TRUTH: Trust and Authentication Scheme in 5G-IIoT

Seyed Ahmad Soleymani, Shidrokh Goudarzi, Mohammad Hossein Anisi, Haitham Cruickshank, Anish Jindal, Nazri Kama

Abstract—Due to the extremely important role of data in the Industrial Internet of Things (IIoT) network, trust and security of data are among the major concerns. In this study, we develop a cloud-integrated 5G-IIoT network architecture enabled by a three-party Authenticated Key Exchange (AKE) protocol with privacy-preserving to secure data exchanged via wireless communication, cope with unauthorized entities and ensure data integrity. Moreover, we develop a trust model based on the Dempster-Shafer theory to check the trustworthiness of data collected by smart devices/sensor nodes. Security analysis performed on our scheme demonstrates that it can withstand different well-known attacks in the IIoT environment. We also analyzed the validity of our scheme by using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. Additionally, the performance evaluation and experimental results prove the effectiveness of the proposed scheme compared to the existing works in terms of accuracy, delay, trust, and throughput.

Index Terms—Trust, Authentication, Privacy, Cloud Computing, 5G, IIoT.

1 Introduction

Industrial Internet of Things (IIoT) based on the standard communication protocols can intelligently connect humans and things anytime and anywhere. IIoT technologies are being more widely adopted across industry sectors and organizations. The automotive industry, aerospace and defense companies, and healthcare industry are some of the major beneficiaries of IIoT technology. Today, 5G cellular network can expand IIoT capabilities to connect billions of sensors/smart devices. 5G-IIoT, as the integration of 5G and IIoT, by extending coverage, higher throughput, and lower latency, allows different industrial things to communicate at a much higher rate in different smart environments. In other words, 5G networks are a major driving force for the growth of IIoT. However, in the 5G-IIoT network, it is difficult to keep track of the deployed sensor nodes/smart devices that produce the analysis of the environment where these are deployed. Besides, accessing such deployed sensor nodes/smart devices in this network by an unauthorized user is always viable. Moreover, the features of such a network, such as the open nature of wireless communication, make it an ideal medium for malicious attackers to intrude on the system. Due to the extremely important role of data in 5G-IIoT networks and taking into account the features of such networks, the trustworthiness of sensor nodes/smart devices as well as trustworthiness of data collected/sensed by them, securing data-in-transit over wireless communication and privacy-preserving of personal data are still very challenging. These challenges are particularly highlighted in data-centric applications such as smart factories and healthcare systems in which high level of data privacy and security is required.

It is obvious that untrustworthy devices/nodes as well as untrustworthy data generated by smart devices will have a negative impact on the performance and efficiency of the system. This data can lead to wrong decisions by the system or users. Therefore, it is essential to ensure the trustworthiness of data and avoid broadcasting unreliable data in the network. Moreover, it is required to establish secure communication and ensure the security of this data during transmission on the network. To this end, there exist different types of security mechanisms, such as cryptography and authentication, however, they are still vulnerable to a number of security threats. For this purpose, authors in [1] explained that in order to establish secure communication, it needs to ensure the trustworthiness of all communicating parties, too.

To deal with these concerns, incorporating an efficient trust model and security scheme is an effective solution. On one hand, a trust model is able to identify untrustworthy nodes and data by monitoring the continuous behavior of IIoT smart devices and as a result, ensure trustworthy of nodes [2] and on the other hand, a security scheme can support security requirements such as confidentiality and integrity for data-in-transit.

This should be done by designing the proper network architecture and in order to reduce the computation and

- *S. A. Soleymani and H. Cruickshank are with the Institute for Communication Systems (ICS), University of Surrey, Guildford GU2 7XH, UK. E-mail: s.soleymani@surrey.ac.uk; h.Cruickshank@surrey.ac.uk*
- *S. Goudarzi is with Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford GU2 7XH, UK. E-mail: s.goudarzi@surrey.ac.uk*
- *M. H. Anisi is with School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK. E-mail: m.anisi@essex.ac.uk*
- *A. Jindal is with Department of Computer Science, Durham University, 3057 Durham, Durham, United Kingdom of Great Britain and Northern Ireland. E-mail: anish.jindal@durham.ac.uk*
- *N. Kama is with the Universiti Teknologi Malaysia (UTM), Malaysia. Email: mdnazri@utm.my*

communication costs. Besides, efficient data processing techniques are required to empower 5G-IIoT with data handling capabilities. Such techniques would be able to reduce computational delay and substantially minimize the cost of data transmission and storage [3]. For instance, cloud-based data analysis techniques are the most appealing strategies to meet the ever-increasing demands [4].

In this study, we employ a cloud-integrated 5G-IIoT to provide a flexible and scalable architecture consisting of sensor nodes, gateway/smart hub, cloud server, and users that are connected through the open wireless channel. Moreover, in order to address the concerns and challenges discussed, we propose a robust scheme to ensure only trusted data is securely transferred among the entities. The proposed security scheme is not only able to improve security and privacy, but also reduces network congestion and enhances bandwidth performance. We develop a trust model in order to establish trust among smart devices/sensor nodes and gateway. The model is able to improve security and performance of distributed networks such as Wireless Sensor Networks (WSNs) used in 5G-IIoT. Given the symmetric encryption method i.e., Advanced Encryption Standard (AES) used in this work, it is firstly required to generate and share the session key between parties securely. For this purpose, we design a three-party scheme based on AKE protocol for secure communication in heterogeneous systems. This protocol allows the participated parties to authenticate each other and share the session key for subsequent communications [5]. The scheme ensures that only authorized users can have access to the data generated by smart devices/sensor nodes deployed in the IIoT environment. The main contributions of this study are as follows:

1- We integrate cloud computing into 5G-IIoT to provide a flexible and scalable architecture for handling complexity.
2- We develop a trust model based on Dempster-shafer theory in order to guarantee the trustworthiness and reliability of the data collected by sensor nodes/smart devices.
3- We develop an authentication scheme based on the AKE to ensure the data integrity and in addition cope with unauthorized entities.
4- We develop TRUTH that is a combination of the proposed trust model and authentication scheme to protect the integrity of trustable data-in-transit.

The remaining of the manuscript is organized as follows: The related works are discussed in Section 2. Section 3 explains the system model and requirements. In Section 4, integration the proposed authentication scheme and trust model is presented. In Section 5, the formal security analysis is discussed. In Section 6, non-mathematical security analysis is carried out. Section 7 presents the performance analysis and practical perspective evaluation is discussed in Section 8. The paper is concluded in Section 9.

2 Related Work

The two main areas of study related to our work are the existing proposals on solving the authentication and privacy concerns in IIoT, and the reliability and trustworthiness of data collected by distributed sensors. Due to the vital role of data in IIoT networks, it is critical to ensure the trustworthiness and integrity of data. Besides, it is required to prevent private and sensitive information leakage.

