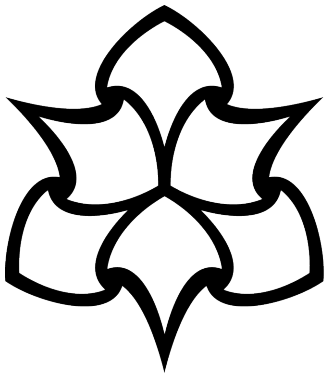

Journal Pre-proof

EBAKE-SE: A novel ECC-based authenticated key exchange between industrial IoT devices using secure element

Chintan Patel, Ali Kashif Bashir, Ahmad Ali AlZubi, Rutvij H. Jhaveri

PII: S2352-8648(22)00242-5

DOI: https://doi.org/10.1016/j.dcan.2022.11.001

Reference: DCAN 556

To appear in: *Digital Communications and Networks*

Received Date: 15 September 2021

Revised Date: 9 September 2022

Accepted Date: 1 November 2022

Please cite this article as: C. Patel, A.K. Bashir, A.A. AlZubi, R.H. Jhaveri, EBAKE-SE: A novel ECC-based authenticated key exchange between industrial IoT devices using secure element, *Digital Communications and Networks* (2022), doi: https://doi.org/10.1016/j.dcan.2022.11.001.

Digital Communications and Networks(DCN)

HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/dcan

EBAKE-SE: A novel ECC-based authenticated key exchange between industrial IoT devices using secure element

Chintan Patel[a], Ali Kashif Bashir[b], Ahmad Ali AlZubi[c], Rutvij H Jhaveri[*d]

[a]**School of Artificial Intelligence, Information Technology and Cyber Security, Rashtriya Raksha University, Gujarat, India**
[b]**Department of Computing and Mathematics, Manchester Metropolitan University, United Kingdom**
[c]**Computer Science Department, Community College, King Saud University, Saudi Arabia**
[d]**Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gujarat, India**

Abstract

Industrial IoT (IIoT) aims to enhance services provided by various industries, such as manufacturing and product processing. IIoT suffers from various challenges, and security is one of the key challenge among those challenges. Authentication and access control are two notable challenges for any Industrial IoT (IIoT) based industrial deployment. Any IoT based Industry 4.0 enterprise designs networks between hundreds of tiny devices such as sensors, actuators, fog devices and gateways. Thus, articulating a secure authentication protocol between sensing devices or a sensing device and user devices is an essential step in IoT security. In this paper, first, we present cryptanalysis for the certificate-based scheme proposed for a similar environment by Das et al. and prove that their scheme is vulnerable to various traditional attacks such as device anonymity, MITM, and DoS. We then put forward an inter-device authentication scheme using an ECC (Elliptic Curve Cryptography) that is highly secure and lightweight compared to other existing schemes for a similar environment. Furthermore, we set forth a formal security analysis using the random oracle-based ROR model and informal security analysis over the Doleve-Yao channel. In this paper, we present comparison of the proposed scheme with existing schemes based on communication cost, computation cost and security index to prove that the proposed EBAKE-SE is highly efficient, reliable, and trustworthy compared to other existing schemes for an inter-device authentication. At long last, we present an implementation for the proposed EBAKE-SE using MQTT protocol.

KEYWORDS: Internet of Things, Authentication, Elliptic Curve Cryptography, Secure Key Exchange, Message Queuing Telemetry Transport

1. Introduction

The industrial Internet of Things (IIoT) network is built up using a highly homogeneous, globally dynamic, deeply deployed, and comparatively resource-constrained devices to provide *"Any type"* service at *"Any location"* to *"Anyone"* on *"Any time"* [1] [2]. The Scale of IIoT data generation is directly proportional to the growing quantity of internet-connected devices. As per recent predictions (June 2019) by the global giant of telecommunications and market intelligence agency International Data Cooperation (IDC), there will be approx 42 billion deployed devices that will generate approx 80 ZettaByte data by 2025 [3] [4].

An IIoT-based devices are a mixture of resource-constrained devices as well as resource-capable devices. Most of the devices deployed on the ground such as smart home, smart industrial factory and smart transportation road are resource-constrained devices, such as sensors and actuators. Devices that collect data from these sensing devices (A.c.a gateway de-

[1]Dr. Chintan Patel,(email:chintan.p592@gmail.com).
[2]Dr. Ali Kashif Bashir, (email:dr.alikashif.b@ieee.org).
[3]Dr. Ahmad Ali AlZubi, (email:aalzubi@ksu.edu.sa).
[4]Dr. Rutvij H. Jhaveri (Corresponding author), (email:rutvij.jhaveri@gmail.com:).

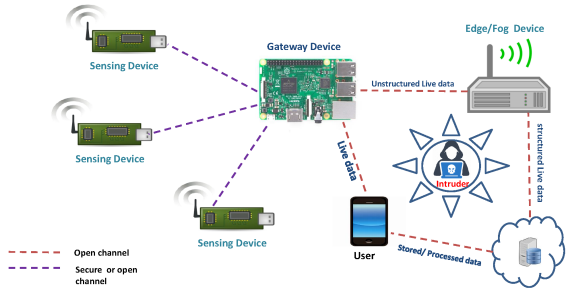

Figure 1: Inter device data transfer in IoT.

vices) are hybrid devices, such as routers, raspberry-pi and node-MCU. The edge device or the fog device receives unstructured data from the sensing devices and performs pre-processing on that data to convert it into structured data. These edge devices are resource-capable and forward only necessary structured data over the cloud or to the user [5]. Edge devices reduce unnecessary traffic over the cloud server through their intelligent pre-processing.

Fig. 1 presents an overview for the generic IIoT *"data chain"*. It highlights how raw material (i.e., unstructured data) is converted into the smart product (i.e., knowledge) used for quick and accurate decision-making. The IoT ecosystem includes three significant aspects. (1) IoT devices (2) reliable, optimized, and secure communication between devices (3) data processing and knowledge generation. A recent survey by Sobin [6] highlights that scalability, lack of standard architectures and protocols, energy efficiency, and security and privacy are still open issues that limit the wide-range deployment of an IoT ecosystem. Other past surveys [7–11] also highlighted that the IoT ecosystem suffers from the numerous privacy and security issues due to its resource-constrained devices, heterogeneous deployment, and dynamic nature.

In the recent past, authors in [7, 12–15] presented a brief study on numerous challenges and issues related to IoT security and privacy. The author highlights an *"authentication"* as a common threat to the IoT ecosystem. A secure and reliable authentication defines as a mutual trust-building between user-device and device-device through a resource-efficient key exchange protocol [16]. In this paper, we provide cryptanalysis for the scheme proposed by Das et al. [17] for device-to-device authentication in a similar environment. We highlight that the scheme proposed by Das et al. is vulnerable to numerous attacks such as device impersonation, Man-In-The-Middle (MITM) attack, and Denial of Service (DoS) attack. We then put forward a considerably reliable and efficient inter-device Remote User Authentication (RUA) scheme using a Secure Element (SE) and an Elliptic Curve Cryptography (ECC). Recently Qureshi et al. presented stream-based authentication for big data networks based on IoT sensing devices [18]

Contribution: In this paper,

- We present cryptanalysis for the authentication scheme proposed by Das et al. [17] for device-to-device authentication. We prove their scheme is not secured against device impersonation, MITM, and a DoS attack.
- We present a novel authentication scheme between two smart IIoT devices via Trusted Authority (TA) using ECC and SE.
- We present an informal security analysis for the proposed scheme using *send* and *receive* based Dolev-Yao channel. We then offer a formal security analysis for the proposed EBAKE-SE using a random oracle-based challenge-response game.
- Next, we demonstrate the implementation scenario and real-time results for the proposed EBAKE-SE using the physical IIoT devices.
- Furthermore, We put forward a comparative analysis of the proposed work with an existing work based on time and space requirements.

