AD^{\mathcal{A},GM_0}_{SRUA-IoT}. \tag{3}$$

4.2.3 GM_2. *In GM_2, $\mathcal{A}$ performs an active attack by simulating Send and Hash quires. All the exchanged messages ME_a, ME_b, and ME_c are protected using the collision resistance SHA function. The communicated message incorporates random number, timestamps, secret identities, and TSs. Therefore, no SHA collision will occur when $\mathcal{A}$ effectuates the Send and Hash quarries. By birthday paradox, the following can be achieved.*

$$|AD^{\mathcal{A},GM_1}_{SRUA-IoT} - AD^{\mathcal{A},GM_2}_{SRUA-IoT}| \leq QR_h^2/(2|HR|). \tag{4}$$

4.2.4 GM_3. *This game effectuates the simulation of CorruptSD query. Typically, RU_y picks low-entropy passwords. By utilizing the password dictionary attack, $\mathcal{A}$ tries to guess the password of RU_y after procuring the information stored on SD_y, including {CT_{lo}, $Auth_y$, Rp, Gen(.), Rep(.), Et}. $\mathcal{A}$ also attempts to guess β_k from the information stored on SD_y. SRUA-IoT employs robust FE that generates highly random $\beta_k \in [0, 1]^{lbk}$, where lbk is the length of β_k. The probability of guessing β_k is nearly $\frac{1}{2^{lbk}}$. Furthermore, in the communication system, only a limited number of wrong password attempts are allowed. Under these conditions, we have*

$$|AD^{\mathcal{A},GM_2}_{SRUA-IoT} - AD^{\mathcal{A},GM_3}_{SRUA-IoT}| \leq \frac{SQ_s}{2^{lbk}|PD|}. \tag{5}$$

After executing the above queries, $\mathcal{A}$ needs to guess bit c upon executing the Test query. Therefore, we have $AD^{\mathcal{A},GM_3}_{SRUA-IoT} = \frac{1}{2}$.

By utilizing the triangular inequality and simplifying (2)–(5), the following is achieved:

$$\begin{aligned}
\frac{1}{2}AD^{\mathcal{A}}_{SRUA-IoT}(T_i) &= |AD^{\mathcal{A},GM_3}_{SRUA-IoT} = \frac{1}{2}| \\
&= |AD^{\mathcal{A},GM_1}_{SRUA-IoT} - AD^{\mathcal{A},GM_3}_{SRUA-IoT}| \\
&\leq |AD^{\mathcal{A},GM_1}_{SRUA-IoT} - AD^{\mathcal{A},GM_2}_{SRUA-IoT}| \\
&\quad + |AD^{\mathcal{A},GM_2}_{SRUA-IoT} - AD^{\mathcal{A},GM_3}_{SRUA-IoT}| \\
&\leq \frac{QR_h^2}{2|HR|} + \frac{SQ_s}{2^{lbk}|PD|}.
\end{aligned} \tag{6}$$

Hence, we get

$$AD^{\mathcal{A}}_{SRUA-IoT}(T_i) \leq \frac{QR_h^2}{|HR|} + \frac{SQ_s}{2^{lbk-1}|PD|}. \tag{7}$$

4.3 Scyther analysis

We employ the well-known formal security validation tool, called Scyther [68], to validate the security properties and correctness of the proposed SRUA-IoT scheme. To that end, the

security protocol description language (SPDL) is utilized to specify SRUA-IoT by employing the operational semantics ascertained in [68]. S4 Fig demonstrates that proclaims are satisfied, which are specified in the SPDL script. In S4 Fig, SRUA-IoT is the name of the devised protocol with the initiator *RU* and *RC* as the helper node and *SN* as the responder. The descriptions of Nisynch and secrecy are provided in [68]. Secrecy signifies that specific information is not disclosed to any attacker, even when the information is exchanged over a public network. Furthermore, Nisynch describes that any claim defined in the devised protocol specification will also appear in the trace. Moreover, SRUA-IoT analysis illustrates that the supplementary security characteristics produced by Scyther, namely, weak agreement (Weakagree), aliveness (Alive), and non-injective agreement (Niagree) are validated.