In [6], an authentication scheme based on Elliptic Curve Cryptography (ECC) is proposed to provide security requirements for WSNs. To cope with security concerns in [6], an authentication scheme is developed in [7] to ensure the security requirements for WSN. In [8], authentication is considered as most important security requirement in IoT network. To meet this requirement, the authors propose a signature based on ECC. Also, a three-factor user authentication protocol by using cryptographic hash function and symmetric encryption/decryption method is proposed for IoT network in [9]. In this work, three parameters such as password and biometric information are also utilized to make this protocol more secure. In comparison to ECC and RSA as public key-based cryptosystems, the execution time of chaotic map-based operations are more less [10]. Building on this, a user authentication scheme based on chaotic-map based user authentication approach for IoT network is developed in [10]. In [11], the key establishment protocol employed in their authentication approach supports the security of data exchanged between smart devices and users. To this end, a session key will be generated and shared between the user and smart device. An anonymous authentication scheme with privacy-preserving is proposed in [12] for the RFID system. In this scheme, the ideal Physically Uncloneable Function (PUF) is considered to assure a tamper-evident feature. Moreover, in order to support the noisy PUF environment, they also introduced an enhanced scheme. In [13], a authentication based on ECC and biometric with privacy-preserving is developed for IIoT network. Although, the mentioned works may be effective in supporting the security of data, however, network congestion that occurs because of the large number of data exchanged among the parties, has not been considered.

In order to deal with network congestion issue, it is required to prevent exchanging untrustworthy data over the network. In smart industries, trustworthiness of data collected/sensed/measured by IIoT nodes/devices is one of the main concerns. Therefore, IIoT devices require a continuous behavior monitoring by using the trust model. A scalable trust model solution based on clustering is proposed [14] for IoT environments. A master node manages each cluster node's metric, which is gathered from peer cluster nodes. Through an algorithm, outliers can be eliminated, then, the overall trust value is determined as an average. However, this study is vulnerable to coalition attacks since the suggested algorithm decides in terms of the evaluation given by the cluster nodes majority. As a majority, the suggested algorithm contains the evaluations presented by the malevolent cluster nodes. Thus, the assessments presented by the fair and good nodes are eliminated. There is no accuracy for the model in calculating trust values since the assessments are not performed in a well-defined context. Therefore, having a context between the monitoring and monitored the nodes is essential to have a high accuracy for trust management. In [15], a trust model based on cloud theory, as a security mechanism, is proposed for underwater acoustic sensor networks. In this scheme, there exist a three step approach to evaluate the trust value of a sensor node: trust evidence generation, direct trust measurement, and indirect trust measurement. The intention of the proposed trust model is to improve the accuracy of trust evaluation and address uncertainty and fuzziness of trust.

Taking into account the presence of faulty sensor/smart devices in the network, broadcasting untrustworthy and unreliable data can lead to reducing the system performance and/or affecting the security of the system. In addition, confidentiality and integrity of data-in-transit is threatened by different types of adversaries. To the best of our knowledge, there is a lack of a proper scheme in IIoT network with low computation and communication cost that considers both trustworthiness of data and the security of the data-in-transit, simultaneously.

3 Preliminaries

In this section, we define the network architecture, the main entities involved in this architecture, the security requirements that we aim to achieve, and the threat model.

3.1 System Model

In this study, we designed a layered network architecture comprising of Trusted Authority (TA), Cloud Server (CS), GateWay (GW), Sensor Nodes (SN), and Users (U). In this architecture, TA and CS are taken into account as fully trusted entities. It is believed that all communication between the TA and CS is routed through a secure way by using wired communication technologies; whereas connection between U and GW, and between U/GW and CS is under 5G communication standard where it allows for high-speed data transmission. TA serves as the registration center of the network and distributes the required materials to all participating entities. In the designed network, CS is responsible to check the authenticity of nearby users and gateways. It generates the required parameters to create the session key and shares them between the authorized user and the gateway. GW can be a router or smart hub providing communication with all sensors and stores the measured data into its storage. An authorized user via a mobile device sends its request to the GW in order to access to the allowable data. In this work, SNs are scattered in a sensing field. These nodes sense/collect data as per the functionality of the devices and send to the GW.

3.2 Security Requirements

In IIoT network, the faulty/tampered smart devices/nodes might be threaten the network by generating the wrong data. In addition, given the open nature of the wireless medium that exists between CS, GW and U, an attacker is able to obtain and also tamper the data.

To tackle these concerns, the integration of a trust model and an authentication scheme can be effective. This security scheme needs to meet the main security requirements. In the following, the most important prerequisites and requirements for a security scheme are summarized:

- **Mutual Authentication:** All entities participating in the network should be able to authenticate each other before starting communication and initiating data sharing.
- **Data Integrity:** It ensures the accuracy, and legitimacy of the data.
- **Data Confidentiality:** It ensures that an attacker is unable to learn and extract anything about the original data.
- **User's Anonymity:** It assures that an attacker will not be able to obtain the legal user's real identity throughout the authentication procedure and as a result, keep the real identity hidden.
- **Data Trustworthiness:** It ensures that all data exchanged between users and gateway are trustworthy.

3.3 Threat Model

The open nature of wireless communication in the 5G-IIoT networks makes it an ideal medium for different and complicated security threats such as eavesdropping, manipulating, and repudiation. The smart devices/sensor nodes are also may be compromised by different attacks and as a result, broadcast false or wrong data in the network. Therefore, it is important to ensure the security of data-in-transit as well as the trustworthiness of nodes and in addition data generated by nodes. In this work, we employ the Dolev-Yao (DY) threat model. Based on DY, $\mathbf{\Lambda}$ is able to read/modify/delete/insert messages during communication between two parties [16] and hence, threatens data integrity and data trustworthiness.

4 TRUTH: A combination of trust model and authentication scheme

As described in [17], the trustworthiness of data sources and data-in-transit is the first step to provide the crucial requirements of security of data. According to [18], trust establishment among entities of a distributed network is also a suitable tool to enhance the security of the network. In other words, trust is an essential complementary function to security schemes. In this study, we introduce a trust model and an authentication scheme which are the core of our security scheme. The trust model is used to check reliability and trustworthiness of data collected by sensors/smart devices. A security scheme based on AKE protocol is proposed to negotiate a session key between the user and gateway with the help of cloud server. This scheme can be bootstrapped by users when one needs to access data stored in the gateway storage. Before access to data, authentication of the user and gateway should be checked by the related cloud server. In the following, we explain the proposed trust model and authentication scheme in detail.

4.1 Proposed Trust Model

The trustworthiness of measured or sensed data by sensor nodes is a security challenge in IIoT. In this network, the faulty and malicious sensor nodes may provide false information that led to compromise the entire system. To address this issue, trust establishment among smart devices, as a powerful tool, can improve the security and performance of the system.

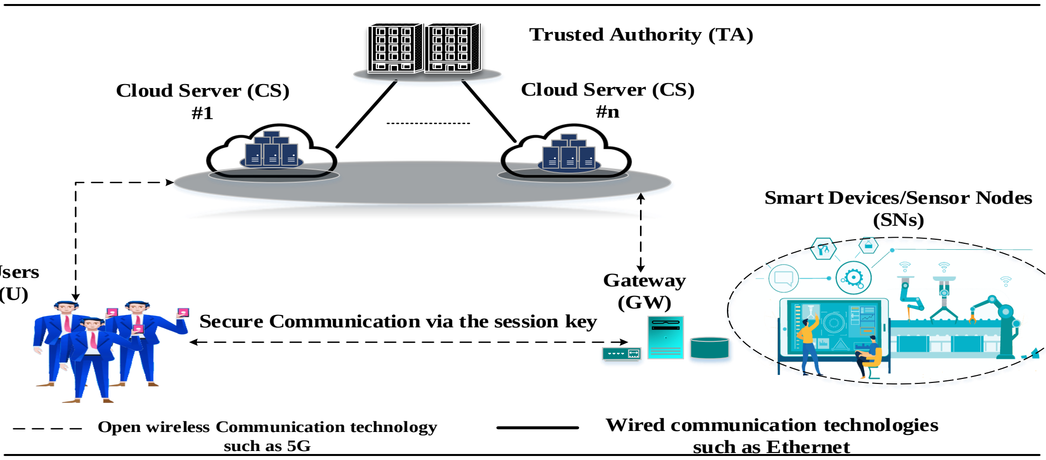

Trusted Authority (TA)
Cloud Server (CS) #1
Cloud Server (CS) #n
Smart Devices/Sensor Nodes (SNs)
Users (U)
Gateway (GW)
Secure Communication via the session key
Open wireless Communication technology such as 5G
Wired communication technologies such as Ethernet

Figure 1: The proposed Cloud-IIoT architecture.