Case study and motivations: IoT is a complex matrix of the numerous resource-constrained devices, as well as countless resourceful Advanced IoT (AdIoT) devices [19]. The internet-connected *smart home* appliances, such as washing machines, refrigerators, ACs, and CCTV systems, are considered as AdIoT devices. Wearable devices, such as smartwatches and smart belts (for health monitoring), are regarded as lightweight, resource-constrained devices. Recent surveys show that 98% of IoT devices communicate through open channels, which is the biggest threat to personal privacy and data confidentiality. The *smart healthcare* system is equipped with numerous remote control devices, such as intelligent ventilators, smart oxygen suppliers, and smart patient monitoring systems. The prosperous attack on these devices can cause complete chaos in the healthcare system. Thus, it is highly desirable to protect these IoT devices from traditional vulnerabilities and attacks is highly desirable. Any IoT system must ensure data confidentiality, data integrity, user privacy, secure device authentication, and secure device access control. Protecting the IoT devices from attacks, such as *DoS, MITM, spoofing, and impersonations* is challenging task for security professionals. It is profoundly anticipated that the IoT system users must not use traditional passwords and update them frequently. They must upgrade their system periodically and configure the latest security patches for their devices to protect them from the ransomware attacks such as *WannaCry* and *NotPetya*.

Road map of the paper: Section 2 briefly summarizes the recent work of the proposed EBAKE-SE and the basic preparatory work used to elaborate this

manuscript. In Section 3, we outline the scheme proposed by Das et al., followed by a cryptanalysis of Das et al.'s scheme in Section 4. In Section 5, we proposed a reliable and efficient device-device authentication scheme between two smart IoT devices using a TA. Section 6 and Section 7 conduct formal and informal formal security analysis for the proposed EBAKE-SE, respectively. Section 8 discusses implementation for the proposed EBAKE-SE. In Section 9, we present a comparison of the proposed scheme with other existing schemes based on communication and computation costs. Finally, we conclude this paper in Section 10.

2. Related work and preliminaries

In this section, we will discuss the work related to the proposed work and clarify the main preparatory work required for this paper.

2.1. Related work

Authentication creates trust among communication devices [20]. An ECC is an efficient and reliable advancement for lightweight cryptography. The ECC provides the same strong security compare with the RSA and other traditional methods in much lighter ways (smaller key size and addition-based discrete logarithm). An ECC plays a key role in the optimized deployments of lightweight cryptography. The ECC is a kind of public-key cryptography that works on the basic assumption that it is impossible to find the discrete logarithm of random elliptic curve elements based on a known base point. Miller introduced the use of ECC in 1985 [21] and populated by koblitz in 1987 [22]. Between 1987 and 2021, numerous authors proposed the ECC-based key exchange and authentication schemes.

In 2019, Dhillon et al. [23] proposed an ECC-based authentication scheme for the *SIP (Session Initiation Protocol)* that is used in *VoIP (Voice-over-IP))* communication and provided a security analysis using AVISPA tool. Wearable devices play a key role in the numerous IoT-based applications such as smart healthcare and smart home. In 2019, Kumar et al. [24] proposed the key exchange protocol between a user device (mobile device) and a wearable device using an ECC. In 2019, Lohachab et al. [25] presented a scheme using an ECC for the MQTT communication and provided a security analysis using the AVISPA and an *ACPT (Access Control Policy Testing)* tool. In 2019, Qi et al. [26] proposed an ECC-based authentication scheme for the secure session key establishment between a system user, *Low Earth Orbit (LEO) satellite*, and the gateway device.

In 2019, Garg et al. [27] also proposed an authentication scheme for the IIoT environment using lightweight operations, such as ECC and *Physically Unclonable Functions(PUF)*. In 2019, Dammak et al. [28] proposed the token-based authentication scheme for the user-gateway-device communication and claimed that their scheme is secured against a token impersonation attack and a stolen verifier attack. Recently, Dang et al. [29] proposed an authentication scheme using an ECC for the smart city environment. Authors in [29] used the Device-Device-Server *(DDS)* network model to articulate their scheme and claimed that the proposed work achieved high energy efficiency.

Designing a fully secured and highly resource-efficient security mechanism for an IoT environment is challenging. The IoT environment suffers from numerous vulnerabilities, such as inadequate physical security of the sensing devices, heterogeneity of the device manufacturers, proper standardizations, lower device synchronizations, and open ground for attackers. Hence, this paper proposes a novel authentication scheme that provides a robust and secured environment for session key generation between two IoT devices.

2.2. Preliminaries

2.2.1. Elliptic Curve Cryptography (ECC)

An ECC provides a lightweight implementation for the public-key cryptography protocols such as an RSA with an equal level of security. We can define an elliptic curve as a cubical curve of the form $E_z(\alpha,\beta)$ with the non-repeatable roots defined over a finite field $\mathcal{F}_z$ where z is a large prime number. We can represent an elliptic curve according to Eq. 1 below.

$$E_z(\alpha,\beta) : Q^2 = (P^3 + \alpha * P + \beta) mod \gamma \qquad (1)$$

Here, P and Q are two curve points denoted by $P_t(P,Q)$. The γ represents a large prime number. Two constants $\{\alpha, \beta\}$ are selected such that $\{\alpha,\beta\} \in \mathcal{F}_z$ and their values must satisfy

$$4 * \alpha^3 + 27 * \beta^2 \neq 0 mod \gamma \qquad (2)$$

We can define the scalar point multiplication operations of an ECC over a point P_t as follows $n * P_t = P_t + P_t + ... + P_t$ for n times. The security of an ECC lies in finding the value of a large prime n from the given P_t and $n * P_t$. We can define the Elliptic Curve Discrete Logarithm Problem (ECDLP) as follows: from the given $R = n*T$, it is difficult to find an integer n in polynomial time where $n \in \mathcal{F}_z$ and R and T are two points on elliptic curve $E_z(\alpha,\beta)$. We can define the Elliptic Curve Diffie-Hellman Problem (ECDHP) as follows: consider $\{\alpha,\beta\} \in \mathcal{F}_z$ and P is a point on the curve $E_z(\alpha,\beta)$. From the given P, $\alpha * P$ and $\beta * P$, it is difficult to compute a value $\alpha * \beta * P$ over $E_z(\alpha,\beta)$ in a polynomial time.

2.2.2. One-way hash function

A cryptographic hash function can be presented as $h : \{0,1\}^* \rightarrow \{0,1\}^n$ that takes string $p \in \{0,1\}^*$ as an

input and outputs a fixed-size binary string $Q \in \{0,1\}^n$. The cryptographic hash function must be collision resistant and preimage resistant for variable-size input and fixed-size output with enough randomness.

2.2.3. Networkmodel

A network model shown in Fig. 2 [17] is followed for designing of authentication scheme. We consider the cloud Trusted Authority(TA) as a master controller in this network model. The IoT devices transmit data to each other over an open channel via the TA. The TA is a cloud MQTT server equipped with a broker. The IoT devices (such as a smart fridge or a gateway device) have a secure element that stores secret credentials in the tamper-resistant environment and the Wi-Fi module (to connect with the internet). The secure element of a first device performs cryptographic operations in a tamper-proof environment and passes its outcome to the Wi-Fi module. This module publishes that data to the TA using the MQTT protocol, and the TA performs authentication operations and communicates with the second device using an MQTT. In this way, each of the three entities mutually authenticates each other, and after completion of the authentication phase, the IoT devices generate a one-time secure session key. Many authors follow another network model [30] in that the gateway device is considered a trusted device due to the absence of a separate TA. Still, for the proposed scheme, we consider the presence of a separate TA (also as a gateway) that setups security parameters for the IoT devices, including a gateway device, if required.

