Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Lightweight authentication scheme for edge control systems in Industrial Internet of Things*

Wenli SHANG[1], Xudong WEN[1], Zhuo CHEN[1], Wenze XIONG[2], Zhiwei CHANG[1], Zhong CAO†‡[1]

[1]*School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China*

[2]*Instrumentation Technology and Economy Institute, Beijing 100055, China*

†E-mail: zhongc@gzhu.edu.cn

Received June 9, 2024; Revision accepted June 16, 2024; Crosschecked Sept. 29, 2024; Published online Nov. 5, 2024

Abstract: In edge control systems (ECSs), edge computing demands more local data processing power, while traditional industrial programmable logic controllers (PLCs) cannot meet this demand. Thus, edge intelligent controllers (EICs) have been developed, making their secure and reliable operation crucial. However, as EICs communicate sensitive information with resource-limited terminal devices (TDs), a low-cost, efficient authentication solution is urgently needed since it is challenging to implement traditional asymmetric cryptography on TDs. In this paper, we design a lightweight authentication scheme for ECSs using low-computational-cost hash functions and exclusive OR (XOR) operations; this scheme can achieve bidirectional anonymous authentication and key agreement between the EIC and TDs to protect the privacy of the devices. Through security analysis, we demonstrate that the authentication scheme can provide the necessary security features and resist major known attacks. Performance analysis and comparisons indicate that the proposed authentication scheme is effective and feasible for deployment in ECSs.

Key words: Edge intelligent controller (EIC); Edge control systems (ECSs); Terminal devices (TDs); Anonymous authentication; Lightweight authentication

https://doi.org/10.1631/FITEE.2400497 **CLC number:** TP309

1 Introduction

The Internet of Things (IoT) is an infrastructure that uses sensing technology and network communication to connect everything and to provide services such as data sensing, transmission, and processing (Khan et al., 2023). As digitalization and intelligence continue to transform the economy and society, IoT has become a major driving force for economic and social transformation and upgrading. The Industrial Internet of Things (IIoT) is an important and influential branch of IoT that empowers intelligent production and manufacturing (Sisinni et al., 2018). By using intelligent sensing technology, IIoT can sense the factory environment and each link in real time, collect manufacturing data, and then send it to the corresponding devices for intelligent decision analysis through network communication technology. The results of decision analysis can improve production efficiency, production quality, cost reduction, and environmental sustainability (Zhou et al., 2019).

The data collected in the IIoT environment need to be stored, processed, and analyzed accordingly, and currently most enterprises handle industrial data in a traditional model based on cloud computing (Liu et al., 2022). The cloud computing system has superb storage and computing capabilities to process

‡ Corresponding author

* Project supported by the National Key R&D Program of China (No. 2021YFB2012400), the National Natural Science Foundation of China (No. 62173101), the Basic and Applied Basic Research Funding of Guangdong Province, China (Nos. 2022A1515011558 and 2022A1515010865), and the Key Laboratory of On-Chip Communication and Sensor Chip of Guangdong Higher Education Institutes, China (No. 2023KSYS002)

ORCID: Zhong CAO, https://orcid.org/0000-0002-2301-8030

and analyze large amounts of data, but centralized processing of large amounts of data will lead to severe network congestion, while long-distance transmission will lead to delay and high energy consumption (Gadekallu et al., 2022). To cope with the limitations of cloud computing, edge computing for real-time data storage and computation closer to the object or data source has emerged (Nkenyereye et al., 2021). Compared to cloud computing, edge computing has low latency and rapid response, which can significantly improve data processing efficiency. Edge computing has contributed to the development and improvement of IIoT, which can optimize industrial production processes for efficient and sustainable production (Zhang Y and Wei, 2021). In recent years, research on edge-computing environments has focused on improving communication resource efficiency and system responsiveness. For example, Xiao et al. (2024a, 2024b) explored the communication problems between multiple high-speed trains and grid supplemental services, respectively. They achieved intelligent resource scheduling and co-design by constructing an event-based finite-state machine framework and a bandwidth-aware event communication mechanism, which improves communication efficiency and resource utilization.

In the field of industrial control, edge computing places higher demands on the local processing power of data. However, the traditional industrial programmable logic controller (PLC) cannot perform such tasks. The edge intelligent controller (EIC) was born in response to the needs of edge computing (Sodhro et al., 2019). An EIC can integrate multidomain functions, such as PLC, personal computer (PC), gateway, motion control, fieldbus protocol, input/output (I/O) data acquisition, machine learning (Sharp et al., 2018), and machine vision (Wang et al., 2018), into one unit, and simultaneously achieve motion control, data acquisition, and data processing for terminal devices (TDs). Obviously, the EIC has become a core component in edge computing based edge control systems (ECSs), and its secure and trusted operation has great significance and impact on the development and promotion of edge computing.

TDs in ECSs, such as intelligent instruments and industrial robots, need to communicate sensitive information with the EIC. Therefore, authentication between the EIC and TDs is one of the most fundamental security issues and an integral part of the security defense of ECSs. Incorporating authentication technology in ECSs can effectively prevent malicious attackers from faking as legitimate devices to intrude into ECSs and stealing sensitive data stored inside ECSs. Since the TDs that need to access the EIC have constrained computing and storage resources and traditional asymmetric cryptography based authentication schemes possess high computational cost, in this paper we propose a lightweight authentication scheme to protect ECSs.

Authentication is an important security technique to achieve data protection. Aman et al. (2019) proposed a token-based authentication scheme with a trade-off between dynamic energy and security for IoT. This scheme can implement different security levels according to different security requirements to reduce the resource consumption of the system. Anonymity protects the device's identity information and prevents attackers from tracking, locating, and targeting attacks, but their scheme does not support anonymity. Wazid et al. (2020) designed a security scheme for remote user authentication and key establishment for smart homes; this enables bidirectional authentication of remote users and smart devices. However, this scheme can remain anonymous only to attackers who intercept authentication messages, but not to disguised attackers. Therefore, the anonymity of the scheme is effective only under certain conditions and does not provide complete anonymity protection against all attack types. Sun et al. (2015) designed an identity authentication and key agreement scheme for smart home networks. However, this scheme does not support anonymity, and the identity information of users and servers is transmitted directly over the network without any anonymization processing. If the anonymity of the device is not guaranteed, it may expose the identity information of the device and cause device privacy leakage. On one hand, if an attacker obtains the identity information of these devices, the attacker may launch targeted attacks on the devices, causing incalculable damage to the devices (Cui et al., 2023). On the other hand, since device privacy may be associated with factory privacy, device privacy leakage may lead to factory privacy leakage, causing incalculable economic loss to the factory and threatening the safety of factory personnel (Zhang QY et al., 2023). It is necessary to design anonymous authentication

schemes to ensure the security of ECSs' device identity authentication and to protect device privacy.

In the IIoT scenario, Esposito et al. (2018) designed an authentication scheme for sensor networks to protect data integrity using a group signature technique. Cui et al. (2021) designed a message authentication scheme for the edge-computing scenario in IIoT using a group signature technique and proxy reencryption, which can guarantee data integrity, confidentiality, and anonymity. Cui et al. (2022) proposed a multiauthority attribute based encryption scheme that protects user privacy by anonymizing attributes in authentication. Cao et al. (2023) designed an efficient revocable anonymous authentication scheme for EICs, which can protect the EIC identity information while achieving traceability and efficient revocability for erroneous EICs. However, all four of these solutions use bilinear pairing operations, which are computationally intensive, and result in some time delay. Furthermore, for IIoT devices with limited resources, it is difficult to execute complex bilinear pairing operations quickly.