In this study, we proposed a lightweight trust model to evaluate the trustworthy of smart devices/sensor nodes. The proposed trust model is based on both Direct Trust (DT) and Recommended Trust (RT) in which DT depends on direct interaction between sensor nodes and GW; whereas RT is based on the recommendations of nearby nodes. Since sensor nodes/smart devices might be threaten by malicious nodes, therefore, relying on only direct trust cannot be effective. To deal with this issue, we take into account both direct and recommended trust values. Due to recommendations generated by different independent sources, it needs to combine such degrees of belief. To that end, Dempster Shafer Theory (DST) can be a useful method.

Due to the designed network architecture, GW, as the base station and data storage, is responsible to collect the sensed data and records data in the device storage. In this architecture, it is believed that GW visits all sensor nodes to gather data through single-hop and direct communication. Therefore, it has to evaluate the value of trust of the available sensor nodes. To this end, GW uses the below equation:

$$T_{GW,SN_i} = w_1 \times DT_{GW,SN_i} + w_2 \times RT_{SN_i} \quad (1)$$

where w_1 and w_2 are respectively the considered weights for direct trust and recommended trust such that $w_1 + w_2 = 1$. DT_{GW,SN_i} indicates direct trust value and RT_{SN_i} refers to recommended trust value of SN_i evaluated by nearby sensor nodes/smart devices SN_j

$$RT_{SN_i} = \frac{\sum_{j=1}^{N} DT_{SN_j,SN_i}}{N}$$

4.1.1 Direct Trust

In this work, the number of successful and failed interactions are used to calculate direct trust value. The history of past interactions is also considered another parameter of direct trust. According to the beta-function-based method [19], the direct trust value is based on only s, as a number of successful interactions, and f, as a number of failed interactions between GW and SN_i. This value in n-th attempts and during a time period t_{n-1} and t_n can be calculated as

$$DT^{n}_{GW,SN_i} = \frac{s+1}{s+f+2} \quad (2)$$

However, because of the impact of network congestion, delay, bandwidth limitation and other factors on interaction between network entities, it is suitable to consider previous trust values as another parameter to calculate current direct trust value. Consider GW holds m of the last direct trust values with each sensor node. Based on the history and the number of successful and failed interactions in the last period of time, the current direct trust value can calculate as following:

$$\begin{aligned} DT_{GW,SN_i} & \\ &= \frac{\alpha_1 DT^{1}_{GW,SN_i} + \alpha_2 DT^{2}_{GW,SN_i} + \cdots + \alpha_m DT^{m}_{GW,SN_i}}{\alpha_1 + \alpha_2 + \cdots + \alpha_m} \\ &= \frac{\sum_{b=1}^{m} \alpha_b DT^{b}_{GW,SN_i}}{\sum_{b=1}^{m} \alpha_b} \end{aligned} \quad (3)$$

where DT^{b}_{GW,SN_i} is the b^{th} of the last m stored direct trust values and $0 < \alpha_b < 1$ is its weight such that $\alpha_1 < \alpha_2 < \cdots < \alpha_m$ which describes that the trust value made more recently are more importance.

4.1.2 Recommended Trust

Recommendations from neighbour nodes have a vital role in determining the trustworthiness of a node and increasing the accuracy in trust computation. In this work, each sensor node sends recommendations about common neighbor sensor nodes where the neighbors' recommendations are independent of each other. Here, DST is used to combine the degree of beliefs received on particular node from multiple neighbor sensor nodes. The output of DST combination for each sensor node determines the value of recommended trust for the sensor node.

Consider $\Omega = \{trustworthy, untrustworthy\}$ as a power set that shows the state of each sensor node: trustworthy and untrustworthy. Let $H_1 = \{\emptyset\}$, $H_2 = \{trustworthy\}$, $H_3 = \{untrustworthy\}$, and $H_4 = \Omega$ be the four hypotheses in our scenario where each hypothesis is assigned a basic probability value $m(H_i) \rightarrow [0, 1]$. It is believed that GW can evaluate the recommended trust value of each sensor node SN_i through evidence provided by one-hop neighbor nodes. To this end, each one-hop

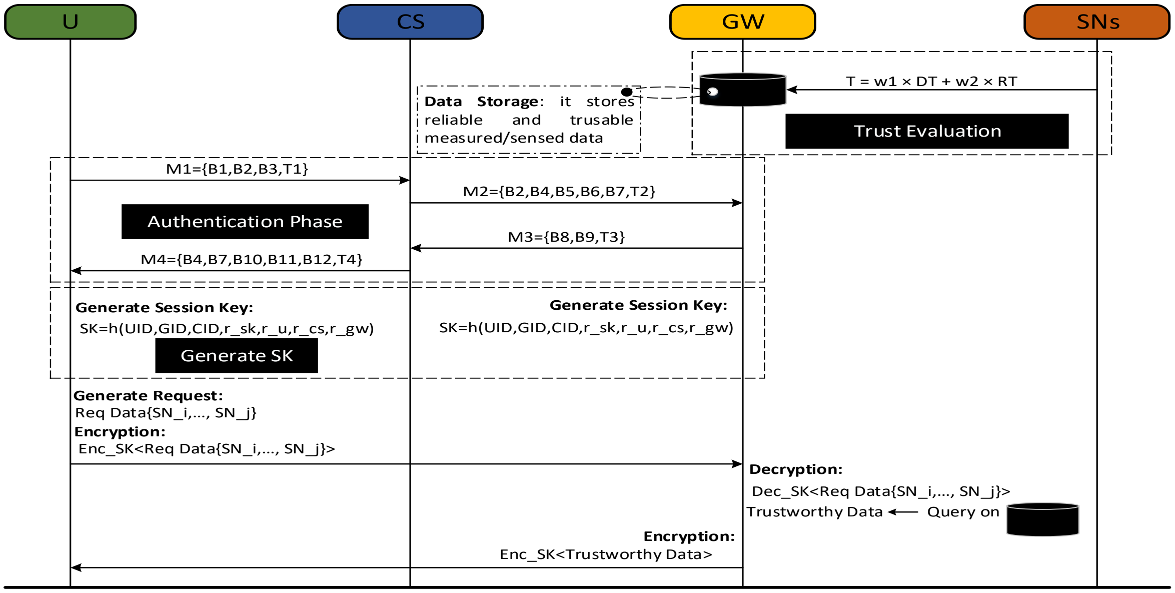


Figure 2: A summary of the whole process of TRUTH.

neighbor sensor node gives evidence from its observation by assigning its beliefs over Ω, i.e., node SN_i is trustworthy. Assume the direct trust value between GW and SN_i, observed by GW, is DT_{GW,SN_i}. If sensor node SN_i believes that node SN_k is trustworthy, then the basic probability value $m_{SN_i}(H_2)$ is DT_{GW,SN_i} and as a result the basic probability value $m_{SN_i}(H_3)$ is 0. From the definition of belief function, $m_{SN_i}(H_4)$ is equal to $1 - DT_{GW,SN_i}$. In contrast, if sensor node SN_i believes that node SN_k is an untrustworthy sensor node, in result $m_{SN_i}(H_2) = 0$, $m_{SN_i}(H_3) = DT_{GW,SN_i}$, and $m_{SN_i}(H_4) = 1 - DT_{GW,SN_i}$.