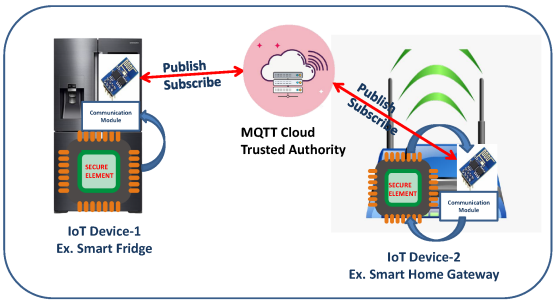

Figure 2: Network model.

2.2.4. Threat model

We adopted the Dolev-Yao channel-based threat model for the proposed scheme. The attacker model or threat model for the proposed scheme is as follows:

- Challenger C can read, access, modify, and store the communication over the open channel.
- Smart IoT devices, including gateway devices (in the presence of separate certificate authority or trusted authority), are not trusted devices.
- Challenger C can capture the smart IoT device and extract the stored data over it.
- The TA is a trusted entity, and the polynomial-time challenger C can not compromise it.
- Challenger C might receive the secrets of a TA in case of system failure.

2.2.5. Notations and symbols

Table 1 gives symbols and notations used for cryptanalysis and designing of the EBAKE-SE.

Table 1: Notations and symbols

Symbols	Descriptions
TA	Trusted Authority
D_x, D_y	Xth and Yth Smart IoT devices
ID_x, ID_y	Identity of xth and yth smart IoT devices
TS_x / T	Time-stamp
Pr_x	Private key of device D_x
Pub_x	Public key of device D_x
SK_{xy}	Generated session key
Topic	MQTT topic
r_d	Random number
K_{dta}	160 bit shared key
N_d	Random nonce
$E_p(a,b)$	Elliptic curve selected by TA
Enc/Dec	Encryption/Decryption
$\oplus$	Exclusive OR operation
P	Basepoint of the elliptic curve
D_{GWN}	Gateway device

3. Review of Das et al.'s scheme

The scheme proposed by Das et al. [17] consists of four phases: (1) *System setup phase* by the TA; (2) *Device registration phase* by smart IoT device with the TA; (3) *Device authentication phase* between two IoT smart device; (4) *Dynamic device addition phase* by TA.

(1) System setup phase:

In this phase, the TA decides finite field $\mathcal{F}_z$ and selects elliptic curve $E_z(a,b)$ (i.e, FIPS 186) over it. The TA also chooses basepoint P of order x such that $x * P = O$ (infinity point). The TA generates a pair of its own private key and public key as a (Pr_{TA}, Pub_{TA}) where Pr_{TA} is a randomly generated number and $Pub_{TA} = Pr_{TA} * P$. Furthermore, the TA chooses the one-way hash function h(.) (i.e, SHA1, MD5) for further processing and consistency between all devices. Finally, the TA publishes $E_p(a,b)$, P, p, Pub_{TA}, h(.) as a public parameters and stores Pr_{TA} as a private parameter. Note that the TA is considered a trusted entity [19].

(2) Device registration phase:

In this phase, the TA generates the pair of $\{ID_x, Pr_x, A_x, c_x, Pub_x, E_p(a,b), P, p, Pub_{TA}$, h(.)$\}$ and then loads it into the memory of the device D_x. Here $Pub_x = Pr_x * P$, $A_x = (Pr_x + l_x) * P$, where l_x is a distinct random number for each device D_x and $c_x = Pr_{TA} +$

$(Pr_x + l_x)$h$(ID_x||A_x)$. The pair of $\{ID_x, Pr_x\}$ is generated by the TA for each device D_x.

(3) Device authentication phase:

In this phase, two smart IoT devices D_x and D_y authenticate with each other and set the session key SK_{xy}. This phase is summarized as follows:

1. $D_x \rightarrow D_y$ **:** The D_X produces random r_x and timestamp TS_x, computes $R_x = r_x * P$, $z_x = c_x + h(A_x||c_x||R_x||Pub_x||TS_x)(r_x + Pr_x)$. The D_x sends *message 1* $= \{TS_x, ID_x, c_x, z_x, A_x, Pub_x, R_x\}$ to another IoT device D_y.
2. $D_y \rightarrow D_x$ **:** The D_y verifies timestamp and $U_y \stackrel{?}{=} c_x * P$ after computing $U_y = Pub_{TA} + h(ID_x||A_x)A_x$, and also verifies $W_y \stackrel{?}{=} z_x * P$ after computing $W_y = c_x * P + h(A_x||c_x||R_x||TS_x||Pub_x)(R_x + Pub_x)$. Next to these verification, the D_y produces TS_y and r_y and computes $R_y = r_y * P$, $z_y = c_y + h(A_y||c_y||R_y||Pub_y||TS_y)(r_y + Pr_y)$, $K_{xy} = pr_y * Pub_x$, $B_{xy} = r_y * R_x$, $SK_{xy} = h(B_{xy}||K_{xy}||TS_y||TS_x||ID_x||ID_y)$, $SKV_{xy} = h(SK_{xy}||TS_y)$, and sends *message 2* $= \{ID_y, TS_y, A_y, c_y, z_y, SKV_{xy}, Pub_y, R_y\}$ to device D_x.
3. $D_x \rightarrow D_y$ **:** The device D_x verifies timestamp and $U_x \stackrel{?}{=} c_y * P$ by computing $U_x = Pub_{TA} + h(ID_y||A_y)A_y$. The device D_x verifies $W_x \stackrel{?}{=} z_y * P$ by computing $W_x = c_y * P + h(A_y||c_y||R_y||TS_y||Pub_y)(R_y + Pub_y)$, computes $K'_{yx} = pr_x * Pub_y$, $B'_{yx} = r_x * R_y$, $SK'_{xy} = h(B_{yx}||K_{yx}||TS_x||TS_y||ID_y||ID_x)$, and verifies $SKV_{xy} \stackrel{?}{=} h(SK'_{xy}||TS_y)$. After this verification, the device D_x produces timestamp TS'_x, computes $SKV'_{yx} = h(SK'_{yx}||TS'_x)$, generates *message 3* $= \{SKV'_{yx}, TS'_x\}$ and sends it to the device D_y.
4. $D_y \rightarrow D_x$ **:** The device D_y verifies timestamp and $SKV_{yx}* \stackrel{?}{=} SKV'_{yx}$ after computing $SKV_{yx}* = h(SK'_{yx}||TS'_x)$. After this verification, both devices D_x and D_y agree on the session key $SK'_{yx} = SK_{xy}$.

(4) Dynamic device addition phase:

In this phase, the TA deploys a new device or replaces device D_x by D'_x. The TA selects ID'_x and private key Pr'_x, computes public key $Pub'_x = Pr'_x * P$, and generates random number l'_x. The TA calculates $A'_x = (Pr'_x + l'_x) * P$, $c'_x = Pr_{TA} + (Pr'_x + l'_x)h(ID'_x||A'_x)$ and stores $\{ID'_x, Pr'_x, A'_x, c'_x, Pub'_x, E_p(a,b), P, p, Pub_{TA}$, h(.)$\}$ into the memory of the device D'_x.

4. Cryptanalysis of Das et al.'s scheme

In this section, we provide cryptanalysis for Das et al.'s and show that their scheme is vulnerable to attacks, such as device impersonation, MITM, and DoS attacks.