A lightweight authentication scheme needs to be designed for IIoT devices with constrained resources, such as limited computing and storage capacity. Esfahani et al. (2019) proposed a machine-to-machine (M2M) lightweight authentication protocol, which is designed based only on hash functions and exclusive OR (XOR) operations and possesses low computational cost. However, this authentication protocol cannot support forward secrecy, and if the current preshared key is compromised, it will cause the past session key to be compromised. Zhang LP et al. (2019) designed a lightweight anonymous authentication and key agreement scheme for smart grids, which implements bidirectional authentication between smart meters and service providers. However, this authentication scheme does not have any measure to resist denial-of-service (DoS) attacks, such as adding timestamps, and the timestamp mechanism can be used to resist DoS attacks (Rose and Jayasree, 2019; Xiao et al., 2022). Jan et al. (2021) proposed a lightweight authentication scheme for smart healthcare, which is based only on hash functions and XOR operations and designed to establish a secure session among a wearable device, a gateway, and a remote server. However, authentication in an operational environment has significant limitations because the registration of the wearable devices is done offline and the lack of forward secrecy of this authentication scheme will have an impact on the session key. Ehui et al. (2022) proposed a lightweight scheme for mutual authentication of sensor nodes and gateways, designed with a hash function, symmetric encryption/decryption algorithm, and hash-based message authentication code (HMAC) for resource-constrained devices. However, this scheme does not have forward secrecy or resistance to DoS attacks, which can break the security of authentication once a malicious attacker exploits these two flaws. In conclusion, the authentication schemes mentioned above have some shortcomings in terms of lightweight and security properties. Therefore, we aim to introduce a novel authentication scheme to address these deficiencies. Our new scheme places greater emphasis on security, particularly focusing on anonymity and forward secrecy, to ensure the privacy of user data. To further enhance the lightweight features, we use simplified encryption and decryption techniques to make the authentication process faster and more energy-efficient. This design allows our authentication solutions to excel in resource-constrained environments while ensuring that security is not compromised. We believe that through these improvements, our authentication scheme will make breakthroughs in all aspects, providing a fresh solution for future security authentication.

The main contributions of this paper are summarized as follows:

1. We propose a lightweight authentication scheme that enables bidirectional anonymous authentication between EICs and TDs.

2. Due to the limited resources of ECS devices, our scheme uses only hash functions and XOR operations for mutual authentication. This effectively reduces the burden of computation and storage while providing adequate protection measures.

3. Through security analysis and performance analysis, the security and effectiveness of our authentication scheme are proved. Compared with other authentication schemes, our authentication scheme offers stronger security properties and lower computational cost, and can be well applied to ECSs.

2 System model and security objectives

To more clearly introduce the proposed lightweight authentication scheme for ECSs, in this

section we present the system architecture and identify the threat model and security objectives of this paper.

2.1 System architecture

The system architecture of the proposed lightweight authentication scheme is shown in Fig. 1. There are three types of participants in this authentication scheme, which are TDs, EIC, and trusted server (TS). TDs are resource-constrained devices, such as intelligent instruments and industrial robots, which are responsible for collecting data and executing commands. The EIC can receive data collected by TDs, process and analyze these data, and control TDs to execute commands. The TS is an entity with high security and independence; at the same time, it has powerful storage and computing capabilities to generate the security parameters required by the system in the ECSs and send them to the corresponding entities.

To protect sensitive data, the TD and the EIC need to authenticate each other. The rough interaction flow is as follows:

1. The TS generates security parameters and distributes them to the TD and the EIC.

2. The EIC identifies and sends the encrypted authentication request to the TD. The TD then verifies the EIC's identity.

3. The TD and the EIC negotiate to generate a session key. Subsequently, the TD and the EIC use the session key to encrypt the data and achieve secure communication.

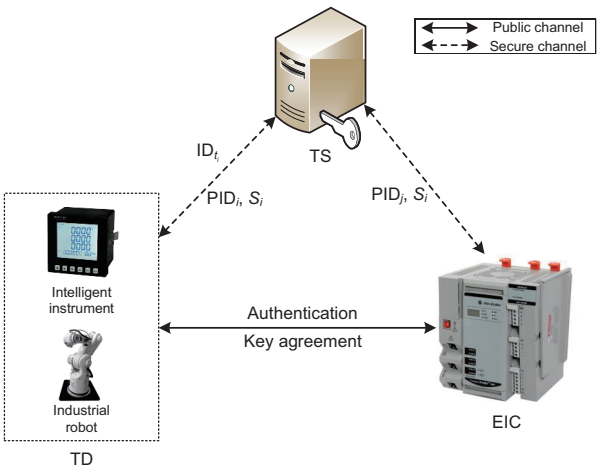

Fig. 1 System architecture

2.2 Threat model

In the proposed authentication scheme, we use the widely accepted Dolev–Yao (DY) threat model (Dolev and Yao, 1983) to make the following assumptions about an adversary's abilities:

1. The communication channel in the registration phase is secure, and the communication channel in the authentication phase is a public channel. An adversary can eavesdrop, intercept, replay, modify, and delete messages on the public channel.

2. The TS is trusted and cannot be captured by adversaries; the latter cannot access or obtain the data stored in the TS.

3. An adversary cannot physically capture the TD or the EIC because these devices are monitored in the factory.

4. An adversary cannot steal the data stored in a secure manner by the EIC and the TD.

2.3 Security objectives

To ensure that the EIC and the TD can operate properly and securely, the main security objectives achieved by our lightweight authentication scheme are as follows:

1. Confidentiality and integrity. Confidentiality means that the secret information between the EIC and the TD must be protected so that an adversary cannot obtain this secret information on the public channel. Integrity refers to detecting whether messages sent between entities have been tampered with, and if modification of a message occurs during mutual authentication, then both parties should immediately discover that the message has been modified.

2. Mutual authentication. To ensure that only legitimate TDs can access the legitimate EIC, mutual authentication between TDs and EICs should be implemented.

3. Anonymity. To protect the privacy of the TD, no third party can obtain the identity information of the TD by intercepting messages, except for the TS (which knows the identity information of the TD).

4. Security features. This authentication scheme provides security features such as forward secrecy and known session key security.

5. Resistance to known attacks. This authentication scheme can resist major known attacks, including resistance to tracking attacks, resistance to impersonation attacks, resistance to man-in-the-

middle (MITM) attacks, resistance to replay attacks, resistance to DoS attacks, resistance to desynchronization attacks, and resistance to stolen-verifier attacks.

3 Proposed scheme

The proposed lightweight authentication scheme consists of two phases: (1) registration phase, in which the TD registers with the TS to obtain the secret, which will be used for mutual authentication with the EIC, while the TS generates its pseudo-identity for the EIC and sends the secret value and its pseudo-identity to the EIC; (2) mutual authentication phase, where the TD and the EIC authenticate each other and generate the session key. The main symbols involved in our authentication scheme and their definitions are shown in Table 1.