5 Performance evaluation

In this section, the performance of SRUA-IoT is compared with Park *et al.* [69], Shuai *et al.* [36], Das *et al.* [30], Shin *et al.* [31], Challa *et al.* [22], Srinivas *et al.* [33], Wazid *et al.* [35], and Chen *et al.* [27] in terms of computational cost, communication cost, security features, and storage cost. We use C/C^{++} based cryptographic library MIRACL and Raspberry PI-3 (RPI-3B) with Quad-core @1.2 GHz, 1BG of RAM, and Ubuntu 16.04 LTS for implementing the proposed SRUA-IoT and the relevant AKE schemes.

5.1 Security features

The proposed SRUA-IoT is compared with the relevant AKE scheme in terms of security functionalities and resistance against various attacks. S4 Table exhibits that Park *et al.* [69] is unprotected against UA, SSC, and PT attacks, Shuai *et al.* [36] is unsafe against de-synchronization attacks, Das *et al.* [30] cannot withstand SSC, PI, and UA attacks and does not ensure SK security, Shin *et al.* [31] is insecure against de-synchronization attack and does not provide revocation phase, Challa *et al.* [22] cannot withstand PI, SSC, UA, PG, and UI attacks, Srinivas *et al.* [33] fails to protect against UI, PI, and SSC attacks, Wazid *et al.* [35] is unsafe against UI, PI, and SSC attacks, and Chen *et al.* [27] cannot protect PI, PG, UA, UI, replay and DoS attacks and also does not ensure mutual authentication. Contrarily, SRUA-IoT is secure as compared to the relevant eminent AKE schemes, as shown in S4 Table.

5.2 Computational cost

In this subsection, the approximated computational overhead of SRUA-IoT and relevant AKE schemes is determined by using computational time of various cryptographic primitives presented in S5 Table. SRUA-IoT has a computational cost of $19T_{SA} + 2T_{ED} + T_{\beta_k} \approx 6.901$ ms, which is less than the benchmark schemes, as shown in S5 Fig and S6 Table. SRUA-IoT has 53.09%, 23.88%, 44.23%, 29.56%, 22.04%, 76.41%, 24.07%, and 38.93% less computational cost as compared to Park *et al.* [69], Shuai *et al.* [36], Das *et al.* [30], Shin *et al.* [31], Srinivas *et al.* [33], Challa *et al.* [22], Wazid *et al.* [35], and Chen *et al.* [27], respectively. Furthermore, SRUA-IoT has a computational overhead of $5T_{SA} \approx 1.275$ms at SN_x, which is less than the benchmark AKE schemes, as shown in S6 Fig and S6 Table. The computational overhead at *GW* increases with the number of users accessing the network resources. S7 Fig shows that SRUA-IoT requires low computational overhead while processing multiple AKE requests simultaneously.

Although the security of SRUA-IoT is verified through formal and informal analyses in Section 4 where the scheme has been shown to resist various covert security attacks, however, an attack or some unexpected event can halt the execution of SRUA-IoT, which may occur at any

step of the AKE phase. Under a specific attack, the execution time can be computed as

$$T_{atp} = \frac{\sum_{i}^{100} T_i}{(1 - \text{attack success probability})}, \quad (8)$$

where T_i denotes time required to accomplish the AKE phase and $\sum_{i}^{100} T_i$ denotes the average time, which is procured after running SRUA-IoT 100 times, and T_{atp} denotes the execution time required to complete the AKE phase under successful attack probability. S8 Fig demonstrates the time utilization of SRUA-IoT and other related schemes with attack success probability. Under various successful attack attempts, SRUA-IoT requires less time to complete its execution than the related AKE schemes.

5.3 Communication cost

The comparative analysis of communication cost is illustrated in this subsection. For SRUA-IoT, the size of timestamp is 32 bits, *ECC* point is 160 bits, SHA output size is 160 bits, random number size is 80 bits, different *PID* size is 80 bits, and AES key size is 192 bits. During the execution of the AKE phase, SRUA-IoT exchanges three message, namely, ME_a: $\{T_x, PID^x, G_1, Auth_{a1}\}$, ME_b: $\{T_y, G_2, G_3, Auth_{a3}$ and ME_c: $\{T_z, G_4, Auth_{a5}\}$, of length $\{32 + 80 + 160 + 160\} = 432$ bits, $\{32 + 160 + 160 + 160\} = 512$ bits, and $\{32 + 160 + 160\} = 412$ bits, respectively. The aggregated communication overheads of SRUA-IoT is 1356 bits. S7 Table and S9 Fig demonstrate the comparison of SRUA-IoT and other related AKE schemes. SRUA-IoT has 75.92%, 21.53%, 11.72%, 29.28%, 46.36%, 11.72%, 20.05%, and 57.2% less communication cost as compared to Park *et al.* [69], Shuai *et al.* [36], Das *et al.* [30], Shin *et al.* [31], Challa *et al.* [22], Srinivas *et al.* [33], and Chen *et al.* [27], respectively.