4.1. Vulnerable against identity theft attack/ device tracking attack

In the device authentication phase between device D_x and D_y,

- Device D_x sends *message 1* $= \{TS_x, ID_x, A_x, c_x, z_x, Pub_x, R_x\}$ to D_y over an open channel.
- The *message 1* contains identity ID_x of the device D_x in the plain text. The device D_x does not protect its identity inside *message 1* through either hash or encryption. Thus, any challenger C can capture the ID_x and use it for tracing the device D_x.
- Device D_y sends *message 2* $= \{ID_y, TS_y, A_y, c_y, z_y, SKV_{xy}, Pub_y, R_y\}$ to D_x over an open channel.
- The *message 2* contains identity ID_y of the device D_y in the plain text. The device D_y does not protect its identity inside *message 2* though either hash or encryption. Thus, any challenger C can capture the ID_y and use it for tracing the device D_y.

4.2. Vulnerable against device impersonation attack / device capturing attack/DoS

Protecting the device from a *physical device capturing* is a significant challenge in the IoT deployment. Authors in [17] do not provide any challenger limitations about the physical capturing of the smart devices. In the attacker model, Das et al. highlighted that the IoT device could be captured by the challenger C. Challenger C can apply the *power analysis attack* [31] on any IoT device and can extract the stored information. Now let us examine Das et al.'s scheme against *device impersonation attacks*.

- In the device registration phase, the TA loads $\{ID_x, Pr_x, A_x, c_x, Pub_x, E_p(a,b), P, p, Pub_{TA}$, h(.)$\}$ on device D_x. Now let us assume that the challenger C physically captures device D_x and applies the power analysis attack on it. After performing successful power analysis attacks, the challenger C already has $\{ID_x, Pr_x, A_x, c_x, Pub_x, E_p(a,b), P, p, Pub_{TA}$, h(.)$\}$.
- Now, let us examine the first message generated by the device D_x. The device D_x sends *message 1* $= \{TS_x, ID_x, A_x, c_x, z_x, Pub_x, R_x\}$ over an open channel. Now, the challenger C tries to generate a valid *message 1**.
- The challenger C already has $\{ID_x, A_x, c_x, z_x, Pub_x\}$. Now the challenger C generates random number r_c from the public parameters of ECC and computes $R_c = r_c$*P. Now, the challenger C also generates timestamp TS_c and sends message *message 1** $= \{TS_c, ID_x, A_x, c_x, z_x, Pub_x, R_c\}$ to device D_y.

- Now the device D_y verifies timestamps, and computes $U_y = Pub_{TA} + h(ID_x||A_x)A_x$, and device D_y successfully verifies $W_y \stackrel{?}{=} z_x * P$ after computing $W_y = c_x * P + h(A_x||c_x||R_c||TS_c||Pub_x)(R_c + Pub_x)$. Thus, the challenger can also generate *message 1** causes a valid *device impersonation*.

- In the scheme proposed by Das et al., the device D_x or the device D_y does not block fake devices even if the sender fails multiple times. Thus, this can easily drain the receiving device's battery and may lead to power failure. Therefore, we can say that any malicious attacker can send fake requests and lead the system to *DoS*.

4.3. Vulnerable against MITM attack / fake session key setup

The scheme of Das et al. is also vulnerable to MITM attacks. In the scheme proposed by Das et al.,

- Let us assume that there is a malicious intruder C eavesdrops public message *message 1* = $\{TS_x, ID_x, A_x, c_x, z_x, Pub_x, R_x\}$ and *message 2* = $\{ID_y, TS_y, A_y, c_y, z_y, SKV_{xy}, Pub_y, R_y\}$. Now let us assume that C computes $B_{cj} = r_c{*}R_x$ and $K_{cj} = x_c{*}Q_x$ generate $SK^{**}_{xc} = h(B_{cj}||K_{cj}||TS_y||TS_x||ID_x||ID_y)$, $SKV^{**}_{xc} = h(SK_{xc}||TS_y)$ and forwards to device D_x.

- We must note here that challenger C only replaces SKV_{xy} by $SKV_{xc}**$ and sends the remaining *message 2* as it is. Thus, device D_x can not identify that the received message is from challenger C, not from the valid device D_y. Device D_x uses Pub_y and R_y from *message 2* (not from the previous knowledge) for the computation of the B'_{ij} and K'_{ij}. Thus, unknowingly, device D_x establishes the session key with challenger C.

5. Proposed scheme : EBAKE-SE

The *Secure Element (SE)* is a tamper-resistant microprocessor chip that stores secret data for the tiny devices and securely runs their applications [32]. The *secure element* is embedded with the IoT devices so that the logical tempering of it becomes an impossible task and the physical tempering of a *secure element* destroys the functioning of the device. In the proposed setup, we consider that both the IoT devices are embedded with the *secure element* on it. Fig. 2 shows the communication model for the proposed EBAKE-SE. In EBAKE-SE, we consider the MQTT Cloud server as a resource-capable, trusted authority that runs the MQTT broker module. We highlight more details about the MQTT protocol in Section 8. In this section, we provide the improvements of the scheme proposed by Das et al. [17]. In the proposed EBAKE-SE, there are two major phases. In the first phase, the TA initializes the system, generates necessary parameters, and stores those parameters in the SE of the smart IoT devices. In the second phase, two IoT devices perform mutual authentication via TA and generate a one-time session key (SK_{xy}) for further secure communications. In this phase, the TA also allocates a temporary (for a session) MQTT topic on which these devices perform encrypted communication. In the proposed EBAKE-SE, each smart IoT device has two connected modules. The first module is the SE module, which runs cryptographic operations. The second module is a wifi module (we used the esp8266 module for implementation), which connects the device with the Internet for communication with TA using the MQTT protocol. The proposed EBAKE-SE overcomes the limitations of the analyzed scheme and introduces some novel features compare with other existing schemes proposed for the similar environment.

5.1. System initialization phase

In this phase, the TA generates credentials for self and smart IoT devices and loads those credentials over the SE of the IoT device. The TA performs initialization phase in a secure environment as follows: the TA selects a basepoint P for the curve $E_p(a, b)$. The TA generates a unique identity for each x^{th} device as ID^x_d, generates random number for each x^{th} device as a r^x_d, and generates shared secret K_{dta} between the device D_x, other IoT devices and itself. TA updates K_{dta} periodically. The TA computes device parameter DP^x_1 : *hash* $\langle ID^x_d, r^x_d, K_{dta} \rangle$. The TA computes public parameter Q^x_d : r^x_d*P for each x^{th} smart IoT device. The TA loads pair $\langle ID^x_d, r^x_d, K_{dta}, DP^x_1 \rangle$ on SE of device D_x. The TA also loads pair $\langle ID^x_d, DP^x_1, K_{dta}, Q^x_d \rangle$ into its own secret memory.

5.2. Mutual authentication phase

In the IoT setup, each party must have trust in the other. In this phase, initially, we perform the mutual authentication between devices $\langle D_x, TA \rangle$, $\langle D_y, TA \rangle$, and $\langle D_x, D_y \rangle$. This is followed by a secure session key generation between devices D_x and D_y as a SK_{xy} and topic allocation by TA. The system performs mutual authentication as follows:

The device D_x generates a temporary id ID^x_T: $\langle W^x, Y^x, Z^x \rangle$ as follows:

Step-1 : The device D_x generates random nonce N^x_d and computes W^x: *Enc* $\langle (K_{dta}, (ID^x_d, r^x_d)) \rangle$, Y^x: *xor* $\langle (DP^x_1, Q^y_d) \rangle$, Z^x: *Enc* $\langle (Q^y_d, (Q^x_d, ID_x N^x_d, T_1)) \rangle$, P^x_d: *hash* $\langle DP^x_1, N^x_d, T_1 \rangle$. Device D_x publishes $\langle ID^x_T, P^x_d, T_1 \rangle$to TA.