Table 1 Main symbols and definitions for the proposed authentication scheme

Symbol	Definition
ID_{t_i}	Identity of TD i at time t
S_i	Secret value created by the TS
PID_i	Pseudo-identity of the TD
PID_j	Pseudo-identity of the EIC
T_i	The i^{th} timestamp, $i = 1, 2, \ldots$
ΔT	Scheduled transmission delay
$h(\cdot)$	Collision-resistant hash function
$\oplus$	XOR operator
$\|\|$	Concatenation operation
SK_{ij}	Session key

TD: terminal device; TS: trusted server; EIC: edge intelligent controller; XOR: exclusive OR

3.1 Registration phase

Each TD must perform a registration process with the TS. Using a secure channel, the TD and the EIC do the following with the TS:

1. The TD transmits its identity ID_{t_i} to the TS via a secure channel.

2. After the TS receives identity ID_{t_i} of the TD, it generates a random number r_s and computes $S_i = h(\mathrm{ID}_{t_i} \,||\, r_\mathrm{s})$. Then, the TS randomly generates a pseudo-identity PID_i for the TD, and sends $\{\mathrm{PID}_i, S_i\}$ to the TD via a secure channel.

3. The TS randomly generates a pseudo-identity PID_j for the EIC, and sends $\{\mathrm{PID}_j, S_i\}$ to the EIC via a secure channel.

4. The TD and the EIC store value S_i in a secure manner.

Fig. 2 shows the steps of this phase.

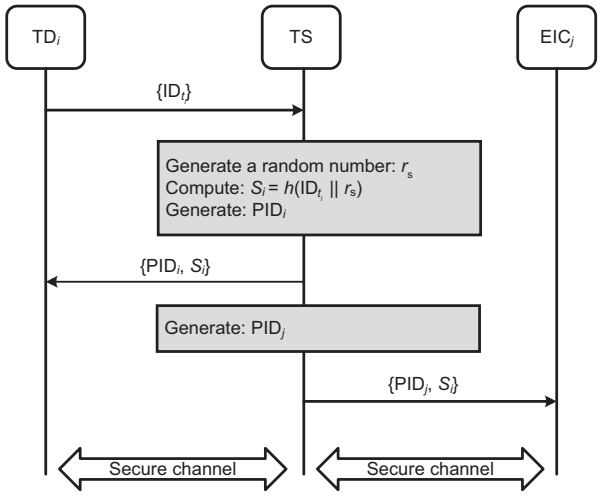

Fig. 2 Registration phase

3.2 Mutual authentication phase

In this phase, the TD and the EIC perform mutual identification and complete authentication. Note that the TD does not use its real identity for authentication.

Therefore, identity ID_{t_i} of the TD cannot be eavesdropped or obtained by a malicious attacker. The authentication process includes the following steps (Fig. 3):

Step 1: The TD generates a random number r_t and a timestamp T_1, computes $C_1 = h(\mathrm{PID}_i \,||\, S_i \,||\, T_1) \oplus r_\mathrm{t}$ and $C_2 = h(r_\mathrm{t} \,||\, T_1 \,||\, S_i)$, and then sends message $M_1 = \{T_1, \mathrm{PID}_i, C_1, C_2\}$ to the EIC.

Step 2: After receiving the authentication request from the TD, the EIC generates a timestamp T_2, verifies $|T_2 - T_1| \leq \Delta T$, where ΔT is the scheduled transmission delay between the TD and the EIC, and terminates the communication immediately if it is not within the delay time. If $r_\mathrm{t} = C_1 \oplus h(\mathrm{PID}_i \,||\, S_i \,||\, T_1)$ is computed within the delay time, it then verifies $C_2 \stackrel{?}{=} h(r_\mathrm{t} \,||\, T_1 \,||\, S_i)$: if the verification is false, the authentication request is rejected and the EIC terminates the communication; if the verification is true, the EIC generates a random number and computes the following:

$C_3 = h(\mathrm{PID}_j \,||\, r_t \,||\, T_2) \oplus r_e$, $C_4 = h(r_e \,||\, S_i \,||\, r_t \,||\, T_2)$, and session key $\mathrm{SK}_{ij} = h(r_t \,||\, r_e \,||\, S_i \,||\, \mathrm{PID}_i \,||\, \mathrm{PID}_j)$. Then, it sends message $M_2 = \{T_2, \mathrm{PID}_j, C_3, C_4\}$ to the TD.

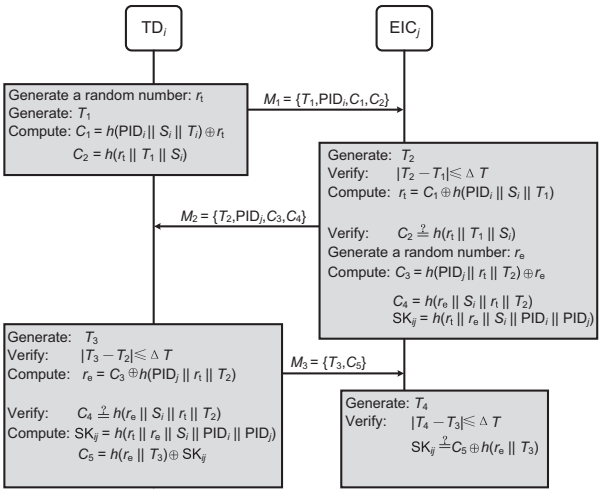

Fig. 3 Mutual authentication phase

Step 3: The TD generates timestamp T_3 and verifies $|T_3 - T_2| \leq \Delta T$. If it is within the delay time, it computes $r_e = C_3 \oplus h(\mathrm{PID}_j \,||\, r_t \,||\, T_2)$ and verifies $C_4 \stackrel{?}{=} h(r_e \,||\, S_i \,||\, r_t \,||\, T_2)$: if the verification is false, it means that the authentication request of the EIC is wrong, and the TD terminates the communication; if the verification is true, the TD computes the session key $\mathrm{SK}_{ij} = h(r_t \,||\, r_e \,||\, S_i \,||\, \mathrm{PID}_i \,||\, \mathrm{PID}_j)$ and $C_5 = h(r_e \,||\, T_3) \oplus \mathrm{SK}_{ij}$, and then sends message $M_3 = \{T_3, C_5\}$ to the EIC.

Step 4: The EIC generates timestamp T_4 and verifies $|T_4 - T_3| \leq \Delta T$. If it is within the delay time, the EIC verifies $\mathrm{SK}_{ij} \stackrel{?}{=} C_5 \oplus h(r_e \,||\, T_3)$ using the session key SK_{ij} generated in step 2: if the verification is false, the EIC terminates the communication; if the verification is true, it means that the TD has the legitimate session key.

4 Formal security analysis

We verify the scheme security using the AVISPA tool, widely used for assessing authentication protocols. AVISPA uses the high-level protocol specification language (HLPSL) to define protocols through different roles: (1) basic roles for the initial knowledge of entities and (2) compositional roles for their interactions. The environment section in HLPSL specifies the intruder's knowledge to uncover vulnerabilities, while the goal section defines security objectives. AVISPA automates verification, classifying protocols as secure or insecure based on the set standards.