Following the Dempster combination rule, it is possible to combine more results from neighbor nodes. Therefore, the recommended trust value RT_{GW,SN_i} is defined as $RT_{GW,SN_k} = m_{SN_{i1}}(H_2) \oplus m_{SN_{i2}}(H_2) \oplus \cdots \oplus m_{SN_{in}}(H_2)$ where $SN_{i1}, SN_{i2}, \cdots, SN_{in}$ are the one-hop neighbor sensor nodes between GW and SN_i.

4.2 Proposed Authentication Scheme

In IIoT network, the data collected by smart devices/sensor nodes need to be transferred to the users in a secure manner. Since the symmetric encryption algorithms are faster than asymmetric algorithms, hence, we employed this method to make secure communication between two parties i.e., user and gateway. In this work, we used the AKE protocol to securely share the secret key between two parties through a fully trusted entity. Here, we explain our scheme in detail. This scheme consists of the initialization phase, the user pseudo-identity generation, and the authentication phase.

4.2.1 Initialization Phase

In this phase, the system parameters will be released by TA to all authorized entities in the network such as users, cloud servers, and gateways. In this network, all entities agree on the group G of order q and the generator element P. To generate system parameters, TA selects $s \in Z_q^*$ as the system private key and then calculates $P_{pub} = s.P$ where P_{pub} refers to the system public key, q is a prime number. In addition, TA selects $\psi \in Z_q^*$ as the master secret key. It sets system parameters $SysPara = \{q, E_p, P, G, P_{pub}, h\}$ where $h : \{0,1\}^* \rightarrow Z_q^*$ is a one-way hash function and $E_p\,(a,b) : y^2 = x^3 + ax + b \ \ mod \ \ p$ is a non-singular elliptic curve. To use $E_p(a,b)$, all parties must agree on the elements $\{a, b, p, G\}$ that define the elliptic curve.

In this network architecture, each cloud server CS_i has a real identity ID_{CS_i}, private key $s_{cs_i} \in Z_q^*$ and public key $Q_{CS_i} = s_{cs_i}.P$. Each gateway GW_i with real identity ID_{GW_i} and pseudo-identity $GID_{GW_i} = h\,(ID_{GW_i} \parallel s_{gw_i} \parallel \psi)$ has $s_{gw_i} \in Z_q^*$ as private key and then computes the public key $Q_{GW_i} = s_{gw_i}.P$. For each user U_i, TA assigns $SK_{U_i} \in Z_q^*$ as private key and compute the public key $PK_{U_i} = SK_{U_i}.P$, real identity ID_{U_i} and PWD_{U_i}.

4.2.2 User Pseudo-Identity Generation

In this work, user anonymity is one of the security requirements. To satisfy this requirement, each user must employ pseudo-identity in order to communicate with other entities. To this end, user U with real identity ID_U and password PWD_U, generates $UID_U = ID_U \oplus h\,(r.P_{pub})$ as pseudo-identity where $r \in Z_q^*$ is random number. User also sends $\{ID_U, PWD_U, UID_U\}$ to TA in order to calculate $SID_U = h\,(UID_U \parallel \psi)$. Finally, TA sends SID_U and $\{UID_U, SID_U\}$ to user and the relevant cloud server CS, respectively. The generated pseudo-identities are valid for a limited time (VT_{PID}) and after the lifetime, a new pseudo-identity should be generated.

4.2.3 Authentication Phase

To fulfill mutual authentication, it is required to perform the process of authentication among all communicating parties. Here, the authentication process and session initiating among U, GW, and CS have been described.

Step 1 - U2CS Communication: In order to connect to a

```
role user(U,CS,GW:agent, SK:symmetric_key,
Mul,Add,H:hash_func, RCV, SND_UCS, SND_UTA,
RCV_CSU:channel(dy))

played_by U

def =
local State: nat, IDu, UIDu, GIDgw, CIDcs, PP

Inc: hash_func

const user_cloudserver_tu, user_cloudserver_ru,
sub1, sub2, sub3, sub4, sub5:protocol_id



init State:= 0

transition

1. State = 0 /\ RCV (start) =|> State':= 1
/\ Ru':= new() /\ T1':= new()
/\ B1' = xor(Ru',H(SIDu.UIDu))
/\ B2' = xor(H(Ru'),UIDu)
/\ B3' = H(SIDu.Ru',UIDu)
/\ SND_UCS(B1'.B2'.B3'.T1
/\ secret({Ru'},sub1,{U,CS})
/\witness(U,CS,user_cloudserver_tu,T1')
/\ witness(U,CS,user_cloudserver_ru,Ru')

2. State = 1 /\
RCV_CSU(B4'.B7'.B10'.B11'.B12'.T4') =|>
State':= 2 /\ Rsk' = xor(B7',Rcs',Ru')
/\ SK' =
H(UIDu,Ru',Rcs',Rgw',Rsk',CIDcs',GIDgw')
/\ secret({SK'},sub2,{U,GW})


end role
```

```
role cloud(U,CS,GW:agent, SK:symmetric_key,
Mul,Add,H:hash_func, SND_CSGW, RCV_GWCS,
SND_CSU,RCV_UCS:channel(dy))

played_by CS

def =
local State: nat, UIDu, GIDgw, CIDcs,
Inc: hash_func

const cloudserver_gateway_tcs,
cloudserver_gateway_rcs,
gateway_cloudserver_tgw
sub1, sub2, sub3, sub4, sub5:protocol_id

init State:= 0
transition
1. State = 0 /\ RCV_UCS (B1'.B2'.B3'.T1') =|>
State':= 2 /\ Rcs':= new()
/\ Rsk':= new()/\ T2':= new()
/\ B4' = xor(Rcs',H(GIDgw))
/\ B5' = xor(Ru',H(GIDgw.CIDcs))
/\ B6' = H(GIDgw.Rcs',CIDcs)
/\ B7' = xor(Rsk'',Rcs',Ru')
/\ SND_CSGW(B2'.B4'.B5'.B6'.B7'.T2')
/\ secret({Rcs',Ru',Rsk'},sub3,{CS,GW})
/\ witness(U,CS,cloudserver_gateway_tcs,T2')
/\ witness(U,CS,cloudserver_gateway_rcs,Rcs')

2. State = 2 /\ RCV_GWCS(B8'.B9'.T4') =|>
State':= 4 /\ T4' = new()
/\ B10'= xor(H(Ru'),GIDgw)
/\ B11'= xor(Rgw',H(SIDu'))
/\ B12'= H(UIDu.Rcs'.Rgw2'.GIDgw')
/\ SND_CSU(B4'.B7'.B10'.B11'.B12'.T4')
/\ secret({Rcs',Rgw'},sub4,{CS,U})
/\ witness(GW,CS,gateway_cloudserver_tgw,T4')

end role
```

```
role gateway(U,CS,GW:agent,
SK:symmetric_key, Mul,Add, H:hash_func,
SND_GWCS,RCV_CSGW:channel(dy))


played_by GW

def =
local State: nat, UIDu,GIDgw,CIDcs,

Inc: hash_func

const gateway_cloudserver_tgw,
sub1, sub2, sub3, sub4, sub5:protocol_id



init State:= 0

transition


1. State = 0 /\
RCV_CSGW(B2'.B4'.B5'.B6'.B7'.T2') =|>
State':= 3 /\ Rgw':= new() /\ T3':= new()
/\ B8' = xor(Rgw',H(CIDcs))
/\ B9' = H(CIDcs.Rgw'.GIDgw)
/\ SND_GWCS(B8'.B9'.T3')
/\ Rsk' = xor(B7',Rcs',Ru')
/\ SK' =
H(UIDu,Ru',Rcs',Rgw',Rsk',CIDcs',GIDgw')
/\ secret({SK'},sub5,{U,GW})
/\witness(GW,CS,gateway_cloudserver_tgw,T3')



end role
```