Step-2 : The TA receives ID^x_T and performs as follows: the TA first verifies the timestamp and then verifies identity of the sending device as follows: the TA verifies $\Delta T \stackrel{?}{\leq} T^*_1 - T_1$, retrieves pair $\langle (ID^x_d*, r^x_d*) \rangle$ by *Dec* $\langle K_{dta}, (W^x) \rangle$. The TA computes DP^x_1*: *hash* $\langle ID^x_d*, r^x_d*, K_{dta}* \rangle$, computes P^x_d*:*hash* $\langle DP^x_1*, T_1 \rangle$ and verifies $P^x_d* \stackrel{?}{\leq} P^x_d$. After three unsuccessful verifications from the same device, the TA blocks the device

for a day. Now, the TA retrieves Q_d^y* : *xor* $\langle(DP_1^x*, Y^x)$. The TA identifies D_y, computes P_d^y: *hash* $\langle DP_1^y, T_2$, and publishes $\langle Z^x, P_d^y, T_2 \rangle$to D_y.

Step-3 : The D_y receives pair $\langle Z^x, P_d^y, T_2 \rangle$. The D_y verifies $\Delta T \stackrel{?}{\leq} T_2^* - T_2$ and retrieves $\langle Q_d^x, ID_d^x, N_d^x, T_1 \rangle$by $Dec \langle(r_d^y, (Z^x)\rangle$. The device D_y verifies $P_d^y \stackrel{?}{\leq} P_d^y*$: *hash* $\langle DP_1^y, T_2$. By this verification, the device D_y authenticates the TA. After three unsuccessful authentications, the device D_y blocks TA for a day by considering it as a DoS attack from the malicious insider. Now the device D_y generates a nonce N_d^y, computes Z^y: $Enc \langle(Q_d^y, (ID_y, N_d^y, T_2)\rangle$, computes P_d^{TA}: *hash* $\langle DP_1^x, ID_d^x, ID_d^y, T_3, ID_d^y$ and publishes pair $\langle Z^y, P_d^{TA}, T_3 \rangle$to TA. The device D_y computes one-time secure session key for the device D_x as SK_{xy} : *hash* $\langle ID_y, N_d^y, T_1, ID_x, N_d^x, T_2, K_{dta}) \rangle$.

Step-4 : The TA receives data from the device D_y and verifies $\Delta T \stackrel{?}{\leq} T_3^* - T_3$. Now the TA also verifies $P_d^TA \stackrel{?}{\leq} P_d^TA$: *hash* $\langle DP_1^x, ID_d^x, ID_d^y, T_3, ID_d^y$. After three unsuccessful verifications from the same device, the TA blocks the device for a day. Now the TA computes P_d^{xx}: *hash* $\langle DP_1^x, Z^y, T_4 \rangle$, and publishes pair $\langle Z^y, T_4 \rangle$along with MQTT topic T to device D_x. The TA shares the same MQTT topic (T) with the device D_y.

Step-5 : The device D_x verifies $\Delta T \stackrel{?}{\leq} T_4^* - T_4$ and $P_d^{xx} \stackrel{?}{\leq} P_d^{xx}*$: *hash* $\langle DP_1^x, Z^y, T_4 \rangle$. By verifying the device, D_x authenticates both the TA and the device D_y. After three unsuccessful authentications, the device D_x blocks the communication with the TA for a day by considering it a DDoS attack from the malicious insider. The device D_x retrieves pair $\langle(Q_d^y, (ID_y, N_d^y, T_2) \rangle$by $Dec \langle(r_d^x, (Z^y) \rangle$, and computes one-time secure session key for the device D_y as SK_{xy} : *hash* $\langle ID_x, N_d^x, T_1, ID_y, N_d^y, T_2, K_{dta} \rangle$. The device D_x and D_y starts SK_{xy} encrypted communication over a given topic T.

Thus, after completion of this phase, both the devices have a pair of $\langle SK_{xy}, T \rangle$. We like to observe that even though we perform mutual authentication via TA, the TA can not compute the final session key SK_{xy} due to a lack of awareness about the random numbers (r_d^x, r_dy) and the random nonces (N_d^y, N_d^y). The verification parameters (P_d^x, P_d^y, P_d^{XX}, P_d^{TA}) provide strength to the proposed work. The use of timestamps prevents an intruder from performing a replay-type attack. In the proposed EBAKE-SE, to protect a device from the DoS and DDoS type attacks, we block malicious devices for a day if the receiver could not authenticate it after three verification. The novelty in the proposed scheme lies with the use of the tamper-resistant SE on each IoT device.

6. Informal security analysis

In this section, we show that the proposed EBAKE-SE achieves desired security goals and resists all well-known attacks with excellent cryptography functions. Table 2 highlights the comparison between the proposed scheme and other existing schemes based on security features.

6.1. Achieves security against traditional and non-traditional attacks

This subsection provides proof of the "informal security" for the proposed EBAKE-SE.

F1. EBAKE-SE is secure against a reply attack: We involve random numbers and timestamps in all the exchanged messages during the mutual authentication phase of the proposed EBAKE-SE. Use of the random numbers $\{N_d^x, N_d^y\}$, and timestamps $\{T_1, T_2, T_3, T_4\}$ guarantees the freshness of the communicated messages. As a result, the proposed EBAKE-SE is free from replay attacks.

F2. EBAKE-SE is secure against an MITM Attack: Suppose a challenger C expropriate the valid authentication messages and tries to modify these messages to another valid authentication message. It is "computationally infeasible challenge" for challenger C to generate a valid authentication message $\{ID_T^x, P_d^x, T_1\}$ due to the unawareness about the shared secret K_{dta} stored in SE and original random nonce N_d^x. Similarly, C can not also generate other valid authentication messages. This obliques that the proposed EBAKE-SE achieves protection from the Man-In-The-Middle attack.

F3. EBAKE-SE is secure against an impersonation attack: In an impersonation attack, challenger C tries to create a valid authentication *message* $\{ID_T^x, P_d^x, T_1\}$, pretending to be a valid device D_x. The challenger C must require secret parameters, such as $\{K_{dta}, ID_d^x, r_d^x\}$, to generate *message*. These secret parameters are stored in SE, and it is impossible for the challenger C to obtain these values. Thus, eavesdropping of *message* will not allow challenger C to generate a similar *message** to impersonate a device D_x. In a similar way, C can not also pretend to be device D_y. Hence, the proposed EBAKE-SE is immune enough against an impersonation attack.

F4. EBAKE-SE retains anonymity and traceability: Suppose challenger C captures messages $\{ID_T^x, P_d^x, T_1\}$, $\{Z^x, P_d^y, T_2\}$, $\{Z^y, P_d^{TA}, T_3\}$, $\{Z^y, T_4\}$ and tries to trace the devices D_x and D_y. To trace the devices, challenger C must require either static messages or public identity. In the proposed EBAKE-SE, each message is an output of the random values, and none of the public messages contains the identity of either device in the plain text. Therefore the proposed EBAKE-SE achieves anonymity and traceability.

Table 2: Security Features and Goals

Scheme	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓
[17]	✓	✗	✗	✓	✓	✗	✓	✓	✓
[33]	✓	✓	✓	✓	✓	✓	✗	✗	✗
[30]	✓	✗	✓	✓	✓	✗	✓	✓	✓
[34]	✓	✓	✓	✓	✗	✓	✓	✗	✗
[35]	✓	✗	✓	✓	✗	✓	✓	✓	✓
[36]	✗	✗	✓	✓	✓	✓	✗	✓	✗

F5. EBAKE-SE can resist secret leakage attacks: In the proposed scheme, we use long term secrets $\{K_{dta}, r_d^x\}$ and session-specific temporary nonces $\{N_d^x, N_d^y\}$. The session key is computed as a SK_{xy} : $hash\{ID_x, N_d^x, T_1, ID_x, N_d^x, T_2, K_{dta}\}$. Now let us assume that challenger C reveals pair $\{K_{dta}, r_d^x\}$, then he/she can not compute the session key because of non availability of $\{ID_x, N_d^x, ID_x, N_d^x\}\ \rangle$. Similarly, exposure of any information does not allow challenger C to validate a key. Hence, we derive that EBAKE-SE can resist secret leakage attack.