We use the Security Protocol ANimator (SPAN) for AVISPA to simulate our proposed scheme, defining two primary roles, the TD and the EIC, along with the mandatory roles "session" and "target and environment." We focus on two AVISPA backends: OFMC and CL-AtSe. CL-AtSe uses a modularly unified algorithm that emphasizes algebraic properties such as XOR, while OFMC excels at analyzing protocols against various cyberattacks. AVISPA uses the DY threat model, in which adversaries can intercept, modify, and send messages within the network. Simulation results in Figs. 4 and 5 for OFMC and CL-AtSe backends respectively confirm our scheme's security under the DY threat model.

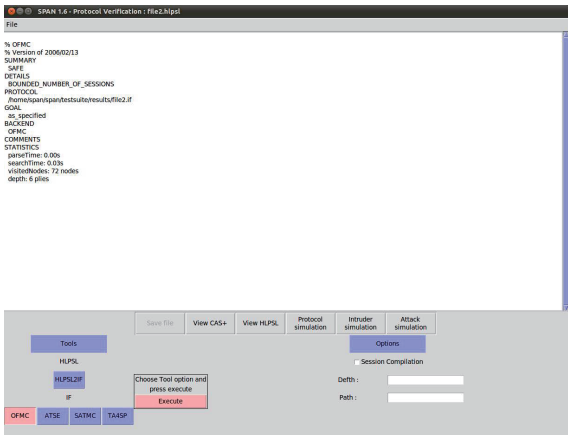

Fig. 4 Results for the OFMC backend

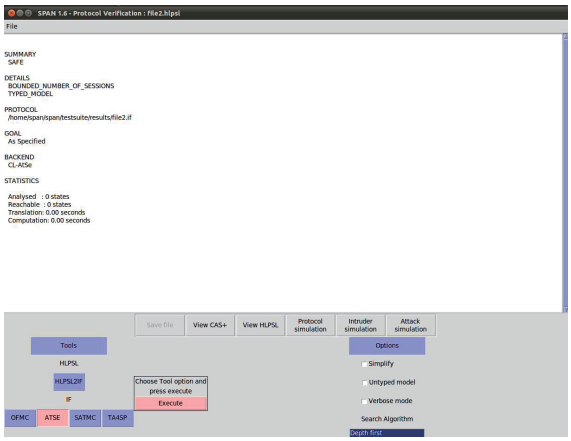

Fig. 5 Results for the CL-AtSe backend

5 Informal security analysis

In this section, we analyze the security of the proposed lightweight authentication scheme for ECSs.

5.1 Confidentiality

All the information that must be kept confidential between the EIC and the TD is encrypted by XOR or masked by collision-resistant hash functions, such as r_{t}, S_i, r_{e}, and SK_{ij}. To send r_{t} with confidentiality, it is encrypted by XOR, i.e., $C_1 = h(\mathrm{PID}_i \,||\, S_i \,||\, T_1) \oplus r_{\mathrm{t}}$, and masked by the hash function, i.e., $C_2 = h(r_{\mathrm{t}} \,||\, T_1 \,||\, S_i)$. To send S_i with confidentiality, it is masked by the hash function, i.e., $C_1 = h(\mathrm{PID}_i \,||\, S_i \,||\, T_1) \oplus r_{\mathrm{t}}$, $C_2 = h(r_{\mathrm{t}} \,||\, T_1 \,||\, S_i)$, and $C_4 = h(r_{\mathrm{e}} \,||\, S_i \,||\, r_{\mathrm{t}} \,||\, T_2)$. To send r_{e} with confidentiality, it is encrypted by XOR, i.e., $C_3 = h(\mathrm{PID}_j \,||\, r_{\mathrm{t}} \,||\, T_2) \oplus r_{\mathrm{e}}$, and masked by the hash function, i.e., $C_4 = h(r_{\mathrm{e}} \,||\, S_i \,||\, r_{\mathrm{t}} \,||\, T_2)$. To send SK_{ij} with confidentiality, it is encrypted by XOR, i.e., $C_5 = h(r_{\mathrm{e}} \,||\, T_3) \oplus \mathrm{SK}_{ij}$. In summary, this prevents an adversary from identifying the secret information from the transmitted message, and therefore confidentiality is achieved in this scheme.

5.2 Integrity

If modification of a message occurs during transmission, then this malicious operation will be detected. This is because with the collision-resistant hash function, once the input information changes, its output hash value will also change. For instance, if the adversary modifies the message in $M_1 = \{T_1, \mathrm{PID}_i, C_1, C_2\}$, then the EIC can use $C_2 \stackrel{?}{=} h(r_{\mathrm{t}} \,||\, T_1 \,||\, S_i)$ to detect it. If the adversary modifies the information in $M_2 = \{T_2, \mathrm{PID}_j, C_3, C_4\}$, then the TD can use $C_4 \stackrel{?}{=} h(r_{\mathrm{e}} \,||\, S_i \,||\, r_{\mathrm{t}} \,||\, T_2)$ to detect it. If the adversary modifies the message in $M_3 = \{T_3, C_5\}$, then the EIC can use $\mathrm{SK}_{ij} \stackrel{?}{=} C_5 \oplus h(r_{\mathrm{e}} \,||\, T_3)$ to detect it. Thus, this scheme achieves integrity.

5.3 Mutual authentication

In the authentication phase, mutual authentication between the EIC and the TD can be achieved based on messages M_1 and M_2. After receiving $M_1 = \{T_1, \mathrm{PID}_i, C_1, C_2\}$, the EIC first verifies whether timestamp T_1 is legitimate. If it is, the EIC then computes $r_{\mathrm{t}} = C_1 \oplus h(\mathrm{PID}_i \,||\, S_i \,||\, T_1)$; afterward, it verifies whether C_2 is equal to $h(r_{\mathrm{t}} \,||\, T_1 \,||\, S_i)$. If it is, the TD is considered to have passed the authentication. When the TD receives $M_2 = \{T_2, \mathrm{PID}_j, C_3, C_4\}$, it first verifies whether timestamp T_2 is legal. If it is, then the TD computes $r_{\mathrm{e}} = C_3 \oplus h(\mathrm{PID}_j \,||\, r_{\mathrm{t}} \,||\, T_2)$; thereafter, it verifies whether C_4 is equal to $h(r_{\mathrm{e}} \,||\, S_i \,||\, r_{\mathrm{t}} \,||\, T_2)$. If it is, the EIC is considered to have passed the authentication. Moreover, if the adversary's goal is to forge into a valid TD or EIC, he/she needs to generate a valid message. However, the adversary cannot generate a valid message because he/she does not have the information of the secret value S_i. Therefore, this scheme achieves mutual authentication.

5.4 Anonymity

From $S_i = h(\mathrm{ID}_{t_i} \,||\, r_{\mathrm{s}})$, we know that ID_{t_i} of the TD is protected by the collision-resistant hash function and the random number r_{s}. Note that r_{s} is a transient random number for the current session. It changes during each session, and its value is unique and unpredictable each time. Therefore, it is impossible for an adversary to know the identity information of the TD, and the secret value S_i is stored in a secure manner and is generally not disclosed. In addition, since the pseudo-identity PID_i of the TD and the pseudo-identity PID_j of the EIC are randomly generated by the TS and change after each session, the values of their pseudo-identities are unique and unpredictable for each session. When performing mutual authentication, both parties are authenticated by pseudo-identity. Therefore, this scheme has anonymity.