Figure 3: The HLPSL specification for the user, cloud server and gateway roles.

cloud server, user U_i should create a login request. To this end, it randomly chooses $r_u \in Z_q^*$ and calculates

- $B_1 = r_u \oplus h\left(UID_{U_i} \parallel SID_{U_i}\right)$
- $B_2 = h(r_u) \oplus UID_{U_i}$
- $B_3 = h(SID_{U_i} \parallel r_u \parallel UID_{U_i})$

Then, U_i submits the login request $M_1 = \{B_1, B_2, B_3, T_1\}$ to CS_l where T_1 is the current timestamp. Before sending M_1, user signs M_1 by its private key $SG_{SK_{U_i}}(M_1)$ and then encrypt it using cloud server's public key $ENC_{Q_{CS_l}}(M_1, SG_{SK_{U_i}}(M_1))$.

Step 2 – CS2GW Communication: Upon receiving a login request from U_i, CS_l decrypts the message by using its private key s_{cs_l} and verifies the signature by user's public key PK_{U_i}. After verification, CS_l has to check the freshness of timestamp T_1. The login request will be rejected if $\triangle T < T_{CS_l} - T_1$ where T_{CS_l} is the current timestamp of the CS_l. Otherwise, if $\triangle T \geq T_{CS_l} - T_1$, it computes $SID^*_{U_i} = h(UID_{U_i} \parallel \psi)$, $r^*_u = B_1 \oplus h\left(SID^*_{U_i} \parallel UID_{U_i}\right)$, and $B^*_3 = h\left(SID^*_{U_i} \parallel r^*_u \parallel UID_{U_i}\right)$. Next, it checks whether $B_3 \overset{?}{=} B^*_3$. If it holds, CS_l generates a request and send it to the gateway i.e. GW_j, otherwise, rejects the login request and shut down the communication channel. To generate the request for GW_j, it selects two random numbers $r_{cs}, r_{sk} \in Z_q^*$ and computes

- $B_4 = r_{cs} \oplus h(GID_{GW_j})$
- $B_5 = r^*_u \oplus h\left(GID_{GW_j} \parallel CID_{CS_l}\right)$
- $B_6 = h\left(GID_{GW_j} \parallel r_{cs} \parallel CID_{CS_l}\right)$
- $B_7 = r_{sk} \oplus r_{cs} \oplus r_u$

Finally, it sends the request message $M_2 = \{B_2, B_4, B_5, B_6, B_7, T_2\}$ to GW_j where T_2 is current timestamp. Prior sending the message, CS_l signs M_2 by its private key $SG_{s_{cs_l}}(M_2)$ and encrypt it using gateway's public key $ENC_{Q_{GW_j}}(M_2, SG_{s_{cs_l}}(M_2))$.

Step 3 – GW2CS Communication: Upon GW_j received a request from the CS_l, it firstly needs to decrypt the message by using its private key s_{gw_j} and verifies the signature by cloud server public key Q_{CS_l}. After verification, GW_j checks the freshness of timestamp T_2. If it is not fresh, the request will be rejected. Otherwise, it extracts CID_{CS_l} from the list of authorized cloud servers $\{CS_1 : CID_{CS_1}, \cdots, CS_n : CID_{CS_n}\}$. Each gateway securely received this list from the cloud server and stored it with care to prevent information leakage to attackers. Then, it calculates $r^*_{cs} = B_4 \oplus h\left(GID_{GW_j}\right)$, $r^{**}_u = B_5 \oplus h\left(GID_{GW_j} \parallel CID_{CS_l}\right), UID^{**}_{U_i} = B_2 \oplus h\left(r^{**}_u\right)$, and $B^*_6 = h\left(GID_{GW_j} \parallel r^*_{cs} \parallel CID_{CS_l}\right)$. If $B_6 \neq B^*_6$, the request is terminated. Otherwise, if $B_6 = B^*_6$, CS_l is authenticated by GW_j. Next, GW_j selects a random number $r_{gw} \in Z_q^*$ and computes

- $B_8 = r_{gw} \oplus h\left(CID_{CS_l}\right)$
- $B_9 = h\left(CID_{CS_l} \parallel r_{gw} \parallel GID_{GW_j}\right)$

Next, it submits $M_3 = \{B_8, B_9, T_3\}$ to CS_l where T_3 is the current timestamp. Similarly, GW_j signs M_3 by its private key $SG_{s_{gw_j}}(M_3)$ and encrypt it using cloud server's public key $ENC_{Q_{CS_l}}(M_3, SG_{s_{gw_j}}(M_3))$ prior sending the M_3.

Step 4 – CS2U Communication: Once CS_l received the M_3 from the GW_j, it firstly needs to decrypt the message by using its private key s_{cs_l} and verifies the signature by gateway public key Q_{GW_j}. If it is verified, CS_l checks the freshness of timestamp T_3. If it is not fresh, the request will be rejected. Otherwise, it computes $r^*_{gw} = B_8 \oplus h\left(h\left(ID_{CS_l} \parallel s_{cs_l} \parallel \psi\right)\right)$. Then, it calculates $B^*_9 = h\left(CID_{CS_l} \parallel r^*_{gw} \parallel GID_{GW_j}\right)$. If $B_9 \neq B^*_9$, the request will be terminated by CS_l. Otherwise, GW_j is authorized and hence CS_l calculates

- $B_{10} = h(r_u) \oplus GID_{GW_i}$
- $B_{11} = r^*_{gw} \oplus h(SID^*_{U_i})$

- $B_{12} = h\left(UID_{U_i} \| r_{cs} \| r_{gw} \| GID_{GW_j}\right)$

Then, it sends the $M_4 = \{B_4, B_7, B_{10}, B_{11}, B_{12}, T_4\}$ to U_i where T_4 is current timestamp. Before sending the message, CS_l similarly signs M_4 by its private key $SG_{s_{cs_l}}(M_4)$ and encrypt it using user's public key $ENC_{PK_{U_i}}(M_4, SG_{s_{cs_l}}(M_4))$.