F6. EBAKE-SE can resist insider attacks: Suppose that a malicious administrator on TA tries to compute the session key using available data, the malicious administrator retrieves stored parameters $\{ID_d^x, DP_1^x, K_{dta}, Q_d^x\}$ and receives public messages $\{ID_T^x, P_d^x, T_1\}$, $\{Z^x, P_d^y, T_2\}$, $\{Z^y, P_d^{TA}, T_3\}$, $\{Z^y, T_4\}$. The malicious administrator does not get random nonces $\{N_d^x, N_d^y\}$ necessary for session key computations. In the proposed EBAKE-SE, the TA does not store $\{r_d^x, r_d^y\}$. Hence, the proposed EBAKE-SE is free from malicious insider attacks.

F7. EBAKE-SE implements the session key agreement: In the proposed EBAKE-SE, the mutual authentication between the smart devices and TA is achieved by following verifications: $P_d^x* \stackrel{?}{\leq} P_d^x$ (By TA for D_x), $P_d^y \stackrel{?}{\leq} P_d^y*$ (By D_y for TA), $P_d^T A \stackrel{?}{\leq} P_d^T A$ (By TA for D_y) and $P_d^{xx} \stackrel{?}{\leq} P_d^{xx}*$ (By D_x for TA and D_y). The session key computation involves insider parameters from these validations SK_{xy} : $hash\ \langle ID_y, N_d^y, T_1, ID_x, N_d^x, T_2, K_{dta})\ \rangle$. Therefore, we derive that the proposed EBAKE-SE achieves session key agreement.

F8. EBAKE-SE can resist perfect forward secrecy : Suppose challenger C obtains shared secret credentials K_{dta}, the challenger intercepts the messages $\{ID_T^x, P_d^x, T_1\}$, $\{Z^x, P_d^y, T_2\}$, $\{Z^y, P_d^{TA}, T_3\}$, $\{Z^y, T_4\}$ communicated between the smart devices via TA. To obtain the previous session key, challenger C must compute $SK'_{xy} = hash\ \langle ID'_y, N_d^y, T'_1, ID'_x, N_d^x, T'_2, K'_{dta}\ \rangle$. Even though, if the adversary also obtains an identity of devices somehow, he/she must extract past random nonces $\{N_d^x, N_d^y\}$ protected through encryption. Hence, the proposed EBAKE-SE provides perfect forward secrecy.

7. Formal security analysis using ROR

In this section, we provide a formal security model for the session key (SK_{xy}) derived as an outcome of EBAKE-SE. A random oracle-based Real-Or-Random (ROR) model is used for the formal security modelling of the proposed EBAKE-SE. Recently, many researchers in [3, 17] adopted the ROR model for their security validations. ROR follows the principle of "indistinguishability" between a real session key and a random number. We first instigate the ROR security model and then provide the security proof for the proposed EBBAC-SE under the instigated model.

7.1. Security Model

We define a security model of the proposed EBAKE-SE using a game between a Probabilistic Polynomial Time(PPT) challenger C and a responder $\mathcal{R}$. In this game, challenger C loads oracle queries, and responder $\mathcal{R}$ responds to these queries. Let us consider three participants (smart IoT device D_x, smart IoT device D_y, and trusted authority TA) in the proposed protocol $\mathcal{P}$.

Responder Model: Let us define that oracle instances for responders O_{TA}^l, $O_{D_x}^m$, $O_{D_y}^n$ are oracles of l, m and n for the TA instances, device D_x and the device D_y respectively. These participants are called fresh if they do not reveal the original session key as a response to the $\mathcal{R}$ query by C. These participants are called partners if they share a common session-id S_{id} transcript of all communicated messages. These participants are commonly considered as $\mathcal{D}_\updownarrow$ if it is not necessary to represent them separately.

Challenger Model: We design a challenger C using the famous *Dolev-Yao model*. The challenger can perform active and passive attacks over the Dolev-Yao channel. Following random oracle, queries define capabilities for a PPT challenger C.

Execute Query: $\mathcal{E}(O_{TA}^l, O_{D_x}^m, O_{D_y}^n)$ query provides all communicated messages over open channel between all participants. This query is a passive attack over the proposed protocol $\mathcal{P}$.

Reveal Query: $\mathcal{R}(O_{D_x}^m)$:) query responds session key SK to challenger C if responder $\mathcal{R}$ accepts it.

Hash Query: $\mathcal{H}(m_x)$ query responds random r_x and stores it in a list $\mathcal{L}_\S$ defined with a null value byresponder $\mathcal{R}$.

Send Query: $\mathcal{S}(O_{D_x}^m, m_x)$ query is presented as an active intrusion over proposed protocol $\mathcal{P}$. The challenger C sends message m_x to the responder $\mathcal{R}$ and gets the reply from $\mathcal{R}$ according to the specifications of the message m_x.

Test Query: $\mathcal{T}(O_{D_x}^m)$:) query responds either true session key or an equal size random element. The responder $\mathcal{R}$ randomly selects a bit u. If $\mathcal{R}$ randomly selects $u = 1$, then it returns the original session key else (means $u = 0$) and it also returns a random element with equal bit length of SK to challenger C.

Corrupt Query: $CR(O^m_{D_x})$:) query responds data stored inside the memory of responder $\mathcal{R}$ to challenger C. Through this query, the challenger can get any data storage in the memory of IoT devices

The challenger tries all these queries for finite times, and after executing these queries, C guesses the value of bit u as u'. Let $Adv_{\mathcal{P}}$ represent the winning event (retrieves original session key) for challenger C and $\mathcal{SUC}$ represents the success position for C. We can define challenger C's advantage of breaking the proposed EBAKE-SE as:

$$Adv_{\mathcal{P}}(C) = 2 * Pr[\mathcal{SUC}] - 1 \tag{3}$$

OR

$$Adv_{\mathcal{P}}(C) = 2 * Pr[u' = u] - 1 \tag{4}$$

Let q_s represent the number of *send* queries, l_h represents the hash length, l_r represents the length of random elements, q_h represents the number of the hash query, and q_e represents the number of executing query, and we can give the formal security proof for the proposed EBAKE-SE as follows:

7.2. Formal security proof

Theorem 1. *We consider the cyclic group $\mathcal{G}$ of order n to define an elliptic curve E over finite field F_p. We define the finite time t_c for challenger C tries q_h, q_e and q_s to break the proposed protocol $\mathcal{P}$. We can define security for the proposed $\mathcal{P}$ against oracle queries loaded by challenger C as*

$$Adv_{\mathcal{P}}(C) \leq \frac{q_h^2}{2^{l_h+1}} + \frac{(q_s+q_e)^2}{2^{l_s+1}} + (4 * q_e + 2 * q * s) \; Adv_{C^{ECDH}}(t^*) + max(q_s, (\frac{1}{2^l}, \rho_{fp})) \tag{5}$$

For any given xP and yP, the $Adv_{C^{ECDH}}(t^*)$ represent the polynomial time (t^*) probability for challenger C to break the elliptic curve diffie-hellman problem and compute the valid xyP value.

Proof. We define four identical security games $\{Gm_0, Gm_1, Gm_2, Gm_3\}$, which proves that the proposed protocol $\mathcal{P}$ is secured against P.P.T. and challenger C under ROR model and $Adv_{\mathcal{P}}(C)$ is negligible under random oracle game. Let Suc_i define the probability of correctly guessing the value of bit u by challenger C for the game Gm_i during the challenge session.