5.4 Storage cost

This subsection provides the storage cost comparison of SRUA-IoT with other AKE schemes. In SRUA-IoT, RU_y, GW, and SN_x store $\{CT_{lo}, Auth_y, Rp, Gen(.), Rep(.), Et\}$, $\{PID^{x+1}, PID^x, RP_{RU_y}, CT_{RU_y}\}$, and $\{PID_{SN_x}, TS_{SN_x}\}$ of length $\{240 + 160 + 160 + 8\} = 568$ bits, $\{80 + 80 + 80 + 240\} = 480$ bits, and $\{80 + 80\} = 160$ bits, respectively. The total storage overhead can be calculated as $\{568 + 480 + 160\} = 1208$ bits. Besides, the storage costs of Park *et al.* [69], Shuai *et al.* [36], Das *et al.* [30], Shin *et al.* [31], Challa *et al.* [22], Srinivas *et al.* [33], Wazid *et al.* [35], and Chen *et al.* [27] are 1600 bits, 1776 bits, 3738 bits, 1160 bits, 4016 bits, 2888 bits, 4126 bits, and 1792 bits, respectively. SRUA-IoT has a slightly higher storage cost as compared to Shin *et al.* [31]. However, SRUA-IoT has less computational and communication cost during the AKE phase in contrast to Shin *et al.* [31]. S10 Fig illustrates the storage cost comparison of SRUA-IoT and the related AKE schemes.

6 Conclusion

Information security is critical in resource-constricted 6LoWPAN-based IoT networks. This paper has presented an AKE scheme called SRUA-IoT for resource-constricted 6LoWPAN devices to validate the legitimacy of remote users interacting in real-time with sensor nodes deployed in smart home networks. The scheme performs user authorization before procuring real-time data from sensors by employing a lightweight secure hash algorithm (SHA-160) and an advanced encryption standard (AES-192) to accomplish the AKE process. The proposed scheme is corroborated both formally and informally to explicate its resistance against various malicious security vulnerabilities. Moreover, numerical results in comparison with benchmarks reveal that SRUA-IoT requires low computational and communication resources in

6LoWPANs to accomplish the AKE phase. Our future work will explore authenticated encryption with associated data to devise a resource-efficient AKE scheme with reduced computational cost for resource-constricted IoT devices.

Supporting information

S1 Fig. The user AKE phase of SRUA-IoT.
(TIF)

S2 Fig. Password change phase.
(TIF)

S3 Fig. Revocation phase.
(TIF)

S4 Fig. Scyther results.
(TIF)

S5 Fig. Comparison of total computation cost required to complete the AKE process.
(TIF)

S6 Fig. Computational overhead at SN_x side.
(TIF)

S7 Fig. Computational delay at *GW* with increasing number of users.
(TIF)

S8 Fig. Computational overhead with attack success probability.
(TIF)

S9 Fig. Communication overhead in the network with increasing number of users.
(TIF)

S10 Fig. Comparison of storage costs.
(TIF)

S1 Table. Comparative analysis of eminent AKE schemes [21–41].
(PDF)

S2 Table. List of key notations.
(PDF)

S3 Table. Description of different ROM queries.
(PDF)

S4 Table. Comparison of security features [22, 30, 31, 33, 35, 36, 69].
(PDF)

S5 Table. Experimental computational cost of various cryptographic operations.
(PDF)

S6 Table. Comparison of computational costs [22, 27, 30, 31, 33, 35, 36, 69].
(PDF)

S7 Table. Comparison of communication costs [22, 27, 30, 31, 33, 35, 36, 69].
(PDF)

S1 Appendix.
(ZIP)

Author Contributions

Data curation: Ghulam Abbas.

Investigation: Muhammad Tanveer.

Methodology: Ziaul Haq Abbas.

Project administration: Thar Baker.

Software: Muhammad Waqas.

Supervision: Dhiya Al-Jumeily OBE.