5.5 Forward secrecy

The session key is computed and generated by $\mathrm{SK}_{ij} = h(r_{\mathrm{t}} \,||\, r_{\mathrm{e}} \,||\, S_i \,||\, \mathrm{PID}_i \,||\, \mathrm{PID}_j)$, where S_i is a secret value that is stored in a secure manner for both the EIC and the TD. From $S_i = h(\mathrm{ID}_{t_i} \,||\, r_{\mathrm{s}})$, we know that S_i is different in each session because the random number r_{s} is different in each session. Even if the current S_i is leaked, it is not possible to compute the previous values of S_i, which are independent and unconnected for each session. Therefore, the adversary cannot use the leaked S_i to obtain the previous S_i values; thus, the previous session keys SK_{ij} are secure. This scheme achieves forward secrecy.

5.6 Known session key security

The key agreement should provide a unique session key for each execution. In our authentication scheme, the session key $\mathrm{SK}_{ij} = h(r_\mathrm{t} \,||\, r_\mathrm{e} \,||\, S_i \,||\, \mathrm{PID}_i \,||\, \mathrm{PID}_j)$ between the EIC and the TD can be used for subsequent secure communication. In addition, the session key depends on the transient random numbers r_t and r_e; it is different and unlinkable for each session. We can conclude that the session key in a session is independent of the session keys of other sessions, and the session key SK_{ij} is unique and unlinkable in each run of the authentication phase. Therefore, an adversary cannot use the leaked session key to compute other session keys, and our authentication scheme achieves known session key security.

5.7 Resistance to tracking attacks

In this attack, the adversary intercepts messages from different sessions and tries to determine the relationship between them, thus inferring whether they belong to the same node (Tan et al., 2008). Since the random numbers $r_\mathrm{s}, r_\mathrm{t}$, and r_e are transient random numbers, the related messages sent in each session are different; for instance, S_i, C_1, C_2, C_3, C_4, and C_5 are different in each session. The pseudo-identity PID_i of the TD and the pseudo-identity PID_j of the EIC are randomly generated by the TS and change after completion of each session. As a result, an adversary will not be able to distinguish whether different sessions belong to the same TD or the same EIC by the intercepted messages. Based on the foregoing analysis, the present authentication scheme has the ability to resist tracking attacks.

5.8 Resistance to impersonation attacks

This is not feasible if an adversary wants to impersonate a TD or an EIC. Because the adversary does not have the secret value S_i, it cannot generate the correct and legitimate $C_1 = h(\mathrm{PID}_i \,||\, S_i \,||\, T_1) \oplus r_\mathrm{t}$ and $C_2 = h(r_\mathrm{t} \,||\, T_1 \,||\, S_i)$, so it cannot impersonate the TD. Similarly, the correct and legitimate $C_3 = h(\mathrm{PID}_j \,||\, r_\mathrm{t} \,||\, T_2) \oplus r_\mathrm{e}$ and $C_4 = h(r_\mathrm{e} \,||\, S_i \,||\, r_\mathrm{t} \,||\, T_2)$ cannot be generated, so the EIC cannot be impersonated. Therefore, this scheme has the ability to resist impersonation attacks.

5.9 Resistance to MITM attacks

When the adversary executes an MITM attack, it will capture and modify the messages sent between the TD and the EIC and render the TD and the EIC undetectable, but this is not feasible in this scheme. Once the adversary modifies the data in message $M_1 = \{T_1, \mathrm{PID}_i, C_1, C_2\}$, it will be detected by the EIC due to the nature of the hash function. If the adversary wants to make the EIC undetectable, then he/she must have the secret value S_i, but S_i is masked by the hash function. Similarly, the adversary who modifies the data in message $M_2 = \{T_2, \mathrm{PID}_j, C_3, C_4\}$ will be detected by the TD due to the nature of the hash function. If the adversary wants to make the TD undetectable, then he/she must have the random number r_t and the secret value S_i, but r_t and S_i are masked by the hash function. Finally, the adversary cannot compute session key $\mathrm{SK}_{ij} = h(r_\mathrm{t} \,||\, r_\mathrm{e} \,||\, S_i \,||\, \mathrm{PID}_i \,||\, \mathrm{PID}_j)$ because there are no random numbers $r_\mathrm{t}, r_\mathrm{e}$ or secret value S_i, so it cannot generate the legitimate C_5, and it cannot complete the verification of the session key. Therefore, this scheme has the ability to resist MITM attacks.

5.10 Resistance to replay attacks

Our scheme is to resist replay attacks through timestamps. By using a timestamp to verify the transmission delay, the transmission delay of messages from old sessions will exceed the allowed time ΔT. The EIC and the TD check the transmission delay at each session. If the adversary modifies timestamp T_i, it will be detected by the EIC with the TD. If the adversary modifies timestamp T_1, the EIC verifies $C_2 \stackrel{?}{=} h(r_\mathrm{t} \,||\, T_1 \,||\, S_i)$; it will find that C_2 does not match, and then it will terminate the communication. If the adversary modifies timestamp T_2, the TD verifies $C_4 \stackrel{?}{=} h(r_\mathrm{e} \,||\, S_i \,||\, r_\mathrm{t} \,||\, T_2)$; it will find that C_4 does not match, and then it will terminate the communication. If the adversary modifies timestamp T_3, the EIC verifies $\mathrm{SK}_{ij} \stackrel{?}{=} C_5 \oplus h(r_\mathrm{e} \,||\, T_3)$; it will find that SK_{ij} does not match, and then it will terminate the communication. Therefore, this scheme has the ability to resist replay attacks.

5.11 Resistance to DoS attacks

In all messages, the EIC and the TD first verify the received timestamp. If an adversary replays

earlier session messages, the EIC and the TD immediately reject them and terminate the communication, protecting computing and storage resources. In addition, all messages rely on the request–response communication principle; i.e., the EIC or the TD performs integrity checks on the received messages and then responds to them with a pass or a reject. If an adversary sends fake messages to the EIC or the TD, the EIC and the TD can detect these messages using the collision resistance of the hash function (as described in Section 5.2) and immediately terminate the communication to protect the computing and storage resources. Therefore, this scheme can resist DoS attacks to a certain extent.

5.12 Resistance to desynchronization attacks

This attack is found mainly in authentication schemes that require update operations on session keys, whereby an adversary can perform desynchronization attacks by intercepting messages between two parties or by intercepting messages and tampering with them. Let us suppose that the adversary intercepts message $M_2 = \{T_2, \mathrm{PID}_j, C_3, C_4\}$ sent by the EIC to the TD; in this case, the TD will not receive M_2 within the given delay time ΔT, and the TD will restart another session, regenerate the random number r_t and timestamp T_1, and send the related message $M_1 = \{T_1, \mathrm{PID}_i, C_1, C_2\}$ to the EIC. The TD and the EIC can negotiate the session key SK_{ij} in the restarted session, and the EIC verifies whether the generated session key is consistent at the end, i.e., $\mathrm{SK}_{ij} \stackrel{?}{=} C_5 \oplus h(r_\mathrm{e} \,||\, T_3)$; if it is not consistent, the EIC will terminate the communication. If the adversary intercepts and tampers with message $M_2 = \{T_2, \mathrm{PID}_j, C_3, C_4\}$ and sends it to the TD, the TD can verify $C_4 \stackrel{?}{=} h(r_\mathrm{e} \,||\, S_i \,||\, r_\mathrm{t} \,||\, T_2)$ to discover that the adversary has modified message M_2, and then terminates the scheme. Thus, this scheme can resist desynchronization attacks.