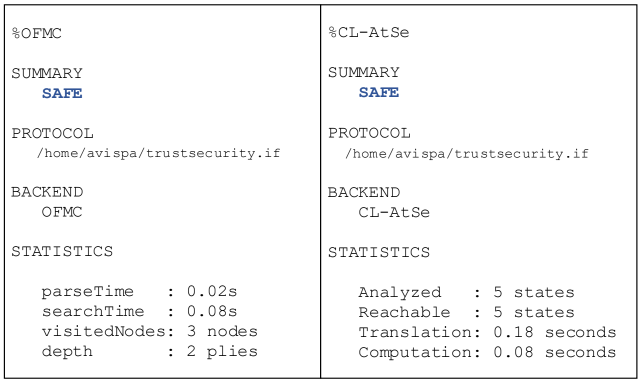
Step 5: : Upon receiving M_4 from the CS_l, U_i decrypts the message by using its private key SK_{U_i} and verifies the signature by cloud server public key Q_{CS_l}. After verification, it checks the freshness of T_4. If it is not fresh, the request will be rejected. Otherwise, it computes $GID^*_{GW_j} = B_{10} \oplus h(r_u)$, $r^{**}_{gw} = B_{11} \oplus h\left(SID_{U_i}\right)$, $r^{**}_{cs} = B_4 \oplus h(GID^*_{GW_j})$, and $B^*_{12} = h\left(UID_{U_i} \| r^{**}_{cs} \| r^{**}_{gw} \| GID^*_{GW_j}\right)$. Then, U_i checks whether $B_{12} \overset{?}{=} B^*_{12}$. If does not hold, the session is terminated. Otherwise, CS_l and in result GW_j are authenticated by U_i. Hence, U_i and GW_j can establish a session communication securely via symmetric encryption by using a session key SK. To this end, U_i and GW_j generate $SK = h\left(UID_U \| r_u \| r_{cs} \| r_{gw} \| r_{sk} \| CID_{CS} \| GID_{GW}\right)$ where $r_{sk} = B_7 \oplus r_{cs} \oplus r_u$. This key will be used for the encryption and decryption of data sent and received between the two parties. Besides, SK is usable for a limited time and will expire after the lifetime that we explain next. Figure 2 shows the complete process of the proposed authentication scheme. As shown in this figure, data generated/measured by sensor nodes/smart devices should be assessed by the gateway in terms of trustworthiness. In other words, all data stored in the data storage of the gateway are reliable and trustable. In addition, the integrity of trustable data will be supported by using the session key generated in the authentication phase.

5 Formal Security Analysis and Verification

In this section, we formally prove the validity of our security scheme by using the AVISPA as a popular tool. As explained in [20], AVISPA is a state-of-the-art tool to analyse a security protocol against adversaries. In this work, our scheme will be modeled using AVISPA by the High-Level Protocol Specification Language (HLPSL) and the role specifications of the user, gateway, and cloud server. Figure 3 shows the HLPSL specification of the role of the user, cloud server, and gateway. The adversary is also modeled by using the DY. We also used On-the-Fly Model-Checker (OFMC) and Constraint Logic-based Attack Searcher (CL-AtSe) as two back-ends integrated into AVISPA to check the DY model when the man-in-the-middle attack occurred. Figure 4 illustrates that OFMC and CL-AtSe have found no attacks which show the safety of the proposed scheme.

6 Non-mathematical Security Analysis and Discussion

In the following, a discussion is provided to show how our scheme achieves the security requirements. Next, we compare our scheme with respect to the existing related works for IIoT environment [13], [21], [22], and [23].



```
%OFMC                                     %CL-AtSe

SUMMARY                                   SUMMARY
   SAFE                                      SAFE

PROTOCOL                                  PROTOCOL
  /home/avispa/trustsecurity.if             /home/avispa/trustsecurity.if

BACKEND                                   BACKEND
   OFMC                                      CL-AtSe

STATISTICS                                STATISTICS

   parseTime    : 0.02s                      Analyzed    : 5 states
   searchTime   : 0.08s                      Reachable   : 5 states
   visitedNodes: 3 nodes                     Translation: 0.18 seconds
   depth        : 2 plies                    Computation: 0.08 seconds
```

Figure 4: Simulation results under CL-AtSe and OFMC back-end.

6.1 Mutual Authentication

In this scheme, we provide mutual authentication among U, CS, and GW. As explained in Section 4.2.3, CS and U can authenticate each other by checking $B_3 \overset{?}{=} B^*_3$ and $B_{12} \overset{?}{=} B^*_{12}$. In other words, CS authenticates U by using the B_3, and in contrast, U authenticates CS by using the B_{12}. Similarly, the mutual authentication between CS and GW can be achieved by checking $B_6 \overset{?}{=} B^*_6$ and $B_9 \overset{?}{=} B^*_9$. Moreover, U is an authorized entity to GW if and only if U is authenticated by CS. Besides, U and GW maintain an authentication session once the session key is established. As a result, mutual authentication between all parties will be established.

6.2 Data Integrity

In this scheme, all data-in-transit will be encrypted and decrypted by the agreed session key (SK) among the authorized parties U and GW. It is also assumed that SK is kept securely. Hence, if the attackers attempt to send tampered data to U and or GW, they need SK to encrypt the tampered data. Without SK, the attackers would be detectable by the authorized parties and the tampered data will be discarded. This enables the proposed scheme to provide data integrity.

6.3 Data Confidentiality

Our scheme provides end-to-end encryption for all sessions between U and GW by using the secret key SK. Due to the symmetric encryption scheme (i.e., AES) employed in this work, the attacker is unable to extract/learn information from the encrypted data-in-transit. It means, the attacker only may capture the encrypted data. Therefore, our scheme can provide data confidentiality.

6.4 User's Anonymity

In this work, to meet privacy-preserving, we attempted to hide the user's real identity from other entities and keep it private and secret. For each communication session, user U_i uses it's pseudo-identity UID_{U_i} and SID_{U_i}. Based on the Discrete Logarithm Problem (DLP), the attacker $\mathbf{\Lambda}$ is unable to extract ID_{U_i} from $UID_{U_i} = ID_U \oplus h(r.P_{pub})$, since r_i

Table 1: Comparison of security and functional features.

Security Attributes	[13]	[21]	[22]	[23]	Our
Mutual Authentication	●	●	●	●	●
Data Integrity	●	●	●	●	●
Data Confidentiality	●	●	●	●	●
User Anonymity	●	●	●	●	●
Data Trust	○	○	○	○	●

Table 2: Cryptographic operations .

Notation	Description
T_{sm}	scalar-point multiplication operation
T_h	on-way hash function
T_{fe}	fuzzy extractor function
T_{xor}	XOR operation function
T_{cm}	Chebysev chaotic-map

Table 3: Elements used in authentication phase.

Element	Size (bits)
ID_{GW}, ID_{CS}	160
$h(.)$	160
$x \in Z_q^*$	160
ID_U	80
T	32
$P \in G$	320
Block of AES	128

Table 4: Performance comparisons.