References

1. Raja SP, Sampradeepraj T. Internet of things: A research-oriented introductory. International Journal of Ad Hoc and Ubiquitous Computing. 2018; 29(1-2):4–14. https://doi.org/10.1504/IJAHUC.2018.10015646
2. Liu R, Weng Z, Hao S, Chang D, Bao C, Li X. Addressless: Enhancing IoT Server Security Using IPv6. IEEE Access. 2020; 8:90294–90315. https://doi.org/10.1109/ACCESS.2020.2993700
3. Thubert P. 6LoWPAN selective fragment recovery. IETF, Internet-Draft–work in progress 05. 2019;.
4. Verma A, Ranga V. Security of RPL based 6LoWPAN Networks in the Internet of Things: A Review. IEEE Sensors Journal. 2020; 20(11):5666–5690. https://doi.org/10.1109/JSEN.2020.2973677
5. Oliveira LML, Rodrigues JJ, de Sousa AF, Denisov VM. Network admission control solution for 6LoWPAN networks based on symmetric key mechanisms. IEEE Transactions on Industrial Informatics. 2016; 12(6):2186–2195. https://doi.org/10.1109/TII.2016.2601562
6. Glissa G, Meddeb A. 6LowPSec: An end-to-end security protocol for 6LoWPAN. Ad Hoc Networks. 2019; 82:100–112. https://doi.org/10.1016/j.adhoc.2018.01.013
7. Meneghello F, Calore M, Zucchetto D, Polese M, Zanella A. IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. IEEE Internet of Things Journal. 2019; 6(5):8182–8201. https://doi.org/10.1109/JIOT.2019.2935189
8. Khan AU, Abbas G, Abbas ZH, Tanveer M, Ullah S, Naushad A. HBLP: A Hybrid Underlay-Interweave Mode CRN for the Future 5G-Based Internet of Things. IEEE Access. 2020; 8:63403–63420. https://doi.org/10.1109/ACCESS.2020.2981413
9. Tanveer M, Abbas G, Abbas ZH, Bilal M, Mukherjee A, Kwak KS. LAKE-6SH: Lightweight User Authenticated Key Exchange for 6LoWPAN-based Smart Homes. IEEE Internet of Things Journal. 2021; p. 1–1. https://doi.org/10.1109/JIOT.2021.3085595
10. Tanveer M, Abbas G, Abbas ZH, Waqas M, Muhammad F, Kim S. S6AE: Securing 6LoWPAN Using Authenticated Encryption Scheme. Sensors. 2020; 20(9):2707. https://doi.org/10.3390/s20092707 PMID: 32397469
11. Oliveira LM, Rodrigues JJ, De Sousa AF, Lloret J. A network access control framework for 6LoWPAN networks. Sensors. 2013; 13(1):1210–1230. https://doi.org/10.3390/s130101210 PMID: 23334610
12. Tomić I, McCann JA. A survey of potential security issues in existing wireless sensor network protocols. IEEE Internet of Things Journal. 2017; 4(6):1910–1923. https://doi.org/10.1109/JIOT.2017.2749883
13. Mavani M, Asawa K. Resilient against spoofing in 6LoWPAN networks by temporary-private IPv6 addresses. Peer-to-Peer Networking and Applications. 2020; 13(1):333–347. https://doi.org/10.1007/s12083-019-00792-6
14. Alloghani M, Alani MM, Al-Jumeily D, Baker T, Mustafina J, Hussain A, et al. A systematic review on the status and progress of homomorphic encryption technologies. Journal of Information Security and Applications. 2019; 48:102362. https://doi.org/10.1016/j.jisa.2019.102362
15. Al-Maytami BA, Fan P, Hussain AJ, Baker T, Liatsis P. An efficient queries processing model based on multi broadcast searchable keywords encryption (mbske). Ad Hoc Networks. 2020; 98:102028. https://doi.org/10.1016/j.adhoc.2019.102028
16. Baker T, Asim M, MacDermott Á, Iqbal F, Kamoun F, Shah B, et al. A secure fog-based platform for SCADA-based IoT critical infrastructure. Software: Practice and Experience. 2020; 50(5):503–518.

17. Baker T, Mackay M, Shaheed A, Aldawsari B. Security-Oriented Cloud Platform for SOA-Based SCADA. In: 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing; 2015. p. 961–970.