Game Gm_0: The game Gm_0 is an identical game to real protocol. If challenger C takes more time than a threshold t^* or does not respond to the game, then the arbitrary value for the bit u will be selected. Thus, it is apparent that

$$Adv_{\mathcal{P}}(C) = 2 * Pr[Suc_0] - 1 \tag{6}$$

Game Gm_1: In this game, challenger C performs executive query $\mathcal{E}$ to eavesdrop communication between devices (D_x and D_y) and the trusted authority (TA).

- $\mathcal{E}(D_x, TA)$: is loaded for capturing the communication between the device D_x and TA.
- $\mathcal{E}(TA, D_y)$: is loaded for receiving the communication between device D_y and the TA.

The challenger C stores all the messages extracted from the above queries and tries to compute the session key SK_{xy}. If the challenger C could compute the session key, then challenger C captures the game Gm_1; otherwise, it is considered that $\Pr[Suc_1] = \Pr[Suc_0]$. In the proposed scheme, we compute the final session key $SK_{xy} : hash \langle ID_x, N^x_d, T_1, ID_y, N^y_d, T_2, K_{dta}) \rangle$ using the random nonces and the nonpublic identities with a shared secret. Hence,

$$Pr[Suc_1] = Pr[Suc_0] \tag{7}$$

In the proposed scheme, we compute the final session key $SK_{xy} : hash \langle ID_x, N^x_d, T_1, ID_y, N^y_d, T_2, K_{dta} \rangle$ using the random nonces and the nonpublic identities with shared secret; hence, it is infeasible for challenger C to compute the session key using captured information that is identical to the game Gm_0. Therefore, the equation 7 holds true.

Game Gm_2: In this game, challenger C performs $\mathcal{H}$ and $\mathcal{S}$ query to communicate with the devices (D_x and D_y) and the TA. In this game, challenger C tries to create a collision for the establishment of a fake trust. We can define collision probability of hash function using the birthday paradox at most $\frac{q_h^2}{2^{l_h+1}}$. Each communicated message in the proposed protocol $\mathcal{P}$ is built up using the random nonces (N^y_d, N^x_d), random numbers (r^x_d,$r^x_d y$) and timestamps (T_i). The collision probability for these values is at most $\frac{(q_s+q_e)^2}{2^{l_s+1}}$. Thus, the game Gm_2 and the game Gm_1 are identical games till the collision arises; hence,

$$Pr[Suc_2] - Pr[Suc_1] \leq \frac{q_h^2}{2^{l_h+1}} + \frac{(q_s+q_e)^2}{2^{l_s+1}}, \tag{8}$$

Game Gm_3: In this game, challenger C performs the *corrupt* query $CR(O^m_{D_x})$:) and *send* query $\mathcal{S}$ or an *execute* query $\mathcal{E}$ with the random oracles. The challenger also tries to solve the ECDH problem of the ECC. Let us consider that the challenger C tries the following queries,

- Using $CR(O^m_{D_x})$**):** query, the challenger retrieves $\langle ID^x_d, r^x_d, K_{dta}, DP^x_1 \rangle$
- Using $CR(O^m_{D_y})$**):** query, the challenger retrieves $\langle ID^x_d, DP^x_1, K_{dta}, Q^x_d \rangle$
- Using $\mathcal{E}(D_x, TA)$**):** query, the challenger retrieves $\langle ID^x_T, P^x_d, T_1 \rangle$, $\langle Z^y, T_4 \rangle$.

- Using $\mathcal{E}(TA, D_y)$**:** query, the challenger retrieves $\langle Z^x, P_d^y, T_2 \rangle, \langle Z^y, P_d^{TA}, T_3 \rangle$.

After performing the following queries for a finite time, the challenger tries to decrypt the data encrypted by the public keys $\{Q_d^x, Q_d^y\}$. These public keys are computed as $Q_d^x = r_d^x$*P and $Q_d^y = r_d^y$*P. For $\{Q_d^x, Q_d^y\}$ and P, it is computationally infeasible to find the value of $\{r_d^x, r_d^y\}$. The probability of solving the ECDH problem is at most (4*q_e+2*q*s)$Adv_{C^{ECDH}}(t^*)$. The probability of guessing the correct random nonces (N_d^y, N_d^x) after performing the $CR(O_{D_x}^m)$) and $CR(O_{D_y}^m)$) is at most max(q_s, ($\frac{1}{2^l}$, ρ_{fp})). It is infeasible for challenger to solve the ECDH problem and guess the correct random numbers simultaneously in polynomial time. Hence, the game Gm_3 is identical to the game Gm_2. Thus we have,

$$Pr[Suc_3] - Pr[Suc_2] \leq (4 * q_e + 2 * q * s) \\ Adv_{C^{ECDH}}(t^*) + max(q_s, (\frac{1}{2^l}, \rho_{fp})) \quad (9)$$

Now, challenger C tries to guess the bit u' and the probability of correct guess is at most $\frac{1}{2}$. Thus, from equations 8 and 9, we can derive

$$Adv_{\mathcal{P}}(C) \leq \frac{q_h^2}{2^{l_h+1}} + \frac{(q_s + q_e)^2}{2^{l_s+1}} + (4 * q_e + 2 * q * s) \\ Adv_{C^{ECDH}}(t^*) + max(q_s, (\frac{1}{2^l}, \rho_{fp})) \quad (10)$$

□

8. Implementation using MQTT

The Message Queuing Telemetry Transport(MQTT) protocol is a widely adopted publish-subscribe-based, lightweight application layer protocol for communicating in the IoT-based environment. In the MQTT protocol, there are three entities, (1) The publisher (who publishes the data), (2) The subscriber (who receives the data), and the broker (who integrates and forwards the data). To implement the proposed protocol, we used Raspberry Pi 3 Model B (with Quad Core 1.2GHz Broadcom BCM2837 64bit CPU and 1GB RAM) as a sensing device and the laptop device installed with the mosquitto broker on it. We can also utilize global brokers (such as AWS and hivemq). For sniffing purposes, we utilized laptop devices and installed the mosquitto broker and Wireshark tool over it. We used the *Paho* library that provides MQTT client services. We implemented the proposed EBAKE-SE using 15 sensing devices (Raspberry Pis) that establish session keys with each other. Fig. 3 shows the final computed session key between the IoT device D_x and the IoT device D_y.

Table 3: Network Model and Cryptographic Operations

Scheme	Model	OP_1	Op_2	Op_3	Op_4	Op_5	Op_6
Ours	D-TA-D	2	4	11	2	-	-
[17]	D-D	-	-	12	-	12	-
[30]	U-G-D	-	-	19	9	6	-
[34]	MD-MD-S	3	-	9	2	13	-
[36]	U-G-D	-	1	22	11	6	-

OP^1: Symmetric Encryption/Decryption, OP^2: Asymmetric Encryption/Decryption, OP^3: Hash function, OP^4: XOR operation, OP^5: ECC point multiplication operation, OP^6: ECC point summation operations, U: User, GW:Gateway, TA:Trusted Authority, D: Sensing device, S: Server, MD: Mobile device.

Figure 3: Session key computation

The MQTT protocol works with three kind of quality of services: *QoS 0* (at most once), *QoS 1* (at least once) and *QoS 2* (exactly once) for packet transmissions. As mentioned earlier, we collected average throughput, packet delivery ratio, and round-trip delay for the setup by analysing the data collected using a wireshark tool. We define the average throughput as an average number of packets transmitted and successfully received per unit time. We observed that the average throughput of the proposed setup was 643 packets per minute. The average packet delivery ratio was around 99.34%, and the packet loss is 0.66%. The average packet delivery ratio may be reduced if we use a global broker. The range of round-trip delay (from D_x to TA, TA to D_y, D_y to TA, and TA to D_x) was around 45 ms - 70 ms because of less computation of the proposed EBAKE-SE protocol.