Ref.	Computation Cost	Communication Cost
[13]	$19T_h + 8T_{xor} + 6T_{sm} + T_{fe}$	340 Bytes
[21]	$30T_h + 16T_{xor} + T_{fe}$	482 Bytes
[22]	$31T_h + 20T_{xor} + 4T_{cm} + T_{fe}$	232 Bytes
[23]	$14T_h + 8T_{xor} + T_{fe}$	340 Bytes
Our Scheme	$25T_h + 17T_{xor}$	316 Bytes

selected by U_i and ID_U are hidden and secure and it is hard to compute the r_i of the user through UID_{U_i} and P. Since a user frequently changes pseudo-identity, hence, it is difficult to trace a user and the relationship between these pseudo-identities can be revealed via TA only.

Definition 1: *Considering G as additive elliptic curve group, two random numbers $P, Q \in G$ on E_q in which $Q = x.P$ and $x \in Z_q^*$. Based on the Discrete Logarithm Problem (DLP), it is hard to measure x from Q.*

6.5 Data Trustworthiness

The collected data by sensor nodes and or smart devices will be evaluated in terms of trustworthiness and reliability by using our proposed trust model. It ensures that only trusted data will be stored in the gateway storage and hence, a user has access to the trusted data. Therefore, our scheme resists against untrustworthy data collected by faulty/malicious sensor nodes.

The comparison reveals that our scheme satisfies all of the security criteria, whereas similar schemes only satisfy a portion of the security requirements. As a consequence, our scheme outperforms the comparable schemes in terms of security and functionality. Table 1 represents a comparison of security features.

7 Performance Analysis

The computation and communication costs are used to evaluate the scheme's performance.

Computation Cost: Because of the different network architecture utilized in our scheme and the comparable schemes, the computational cost of each entity and/or each layer needs to be calculated, separately. For simplicity of analysis, we firstly extracted the cryptographic operations utilized in our scheme and other related works (see Table 2).

In terms of authentication, the total cost of computation for our scheme compromise of 25 hash function operations and 17 XOR operations $25T_h + 17T_{xor}$ wherein $7T_h + 5T_{xor}$ is on the user side in the, $12T_h + 8T_{xor}$ in the cloud server and $6T_h + 4T_{xor}$ in the gateway.

Communication Cost: For convenience, we extracted the size of all the elements utilized in our scheme and other comparable schemes (see Table 3) and then calculated the number of bits of all messages exchanged during the authentication phase. Accordingly, our scheme consumes the number of bits in four messages M_1, M_2, M_3, and M_4 such that $|M_1|= (160 + 160 + 160 + 32)$, $|M_2|= (160 + 160 + 160 + 160 + 160 + 32)$, $|M_3|= (160 + 160 + 32)$, and $|M_4|= (160+160+160+160+160+32)$ and the cumulative communication cost consumed is 2528 bits. Table 4 presents the obtained computation and communication costs related to our scheme and other related schemes.

8 Practical Perspective

We used OMNET++ to simulate the designed network architecture [24]. In this study, a real-life scenario with 100 sensor devices, 1 gateway, 1 node as cloud server, and 5 users is simulated for 30 minutes.

8.1 Authentication Scheme Evaluation

We evaluated our scheme practicality by using two parameters, End-2-End Delay (E2ED) and Network THRoughput (NTHR).

End-to-End Delay: This parameter assesses the performance of the network. It is based on the average time taken by the message from the sender to the receiver. This parameter is defined as

$$E2ED = \frac{\sum_{i=1}^{N} (T_{r_i} - T_{s_i})}{N}$$

where T_{r_i} refers to the receiving time of the packet i and T_{s_i} is the time of sending packet i and N is the number of packets in total .

As shown in Figure 5, E2ED for our scheme, during the authentication process, is $0.044\ ms$, whereas it is $0.058\ ms$, $0.056\ ms$, $0.071\ ms$, and $0.043\ ms$ respectively for [13], [21], [22], and [23]. This figure shows that [23] has less end-to-end delay than the proposed scheme, [13], [21] and [22]. This is mainly because the message size in [23] is smaller than others.

Network Throughput: This parameter is defined as the number of bits exchanged over the network per unit time. It is formulated as

$$NTHR = \frac{N_r \times |Pkt|}{T_d}$$

where $|Pkt|$ is the size of each packet per bit, N_r is the total number of received packet, and T_d is the total time expense for this transmission.

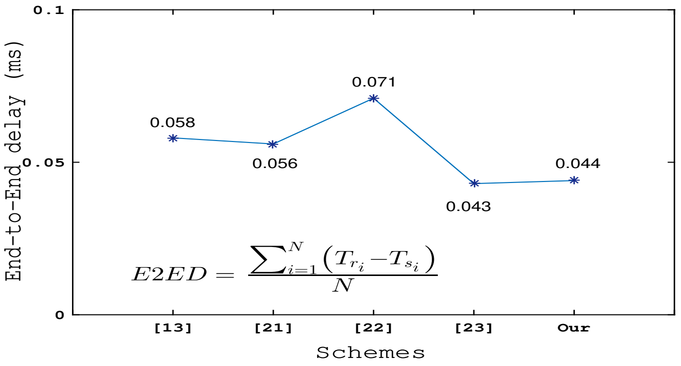


Figure 5: End-to-End delay

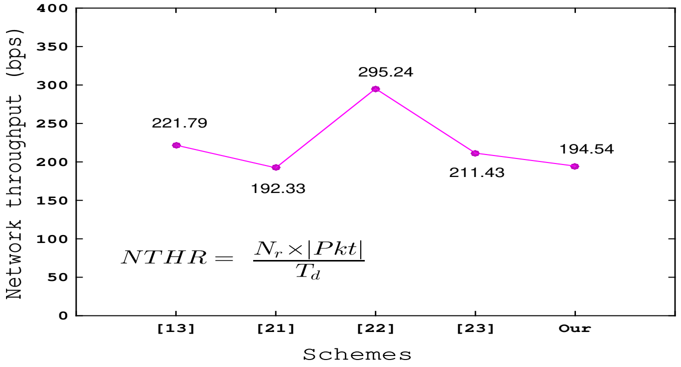


Figure 6: Network throughput

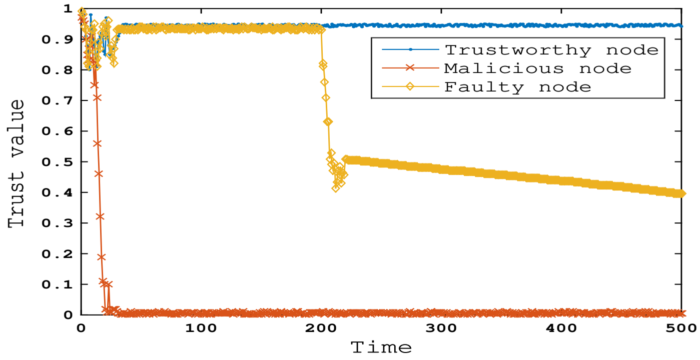


Figure 7: Trust value

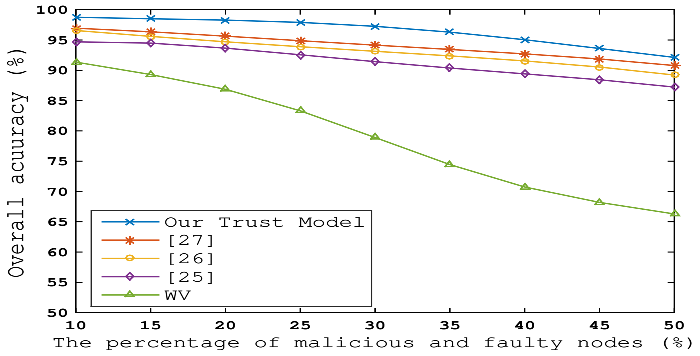


Figure 8: Overall accuracy

As shown in Figure 6, the proposed scheme in [21] has lowest throughput. Moreover, the throughput of the proposed scheme is less than [13], [22], and [23]. This is because of the less-sized messages in the authentication process.