18. Tanveer M, Zahid AH, Ahmad M, Baz A, Alhakami H. LAKE-IoD: Lightweight authenticated key exchange protocol for the Internet of Drone environment. IEEE Access. 2020; 8:155645–155659. https://doi.org/10.1109/ACCESS.2020.3019367

19. Zhang Y, He D, Li L, Chen B. A lightweight authentication and key agreement scheme for Internet of Drones. Computer Communications. 2020;. https://doi.org/10.1016/j.comcom.2020.02.067

20. Tanveer M, Khan AU, Kumar N, Hassan MM. RAMP-IoD: A Robust Authenticated Key Management Protocol for the Internet of Drones. IEEE Internet of Things Journal. 2021; p. 1–1. https://doi.org/10.1109/JIOT.2021.3084946

21. Qiu Y, Ma M. A mutual authentication and key establishment scheme for M2M communication in 6LoWPAN networks. IEEE Transactions on Industrial Informatics. 2016; 12(6):2074–2085. https://doi.org/10.1109/TII.2016.2604681

22. Challa S, Wazid M, Das AK, Kumar N, Reddy AG, Yoon EJ, et al. Secure signature-based authenticated key establishment scheme for future IoT applications. IEEE Access. 2017; 5:3028–3043. https://doi.org/10.1109/ACCESS.2017.2676119

23. Vijayakumar P, Azees M, Kannan A, Jegatha Deborah L. Dual Authentication and Key Management Techniques for Secure Data Transmission in Vehicular Ad Hoc Networks. IEEE Transactions on Intelligent Transportation Systems. 2016; 17(4):1015–1028. https://doi.org/10.1109/TITS.2015.2492981

24. Jung J, Moon J, Lee D, Won D. Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. Sensors. 2017; 17(3):644. https://doi.org/10.3390/s17030644 PMID: 28335572

25. Qi M, Chen J. An efficient two-party authentication key exchange protocol for mobile environment. International Journal of Communication Systems. 2017; 30(16):e3341. https://doi.org/10.1002/dac.3341

26. Chaudhry SA, Naqvi H, Khan MK. An enhanced lightweight anonymous biometric based authentication scheme for TMIS. Multimedia Tools and Applications. 2018; 77(5):5503–5524. https://doi.org/10.1007/s11042-017-4464-9

27. Chen Y, López L, Martínez JF, Castillejo P. A lightweight privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: LightPriAuth. Journal of Sensors. 2018; 2018:1–16.

28. Amin R, Islam SH, Biswas G, Khan MK, Kumar N. A robust and anonymous patient monitoring system using wireless medical sensor networks. Future Generation Computer Systems. 2018; 80:483–495. https://doi.org/10.1016/j.future.2016.05.032

29. Das AK, Wazid M, Yannam AR, Rodrigues JJ, Park Y. Provably secure ECC-based device access control and key agreement protocol for IoT environment. IEEE Access. 2019; 7:55382–55397. https://doi.org/10.1109/ACCESS.2019.2912998

30. Das AK, Wazid M, Kumar N, Vasilakos AV, Rodrigues JJ. Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment. IEEE Internet of Things Journal. 2018; 5(6):4900–4913. https://doi.org/10.1109/JIOT.2018.2877690

31. Shin S, Kwon T. A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes. Sensors. 2019; 19(9):2012. https://doi.org/10.3390/s19092012 PMID: 31035690

32. Lu Y, Xu G, Li L, Yang Y. Robust privacy-preserving mutual authenticated key agreement scheme in roaming service for global mobility networks. IEEE Systems Journal. 2019; 13(2):1454–1465. https://doi.org/10.1109/JSYST.2018.2883349

33. Srinivas J, Das AK, Kumar N, Rodrigues JJ. TCALAS:Temporal credential-based anonymous lightweight authentication scheme for Internet of Drones environment. IEEE Transactions on Vehicular Technology. 2019; 68(7):6903–6916. https://doi.org/10.1109/TVT.2019.2911672

34. Vinoth R, Deborah LJ, Vijayakumar P, Kumar N. Secure Multifactor Authenticated Key Agreement Scheme for Industrial IoT. IEEE Internet of Things Journal. 2021; 8(5):3801–3811. https://doi.org/10.1109/JIOT.2020.3024703

35. Wazid M, Das AK, Kumar N, Vasilakos AV, Rodrigues JJ. Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of Drones deployment. IEEE Internet of Things Journal. 2018; 6(2):3572–3584. https://doi.org/10.1109/JIOT.2018.2888821