5.13 Resistance to stolen-verifier attacks

This attack targets mainly at verifiers, who have a verification repository or verification database that records information about the provers; the attacker can impersonate a legitimate device using the information in the verification repository or verification database (Mahmood et al., 2018). According to the description of this authentication scheme, the EIC and the TD do not maintain any verification repository or verification database. In addition, since the secret value S_i is stored in a secure manner by the EIC and the TD, an adversary cannot steal these data. Therefore, this scheme has the ability to resist stolen-verifier attacks.

6 Performance analysis and comparisons

In this section, we evaluate the performance of the proposed lightweight authentication scheme for ECSs in terms of security properties, communication overhead, and computational cost.

6.1 Security properties

Table 2 gives a comparison of the security properties of our authentication scheme and other authentication schemes.

In Sun et al. (2015), the identity information of the authentication scheme was not anonymized in any way, so this authentication scheme does not have anonymity. In the models of Esfahani et al. (2019), Jan et al. (2021), and Ehui et al. (2022), the keys in their authentication schemes are always constant, and if an adversary obtains the key at a certain time, the session key before this time is not guaranteed to be legitimate. Therefore, these authentication schemes do not ensure forward secrecy. The schemes of Sun et al. (2015), Esfahani et al. (2019), and Ehui et al. (2022) do not provide the security property (SP) of untraceability, and an adversary can determine the relationship between entities based on the messages they send to each other, thus inferring whether they belong to the same node. Therefore, these schemes are not resistant to tracking attacks. In Sun et al. (2015) and Jan et al. (2021), the schemes do not provide the SP to resist MITM attacks, which can undermine the correctness of the authentication. Sun et al. (2015), Esfahani et al. (2019), Zhang LP et al. (2019), and Ehui et al. (2022) did not provide measures to resist DoS attacks, such as adding timestamps, and the timestamp mechanism can be used to resist DoS attacks (Rose and Jayasree, 2019; Xiao et al., 2024b). Therefore, these schemes are not resistant to DoS attacks. In the schemes of Sun et al. (2015), Esfahani et al. (2019), and Ehui et al. (2022), security against desynchronization attacks is not provided, which can cause

Table 2 Comparison of security properties among different authentication schemes

SP	Sun et al. (2015)'s	Esfahani et al. (2019)'s	Zhang LP et al. (2019)'s	Jan et al. (2021)'s	Ehui et al. (2022)'s	Ours
SP1	Yes	Yes	Yes	Yes	Yes	Yes
SP2	Yes	Yes	Yes	Yes	Yes	Yes
SP3	Yes	Yes	Yes	Yes	Yes	Yes
SP4	No	Yes	Yes	Yes	Yes	Yes
SP5	Yes	No	Yes	No	No	Yes
SP6	Yes	Yes	Yes	Yes	Yes	Yes
SP7	No	No	Yes	Yes	No	Yes
SP8	Yes	Yes	Yes	Yes	Yes	Yes
SP9	No	Yes	Yes	No	Yes	Yes
SP10	Yes	Yes	Yes	Yes	Yes	Yes
SP11	No	No	No	Yes	No	Yes
SP12	No	No	Yes	Yes	No	Yes
SP13	Yes	No	Yes	No	No	Yes

DoS: denial-of-service; MITM: man-in-the-middle; SP: security property; SP1: confidentiality; SP2: integrity; SP3: mutual authentication; SP4: anonymity; SP5: forward secrecy; SP6: known session key security; SP7: resistance to tracking attacks; SP8: resistance to impersonation attacks; SP9: resistance to MITM attacks; SP10: resistance to replay attacks; SP11: resistance to DoS attacks; SP12: resistance to desynchronization attacks; SP13: resistance to stolen-verifier attacks

session keys to be unsynchronized and thereby affect the secure communication later. Esfahani et al. (2019), Jan et al. (2021), and Ehui et al. (2022) did not provide the SP to resist stolen-verifier attacks in their schemes. Therefore, the authentication scheme proposed in this paper provides more security properties compared to other authentication schemes.

6.2 Communication overhead

To put our authentication scheme into practice, we set the parameters of the authentication scheme as shown in Table 3.

Table 3 Setting of parameters

Parameter	Value (bit)
Random number	128
Identity/Pseudo-identity	128
Hash value	160
HMAC value	160
Timestamp	32
AES encryption	128
AES decryption	128

HMAC: hash-based message authentication code; AES: advanced encryption standard

We compute the communication cost of our authentication scheme and the authentication schemes of Sun et al. (2015), Esfahani et al. (2019), Zhang LP et al. (2019), Jan et al. (2021), and Ehui et al. (2022) with reference to the parameters in Table 3.

As shown in Table 4, our authentication scheme has the smallest communication cost, while the authentication scheme of Ehui et al. (2022) has the largest communication cost. In our authentication scheme, the messages transmitted in the registration phase are ID_{t_i}, $\{\mathrm{PID}_i, S_i\}$, and $\{\mathrm{PID}_j, S_i\}$, requiring a communication overhead of 704 bits. The messages transmitted in the authentication phase are $M_1 = \{T_1, \mathrm{PID}_i, C_1, C_2\}$, $M_2 = \{T_2, \mathrm{PID}_j, C_3, C_4\}$, and $M_3 = \{T_3, C_5\}$, which involve communication overheads of 448, 448, and 192 bits, respectively. The total communication cost of our authentication scheme is only 160 bits more than that of the smallest one.

Table 4 Comparison of communication overhead

Scheme	Communication overhead (bit)		Total (bit)
	Registration phase	Authentication phase	
Sun et al. (2015)'s	512	1248	1760
Esfahani et al. (2019)'s	448	1184	1632
Zhang LP et al. (2019)'s	512	1344	1856
Jan et al. (2021)'s	640	1824	2464
Ehui et al. (2022)'s	–	2560	2560
Ours	704	1088	1792

6.3 Computational cost

In this subsection, the calculation cost of our certification scheme is compared with those of other certification schemes, as shown in Table 5.

The execution time of the XOR operation is not considered in the analysis because it is negligible compared to those of other operations (Li et al., 2021).