8.2 Trust Model Evaluation

In this section, the performance of our trust model is evaluated. To this end, two indexes, the trust value and overall accuracy, are taken into account.

Trust Value: It reflects the value of trust of sensor node/smart device measured by our trust model. Here, we show the comparison of the measured trust value of a legitimate and trustworthy SN against a malicious and faulty SN. To this end, the simulated malicious attack is a data forgery attack in which a malicious node broadcasts forge data to the network. And, the simulated faulty node initially acts as a legitimate node and toggles its behaviour after a specific time and broadcast wrong data to the gateway. The wrong data here means data related to the sensor but outside the logical and actual range of the sensor. As we can see in Figure 7, the trust value of a trustworthy SN is a high value, whereas this amount for a malicious node dramatically reduces and become 0. The trust value of a faulty SN is high till it sends proper data to the gateway, but this amount gradually decreases and become almost constant whenever detected as a faulty node by our trust model. We consider 500 seconds to run the simulation.

Overall Accuracy: It reflects the proportion of the overall number of accurate findings. The following equation is used to computes the overall accuracy.

$$Acuracy(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

where TP and TN are the numbers of untrustworthy and trustworthy data properly detected by our trust model, whereas FP and FN are respectively the numbers of untrustworthy and trustworthy data incorrectly detected by the proposed trust model.

Here, we compare the accuracy of our trust model with [25], [26], [27], and Weighted Voting (WV) method as baseline method that has been extensively used in many previous trust management schemes for wireless networks.

To this end, the simulated malicious attack is a data forgery attack that in this work we vary the percentage of malicious nodes from 10% to 50% with a 5% increment. We also consider 5% of total sensor nodes and smart devices in the network as faulty nodes, where the number of nodes is fixed at 500. The faulty nodes send the wrong data to the gateway. The obtained results indicate that our trust model has better performance than [25], [26], [27] and WV and as a result it is more accurate. Figure 8 shows the overall accuracy of our trust model is 98.7% when 10% of sensor nodes/smart devices in the network behave improperly. In the worst case, when 50% of nodes in the network are malicious and faulty, our trust model can achieve an accuracy of approximately 92%.

9 Conclusion

In this work, we have proposed a security scheme that is a combination of a trust model and authentication scheme called ”TRUTH”. The proposed trust model is based on Dampster-Shafer theory in order to deal with untrustworthy data collected/measured by IIoT smart devices/sensor nodes. The proposed trust model ensures that only trusted data will be stored and transferred to the users. To maintain the integrity of data-in-transit, we have also developed a three-party security scheme based on AKE protocol. The non-mathematical analysis have proved that our scheme meets mutual authentication, data confidentiality, and user's anonymity. The formal verification by using AVISPA has demonstrated that the proposed scheme is efficient against the DY threat model. Furthermore, we have compared our scheme with some related works in terms of security aspects and performance. The comparison of the obtained results also illustrates that our scheme is more secure as compared to existing schemes. In the future, we plan to improve our scheme by adding a module to protect data-at-rest.

References

[1] J. Jiang, G. Han, F. Wang, L. Shu, and M. Guizani, “An efficient distributed trust model for wireless sensor networks,” *IEEE transactions on parallel and distributed systems*, vol. 26, no. 5, pp. 1228–1237, 2014.

[2] C. Boudagdigue, A. Benslimane, A. Kobbane, and J. Liu, “Trust management in industrial internet of things,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3667–3682, 2020.

[3] S. Garg, A. Singh, K. Kaur, G. S. Aujla, S. Batra, N. Kumar, and M. S. Obaidat, “Edge computing-based security framework for big data analytics in vanets,” *IEEE Network*, vol. 33, no. 2, pp. 72–81, 2019.

[4] I. García-Magariño, M. M. Nasralla, and J. Lloret, “A repository of method fragments for agent-oriented development of learning-based edge computing systems,” *IEEE Network*, vol. 35, no. 1, pp. 156–162, 2021.

[5] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, “Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1382–1392, 2017.

[6] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, “An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 76, pp. 37–48, 2016.

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

[8] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E.-J. Yoon, and K.-Y. Yoo, “Secure signature-based authenticated key establishment scheme for future iot applications,” *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[9] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, “Design of secure user authenticated key management protocol for generic iot networks,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 269–282, 2017.

[10] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, “Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2884–2895, 2017.

[11] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, “Secure remote user authenticated key establishment protocol for smart home environment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 391–406, 2017.

[12] P. Gope, J. Lee, and T. Q. Quek, “Lightweight and practical anonymous authentication protocol for rfid systems using physically unclonable functions,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.

[13] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, “A robust ECC-based provable secure authentication protocol with privacy preserving for industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2017.

[14] M. D. Alshehri, F. K. Hussain, and O. K. Hussain, “Clustering-driven intelligent trust management methodology for the internet of things (citm-iot),” *Mobile networks and applications*, vol. 23, no. 3, pp. 419–431, 2018.

[15] J. Jiang, G. Han, L. Shu, S. Chan, and K. Wang, “A trust model based on cloud theory in underwater acoustic sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 342–350, 2015.

[16] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[17] S. Suhail, R. Hussain, R. Jurdak, and C. S. Hong, “Trustworthy digital twins in the industrial internet of things with blockchain,” *IEEE Internet Computing*, 2021.

[18] D. He, C. Chen, S. Chan, J. Bu, and A. V. Vasilakos, “Retrust: Attack-resistant and lightweight trust management for medical sensor networks,” *IEEE transactions on information technology in biomedicine*, vol. 16, no. 4, pp. 623–632, 2012.

[19] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[20] S. A. Soleymani, S. Goudarzi, M. H. Anisi, Z. Movahedi, A. Jindal, and N. Kama, “PACMAN: Privacy-preserving authentication scheme for managing cybertwin-based 6G networking,” *IEEE Transactions on Industrial Informatics*, 2021.

[21] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, “Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4900–4913, 2018.

[22] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, “Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial internet of things,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1133–1146, 2018.

[23] G. S. Poh, P. Gope, and J. Ning, “Privhome: Privacy-preserving authenticated communication in smart home environment,” *IEEE Transactions on Dependable and Secure Computing*, 2019.

[24] A. Varga and R. Hornig, “An overview of the omnet++ simulation environment,” in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, 2008, pp. 1–10.

[25] W. Li and H. Song, “ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks,” *IEEE transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 960–969, 2015.

[26] V. B. Reddy, S. Venkataraman, and A. Negi, “Communication and data trust for wireless sensor networks using d–s theory,” *IEEE Sensors Journal*, vol. 17, no. 12, pp. 3921–3929, 2017.

[27] B. Pang, Z. Teng, H. Sun, C. Du, M. Li, and W. Zhu, “A malicious node detection strategy based on fuzzy trust model and the abc algorithm in wireless sensor network,” *IEEE Wireless Communications Letters*, 2021.