36. Shuai M, Yu N, Wang H, Xiong L. Anonymous authentication scheme for smart home environment with provable security. Computers & Security. 2019; 86:132–146. https://doi.org/10.1016/j.cose.2019.06.002

37. Barman S, Shum HP, Chattopadhyay S, Samanta D. A secure authentication protocol for multi-server-based e-healthcare using a fuzzy commitment scheme. IEEE Access. 2019; 7:12557–12574. https://doi.org/10.1109/ACCESS.2019.2893185

38. Singh J, Gimekar A, Venkatesan S. An efficient lightweight authentication scheme for human-centered industrial Internet of Things. International Journal of Communication Systems. 2019; p. e4189. https://doi.org/10.1002/dac.4189

39. Sadhukhan D, Ray S, Biswas G, Khan M, Dasgupta M. A lightweight remote user authentication scheme for IoT communication using elliptic curve cryptography. The Journal of Supercomputing. 2021; 77(2):1114–1151. https://doi.org/10.1007/s11227-020-03318-7

40. Ali Z, Chaudhry SA, Ramzan MS, Al-Turjman F. Securing Smart City Surveillance: A Lightweight Authentication Mechanism for Unmanned Vehicles. IEEE Access. 2020; 8:43711–43724. https://doi.org/10.1109/ACCESS.2020.2977817

41. Tanveer M, Abbas G, Abbas ZH. LAS-6LE: A Lightweight Authentication Scheme for 6LoWPAN Environments. In: 2020 14th International Conference on Open Source Systems and Technologies (ICOSST). IEEE; 2020. p. 1–6.

42. Vijayakumar P, Azees M, Deborah LJ. CPAV: Computationally efficient privacy preserving anonymous authentication scheme for vehicular ad hoc networks. In: 2015 IEEE 2nd international conference on cyber security and cloud computing. IEEE; 2015. p. 62–67.

43. Vijayakumar P, Obaidat MS, Azees M, Islam SH, Kumar N. Efficient and secure anonymous authentication with location privacy for IoT-based WBANs. IEEE Transactions on Industrial Informatics. 2019; 16 (4):2603–2611. https://doi.org/10.1109/TII.2019.2925071

44. Azees M, Vijayakumar P, Karuppiah M, Nayyar A. An efficient anonymous authentication and confidentiality preservation schemes for secure communications in wireless body area networks. Wireless Networks. 2021; 27(3):2119–2130. https://doi.org/10.1007/s11276-021-02560-y

45. Azees M, Pandi V, Lazarus JD, Karuppiah M, Christo MS. BBAAS: Blockchain-Based Anonymous Authentication Scheme for Providing Secure Communication in VANETs. Security and Communication Networks. 2021; 2021.

46. Vijayakumar P, Azees M, Chang V, Deborah J, Balusamy B. Computationally efficient privacy preserving authentication and key distribution techniques for vehicular ad hoc networks. cluster computing. 2017; 20(3):2439–2450. https://doi.org/10.1007/s10586-017-0848-x

47. Mishra D, Vijayakumar P, Sureshkumar V, Amin R, Islam SH, Gope P. Efficient authentication protocol for secure multimedia communications in IoT-enabled wireless sensor networks. Multimedia Tools and Applications. 2018; 77(14):18295–18325. https://doi.org/10.1007/s11042-017-5376-4

48. Wei F, Vijayakumar P, Jiang Q, Zhang R. A Mobile Intelligent Terminal Based Anonymous Authenticated Key Exchange Protocol for Roaming Service in Global Mobility Networks. IEEE Transactions on Sustainable Computing. 2020; 5(2):268–278. https://doi.org/10.1109/TSUSC.2018.2817657

49. Wei F, Vijayakumar P, Kumar N, Zhang R, Cheng Q. Privacy-Preserving Implicit Authentication Protocol Using Cosine Similarity for Internet of Things. IEEE Internet of Things Journal. 2021; 8(7):5599–5606. https://doi.org/10.1109/JIOT.2020.3031486

50. Ali Z, Hussain S, Rehman RHU, Munshi A, Liaqat M, Kumar N, et al. ITSSAKA-MS: An Improved Three-Factor Symmetric-Key Based Secure AKA Scheme for Multi-Server Environments. IEEE Access. 2020; 8:107993–108003. https://doi.org/10.1109/ACCESS.2020.3000716

51. Das AK, Kumar N, Alazab M, et al. Designing Authenticated Key Management Scheme in 6G-enabled Network in a Box Deployed for Industrial Applications. IEEE Transactions on Industrial Informatics. 2020;.