Sun et al. (2015) proposed an authentication scheme mainly for home network services. In the

Table 5 Comparison of computational cost

Scheme	Computational cost	
	Registration phase	Authentication phase
Sun et al. (2015)'s	User: – Server: $T_h + T_e$	User: $4T_h + T_e$ Server: $5T_h + 2T_d$
Esfahani et al. (2019)'s	Smart sensor: – Authentication server: $2T_h$	Smart sensor: $7T_h$ Router: $7T_h$
Zhang LP et al. (2019)'s	Smart sensor: – Service provider: $2T_h + T_e$	Smart sensor: $7T_h + T_d$ Service provider: $9T_h + 2T_e + T_d$
Jan et al. (2021)'s	Client: $2T_h$ Remote server: $3T_h$	Client and gateway: $9T_h$ Remote server: $8T_h$
Ehui et al. (2022)'s	Sensor: – Gateway: –	Sensor: $4T_e + 4T_d + 3T_h + 4T_{hmac}$ Gateway: $4T_e + 4T_d + 2T_h + 4T_{hmac}$
Ours	TD: – TS: T_h	TD: $6T_h$ EIC: $6T_h$

EIC: edge intelligent controller; TD: terminal device; TS: trusted server. T_{hmac}: execution time of the hash-based message authentication code (HMAC); T_h: execution time of the hash function; T_e: execution time of advanced encryption standard (AES) encryption; T_d: execution time of AES decryption

registration phase, the user sends a registration request to the server, and the server needs $T_h + T_e$ to respond to the registration request and complete the registration of the user. During the authentication phase, the user and the server need $4T_h + T_e$ and $5T_h + 2T_d$, respectively, to complete the authentication. A lightweight authentication scheme was proposed mainly for the IIoT by Esfahani et al. (2019). In the registration phase, the smart sensor executes the registration procedure with the authentication server through a secure channel, and the authentication server needs $2T_h$ to complete the registration process. In the authentication phase, the smart sensor and the router both need $7T_h$ to complete the authentication. In Zhang LP et al. (2019), a lightweight authentication scheme was proposed mainly for the smart grid. In the registration phase, the smart sensor wants to access the corresponding service provider, and the service provider needs $2T_h + T_e$ to respond to the smart sensor's access request and complete the registration process. In the authentication phase, the smart sensor and the service provider need $7T_h + T_d$ and $9T_h + 2T_e + T_d$ to complete the authentication, respectively. A lightweight authentication scheme was proposed mainly for smart healthcare systems in the IoT by Jan et al. (2021). In the registration phase, the client (wearable device) and the remote server need $2T_h$ and $3T_h$, respectively, to complete the registration process. In the authentication phase, the client and gateway need $9T_h$, and the remote server needs $8T_h$, to achieve authentication. Ehui et al. (2022) proposed a lightweight authentication scheme mainly for the IoT. This authentication scheme has no registration phase since the secret key to establish secure authentication is shared directly between the sensor and the gateway. In the authentication phase, the sensor and the gateway need $4T_e+4T_d+3T_h+4T_{hmac}$ and $4T_e+4T_d+2T_h+4T_{hmac}$, respectively, to complete the authentication. In our authentication scheme, in the registration phase, the TS needs T_h to complete the registration process. In the authentication phase, the TD and the EIC both need $6T_h$ to complete the authentication.

We use an industrial control computer with an Intel Celeron J1900 (2.0 GHz) CPU and 8 GB RAM as the EIC and an embedded development board with I.MX6ULL (800 MHz) CPU and 512 MB RAM as the TD in our simulations. We use C/C++ programming language and the OpenSSL library to implement the hash algorithm, HMAC algorithm, and the advanced encryption standard (AES) encryption/decryption algorithm for the registration and authentication phases. The execution time of each algorithm is shown in Table 6.

From Table 6, we can see that the HMAC algorithm is the most time-consuming and is the key to the time delay of the authentication process. Next, we put the authentication schemes in the literature in the simulation conditions of this study to compare the computation time at the authentication phase. The comparison results are shown in Table 7.

According to Table 7, we can conclude that,

Table 6 Execution time

Device	T_{h} (μs)	T_{e} (μs)	T_{d} (μs)	T_{hmac} (μs)
EIC	5.400	10.992	13.116	173.500
TD	24.000	38.333	40.000	432.000

EIC: edge intelligent controller; TD: terminal device; T_{hmac}: execution time of the hash-based message authentication code (HMAC); T_{h}: execution time of the hash function; T_{e}: execution time of advanced encryption standard (AES) encryption; T_{d}: execution time of AES decryption

Table 7 Comparison of computation time

Scheme	Computation time (μs)		Total (μs)
	TD	EIC	
Sun et al. (2015)'s	134.333	53.232	187.565
Esfahani et al. (2019)'s	168.000	37.800	205.800
Zhang LP et al. (2019)'s	208.000	83.700	291.700
Jan et al. (2021)'s	216.000	43.200	259.200
Ehui et al. (2022)'s	2113.332	801.232	2914.564
Ours	144.000	32.400	176.400

EIC: edge intelligent controller; TD: terminal device

on the TD, the execution time of the authentication scheme of Sun et al. (2015) is the least, while the execution time of our authentication scheme is 9.667 μs more. The authentication scheme of Ehui et al. (2022) has the highest execution time. This is because this authentication scheme contains the HMAC algorithm. In Esfahani et al. (2019), Zhang LP et al. (2019), and Jan et al. (2021), the execution times of their authentication schemes are 24.000, 64.000, and 72.000 μs more than the execution time of our authentication scheme, respectively.

Regarding the EIC, the authentication scheme of Ehui et al. (2022) has the highest execution time, and our authentication scheme has the lowest execution time. In Sun et al. (2015), Esfahani et al. (2019), Zhang LP et al. (2019), and Jan et al. (2021), the execution times of their authentication schemes are 20.832, 5.400, 51.300, and 10.800 μs more than the execution time of our authentication scheme, respectively.

Considering the total cost, the authentication scheme of Ehui et al. (2022) has the highest execution time, and our authentication scheme has the lowest execution time. The execution times of the authentication schemes proposed by Sun et al. (2015), Esfahani et al. (2019), Zhang LP et al. (2019), and Jan et al. (2021) are 11.165, 29.400, 115.300, and 82.800 μs more than the execution time of our authentication scheme, respectively. Therefore, our authentication scheme has the best performance.

7 Conclusions

In the context of ECSs, authentication between the EIC and TDs is one of the most fundamental security issues. In this paper, we propose a lightweight authentication scheme for ECSs that enables bidirectional anonymous authentication and key agreement between the EIC and TDs, and the negotiated session key can be used for subsequent secure communication between the two parties. The security analysis shows that the authentication scheme can provide the necessary security properties. In addition, performance analysis demonstrates the benefits of the authentication scheme in terms of security properties, communication overhead, and computational cost. Due to the large number of TDs accessing the EIC, however, achieving batch authentication of TDs is the focus of our future work.