52. Park K, Park Y, Park Y, Das AK. 2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment. IEEE Access. 2018; 6:30225–30241. https://doi.org/10.1109/ACCESS.2018.2844190

53. Gao L, Zhang L, Feng L, Ma M. An Efficient Secure Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN in Unattended Scenarios. Wireless Personal Communications. 2020[in press]; p. 1–19.

54. Abbas N, Asim M, Tariq N, Baker T, Abbas S. A mechanism for securing IoT-enabled applications at the fog layer. Journal of Sensor and Actuator Networks. 2019; 8(1):16. https://doi.org/10.3390/jsan8010016

55. Ali W, Abbas G, Abbas ZH. Joint Sybil Attack Prevention and Energy Conservation in Wireless Sensor Networks. In: 2019 International Conference on Frontiers of Information Technology (FIT). IEEE; 2019. p. 179–1795.

56. Banerjee S, Odelu V, Das AK, Chattopadhyay S, Park Y. An Efficient, Anonymous and Robust Authentication Scheme for Smart Home Environments. Sensors. 2020; 20(4):1215. https://doi.org/10.3390/s20041215 PMID: 32098448

57. Qiu Y, Ma M. Secure group mobility support for 6lowpan networks. IEEE Internet of Things Journal. 2018; 5(2):1131–1141. https://doi.org/10.1109/JIOT.2018.2805696

58. Alzahrani BA, Chaudhry SA, Barnawi A, Al-Barakati A, Alsharif MH. A privacy preserving authentication scheme for roaming in IoT-based wireless mobile networks. Symmetry. 2020; 12(2):287. https://doi.org/10.3390/sym12020287

59. Chaudhry SA, Yahya K, Al-Turjman F, Yang MH. A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems. IEEE Access. 2020; 8:139244–139254. https://doi.org/10.1109/ACCESS.2020.3012121

60. Lee H, Kang D, Ryu J, Won D, Kim H, Lee Y. A three-factor anonymous user authentication scheme for Internet of Things environments. Journal of Information Security and Applications. 2020; 52:102494. https://doi.org/10.1016/j.jisa.2020.102494

61. Ali Z, Ghani A, Khan I, Chaudhry SA, Islam SH, Giri D. A robust authentication and access control protocol for securing wireless healthcare sensor networks. Journal of Information Security and Applications. 2020; 52:102502. https://doi.org/10.1016/j.jisa.2020.102502

62. Liu CH, Chung YF. Secure user authentication scheme for wireless healthcare sensor networks. Computers & Electrical Engineering. 2017; 59:250–261. https://doi.org/10.1016/j.compeleceng.2016.01.002

63. Wang D, Cheng H, Wang P, Huang X, Jian G. Zipf's Law in Passwords. IEEE Transactions on Information Forensics and Security. 2017; 12(11):2776–2791. https://doi.org/10.1109/TIFS.2017.2721359

64. Dey S, Hossain A. Session-key establishment and authentication in a smart home network using public key cryptography. IEEE Sensors Letters. 2019; 3(4):1–4. https://doi.org/10.1109/LSENS.2019.2905020

65. Kumar P, Gurtov A, Iinatti J, Ylianttila M, Sain M. Lightweight and secure session-key establishment scheme in smart home environments. IEEE Sensors Journal. 2015; 16(1):254–264. https://doi.org/10.1109/JSEN.2015.2475298

66. Majumder S, Ray S, Sadhukhan D, Khan MK, Dasgupta M. ECC-CoAP: Elliptic Curve Cryptography Based Constraint Application Protocol for Internet of Things. Wireless Personal Communications. 2020; p. 1–30.

67. Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory. 1983; 29(2):198–208. https://doi.org/10.1109/TIT.1983.1056650

68. Cremers CJ. The Scyther Tool: Verification, falsification, and analysis of security protocols. In: International Conference on Computer Aided Verification. Springer; 2008. p. 414–418.

69. Park Y, Park Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors. 2016; 16(12):2123. https://doi.org/10.3390/s16122123 PMID: 27983616