Contributors

Xudong WEN and Zhuo CHEN performed numerical simulations. Wenze XIONG accomplished experimental verification. Wenli SHANG drafted the paper. Zhiwei CHANG revised the paper. Zhong CAO supervised the project and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Aman MN, Taneja S, Sikdar B, et al., 2019. Token-based security for the Internet of Things with dynamic energy-quality tradeoff. *IEEE Int Things J*, 6(2):2843-2859. https://doi.org/10.1109/JIOT.2018.2875472

Cao Z, Chen Z, Shang WL, et al., 2023. Efficient revocable anonymous authentication mechanism for edge intelligent controllers. *IEEE Int Things J*, 10(12):10357-10367. https://doi.org/10.1109/JIOT.2023.3237609

Cui J, Wang FQ, Zhang QY, et al., 2021. Anonymous message authentication scheme for semitrusted edge-enabled IIoT. *IEEE Trans Ind Electron*, 68(12):12921-12929. https://doi.org/10.1109/TIE.2020.3039227

Cui J, Bian FY, Zhong H, et al., 2022. An anonymous and outsourcing-supported multiauthority access control scheme with revocation for edge-enabled IIoT system. *IEEE Syst J*, 16(4):6569-6580. https://doi.org/10.1109/JSYST.2022.3189219

Cui J, Wang FQ, Zhang QY, et al., 2023. Efficient batch authentication scheme based on edge computing in IIoT. *IEEE Trans Netw Serv Manag*, 20(1):357-368. https://doi.org/10.1109/TNSM.2022.3206378

Dolev D, Yao A, 1983. On the security of public key protocols. *IEEE Trans Inform Theory*, 29(2):198-208. https://doi.org/10.1109/TIT.1983.1056650

Ehui BB, Han YR, Guo H, et al., 2022. A lightweight mutual authentication protocol for IoT. *J Commun Inform Netw*, 7(2):181-191. https://doi.org/10.23919/JCIN.2022.9815201

Esfahani A, Mantas G, Matischek R, et al., 2019. A lightweight authentication mechanism for M2M communications in Industrial IoT environment. *IEEE Int Things J*, 6(1):288-296. https://doi.org/10.1109/JIOT.2017.2737630

Esposito C, Castiglione A, Palmieri F, et al., 2018. Integrity for an event notification within the Industrial Internet of Things by using group signatures. *IEEE Trans Ind Inform*, 14(8):3669-3678. https://doi.org/10.1109/TII.2018.2791956

Gadekallu TR, Pham QV, Nguyen DC, et al., 2022. Blockchain for Edge of Things: applications, opportunities, and challenges. *IEEE Int Things J*, 9(2):964-988. https://doi.org/10.1109/JIOT.2021.3119639

Jan MA, Khan F, Mastorakis S, et al., 2021. Light-IoT: lightweight and secure communication for energy-efficient IoT in health informatics. *IEEE Trans Green Commun Netw*, 5(3):1202-1211. https://doi.org/10.1109/TGCN.2021.3077318

Khan R, Teo J, Jan MA, et al., 2023. A trustworthy, reliable, and lightweight privacy and data integrity approach for the Internet of Things. *IEEE Trans Ind Inform*, 19(1):511-518. https://doi.org/10.1109/TII.2022.3179728

Li JL, Su Z, Guo DK, et al., 2021. PSL-MAAKA: provably secure and lightweight mutual authentication and key agreement protocol for fully public channels in Internet of Medical Things. *IEEE Int Things J*, 8(17):13183-13195. https://doi.org/10.1109/JIOT.2021.3055827

Liu Y, Chi C, Zhang YW, et al., 2022. Identification and resolution for Industrial Internet: architecture and key technology. *IEEE Int Things J*, 9(18):16780-16794. https://doi.org/10.1109/JIOT.2022.3160737

Mahmood K, Chaudhry SA, Naqvi H, et al., 2018. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Fut Gener Comput Syst*, 81:557-565. https://doi.org/10.1016/j.future.2017.05.002

Nkenyereye L, Hwang J, Pham QV, et al., 2021. Virtual IoT service slice functions for multiaccess edge computing platform. *IEEE Int Things J*, 8(14):11233-11248. https://doi.org/10.1109/JIOT.2021.3051652

Rose SGH, Jayasree T, 2019. Detection of jamming attack using timestamp for WSN. *Ad Hoc Netw*, 91:101874. https://doi.org/10.1016/j.adhoc.2019.101874

Sharp M, Ak R, Hedberg TJr, 2018. A survey of the advancing use and development of machine learning in smart manufacturing. *J Manuf Syst*, 48:170-179. https://doi.org/10.1016/j.jmsy.2018.02.004

Sisinni E, Saifullah A, Han S, et al., 2018. Industrial Internet of Things: challenges, opportunities, and directions. *IEEE Trans Ind Inform*, 14(11):4724-4734. https://doi.org/10.1109/TII.2018.2852491

Sodhro AH, Pirbhulal S, de Albuquerque VHC, 2019. Artificial intelligence-driven mechanism for edge computing-based industrial applications. *IEEE Trans Ind Inform*, 15(7):4235-4243. https://doi.org/10.1109/TII.2019.2902878

Sun XB, Men S, Zhao CL, et al., 2015. A security authentication scheme in machine-to-machine home network service. *Secure Commun Netw*, 8(16):2678-2686. https://doi.org/10.1002/sec.551

Tan CC, Sheng B, Li Q, 2008. Secure and serverless RFID authentication and search protocols. *IEEE Trans Wirel Commun*, 7(4):1400-1407. https://doi.org/10.1109/TWC.2008.061012

Wang JJ, Ma YL, Zhang LB, et al., 2018. Deep learning for smart manufacturing: methods and applications. *J Manuf Syst*, 48:144-156. https://doi.org/10.1016/j.jmsy.2018.01.003

Wazid M, Das AK, Odelu V, et al., 2020. Secure remote user authenticated key establishment protocol for smart home environment. *IEEE Trans Depend Secure Comput*, 17(2):391-406. https://doi.org/10.1109/TDSC.2017.2764083

Xiao SY, Ge XH, Han QL, et al., 2022. Secure distributed adaptive platooning control of automated vehicles over vehicular ad-hoc networks under denial-of-service attacks. *IEEE Trans Cybern*, 52(11):12003-12015. https://doi.org/10.1109/TCYB.2021.3074318

Xiao SY, Ge XH, Ding L, et al., 2024a. A bandwidth-conscious event-based control approach to secondary frequency regulation under vehicle-to-grid service. *IEEE Trans Smart Grid*, 15(4):3739-3750. https://doi.org/10.1109/TSG.2024.3365473

Xiao SY, Ge XH, Wu Q, et al., 2024b. Co-design of bandwidth-aware communication scheduler and cruise controller for multiple high-speed trains. *IEEE Trans Veh Technol*, 73(4):4993-5004. https://doi.org/10.1109/TVT.2023.3332609

Zhang LP, Zhao LC, Yin SJ, et al., 2019. A lightweight authentication scheme with privacy protection for smart grid communications. *Fut Gener Comput Syst*, 100:770-778. https://doi.org/10.1016/j.future.2019.05.069

Zhang QY, Wu J, Zhong H, et al., 2023. Efficient anonymous authentication based on physically unclonable function in Industrial Internet of Things. *IEEE Trans Inform Forens Secur*, 18:233-247. https://doi.org/10.1109/TIFS.2022.3218432

Zhang Y, Wei HY, 2021. Risk-aware cloud-edge computing framework for delay-sensitive industrial IoTs. *IEEE Trans Netw Serv Manag*, 18(3):2659-2671. https://doi.org/10.1109/TNSM.2021.3092790

Zhou W, Jia Y, Peng AN, et al., 2019. The effect of IoT new features on security and privacy: new threats, existing solutions, and challenges yet to be solved. *IEEE Int Things J*, 6(2):1606-1616. https://doi.org/10.1109/JIOT.2018.2847733