9. Comparative analysis

In this section, we will compare and analyze the proposed scheme according to the number of cryptographic operations, computation time (in ms) and communication cost (in bits) to emphasize the computational efficiency of the proposed scheme. We compare the proposed EBAKE-SE with other recently proposed schemes for a similar environment.

9.1. Cryptographic Operations

Table 3 highlights the comparative analysis of the proposed scheme (only authentication phase) with other existing schemes based on the number of cryptographic operations required.

9.2. Computation Time

Table 4 highlights a comparative analysis of EBAAC-SE with other existing schemes based on the computation time required by the scheme. In the initial phase of our implementation, we collected results for basic cryptographic operations. These results are collected for the environment discussed in Section 8. Observations of these computations were as follows: the time required for the single hash function using SHA was (T_h) 0.043 ms. The time required by a single elliptic curve point addition operation was (T_{pa}) 0.068 ms. The time required by a single elliptic curve point multiplication operation was (T_{pm}) 12.226 ms. The time required for single symmetric encryption over AES was (T_{sym}) 0.046ms. The time required by single ECC encryption is ($T_{asym} \approx T_{pm}$) 12.268 ms. Based on these observations, in Table 4, we highlight a computation time-based comparison between the proposed scheme and other existing schemes.

10. Conclusions and future work

In this paper, we proposed an ECC-based authenticated key exchange scheme between two Industrial IoT devices via trusted authority. We use a tamper-proof microprocessor called a Secret Element (SE) to store the secret parameters of sensing devices. We provided cryptanalysis for the RUA scheme proposed by Das et al. for a similar environment and highlighted numerous vulnerabilities, such as MITM attacks and impersonation attacks. Afterwards, we offered an RUA using ECC between two advanced-IoT devices via cloud trusted authority. We presented informal security analysis as well as formal analysis on EBAKE-SE. We compared the presented EBAKE-SE with existing schemes based on security features, computation time, and several cryptography operations. Furthermore, we presented an implementation environment using the publish-subscribe-based MQTT protocol. The numerous IoT-based industries (such as smart homes, smart healthcare, smart transport, smart security, and surveillance system) can use the proposed EBAKE-SE to enhance their security mechanism with acceptable reliability and efficiency.

Acknowledgements

This work was supported by the Researchers Supporting Project (No. RSP-2021/395), King Saud University, Riyadh, Saudi Arabia.).

References

[1] H. Wang, X. Li, R. H. Jhaveri, T. R. Gadekallu, M. Zhu, T. A. Ahanger, S. A. Khowaja, Sparse bayesian learning based channel estimation in fbmc/oqam industrial iot networks, Computer Communications (2021).

[2] M. Alkhelaiwi, W. Boulila, J. Ahmad, A. Koubaa, M. Driss, An efficient approach based on privacy-preserving deep learning for satellite image classification, Remote Sensing 13 (2021). doi:10.3390/rs13112221.

[3] R. Jhaveri, R. Sagar, G. Srivastava, T. R. Gadekallu, V. Aggarwal, Fault-resilience for bandwidth management in industrial software-defined networks, IEEE Transactions on Network Science and Engineering (2021).

[4] M. Driss, D. Hasan, W. Boulila, J. Ahmad, Microservices in iot security: Current solutions, research challenges, and future directions, Procedia Computer Science 192 (2021) 2385–2395. doi:https://doi.org/10.1016/j.procs.2021.09.007, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021.

[5] A. Ali, H. Rafique, T. Arshad, M. A. Alqarni, S. H. Chauhdary, A. K. Bashir, A fractal-based authentication technique using sierpinski triangles in smart devices, Sensors 19 (2019) 678.

[6] C. C. Sobin, A survey on architecture, protocols and challenges in iot, Wireless Personal Communications 112 (2020) 1383–1429. URL: https://doi.org/10.1007/s11277-020-07108-5. doi:10.1007/s11277-020-07108-5.

[7] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A survey, Computer Networks, Elsevier 54 (2010) 2787–2805. URL: http://dx.doi.org/10.1016/j.comnet.2010.05.010. doi:10.1016/j.comnet.2010.05.010. arXiv:arXiv:1011.1669v3.

[8] A. Al-fuqaha, S. Member, M. Guizani, M. Mohammadi, S. Member, Internet of Things : A Survey on Enabling, IEEE Communication 17 (2015) 2347–2376.

[9] P. Sethi, S. R. Sarangi, Internet of Things: Architectures, Protocols, and Applications, Journal of Electrical and Computer Engineering, Hindawi 2017 (2017). doi:10.1155/2017/9324035.

[10] Y. Hajjaji, W. Boulila, I. R. Farah, I. Romdhani, A. Hussain, Big data and iot-based applications in smart environments: A systematic review, Computer Science Review 39 (2021) 100318. doi:https://doi.org/10.1016/j.cosrev.2020.100318.

[11] S. B. Atitallah, M. Driss, W. Boulila, H. B. Ghzala, Leveraging deep learning and iot big data analytics to support the smart cities development: Review and future directions, Computer Science Review 38 (2020) 100303. doi:https://doi.org/10.1016/j.cosrev.2020.100303.

[12] C. Maple, Security and privacy in the internet of things, Journal of Cyber Policy 2 (2017) 155–184. doi:10.1080/23738871.2017.1366536.

[13] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, E. K. Markakis, A survey on the internet of things (iot) forensics: Challenges, approaches and open issues, IEEE Communications Surveys & Tutorials (2020).

[14] A. Irshad, M. Usman, S. A. Chaudhry, A. K. Bashir, A. Jolfaei, G. Srivastava, Fuzzy-in-the-loop-driven low-cost and secure biometric user access to server, IEEE Transactions on Reliability (2020).

[15] M. binti [Mohamad Noor], W. H. Hassan, Current research on internet of things (iot) security: A survey, Computer Networks 148 (2019) 283 – 294. doi:https://doi.org/10.1016/j.comnet.2018.11.025.

[16] R. Arul, G. Raja, A. K. Bashir, J. Chaudry, A. Ali, A console grid leveraged authentication and key agreement mechanism for lte/sae, IEEE Transactions on Industrial Informatics 14 (2018) 2677–2689.

[17] A. K. Das, M. Wazid, A. R. Yannam, J. J. P. C. Rodrigues, Y. Park, Provably secure ecc-based device access control and key agreement protocol for iot environment, IEEE Access 7 (2019) 55382–55397.

[18] N. M. F. Qureshi, I. F. Siddiqui, A. Abbas, A. K. Bashir, C. S. Nam, B. S. Chowdhry, M. A. Uqaili, Stream-based authentication strategy using iot sensor data in multi-homing subaqueous big data network, Wireless Personal Communica-

Table 4: Computation time

Scheme	Device-1	Gateway/ cloud	Device-2	Total operations	Total time
Proposed	$T_{sym} + 2T_{asym} + 3T_h$	$T_{sym} + 5T_h$	$2T_{asym} + 3T_h$	$2T_{sym} + 4T_{asym} + 11T_h$	49.469 ms
[17]	$6T_{pm} + 6T_h + 2T_{pa}$	-	$6T_{pm} + 6T_h + 2T_{pa}$	$12T_{pm} + 12T_h + 4T_{pa}$	147.5 ms
[33]	$5T_{pm} + 4T_h$	$4T_{pm} + 3T_h$	$14T_{pm} + 12T_h$	$23T_{pm} + 19T_h$	171.68 ms
[30]	$3T_{pm} + 8T_h$	$T_{pm} + 7T_h$	$2T_{pm} + 4T_h$	$6T_{pm} + 19T_h$	74.173 ms
[34]	$6T_{pm} + 4T_h + 2T_{sym}$	$4T_{pm} + 5T_h$	$3T_{pm} + T_{sym}$	$13T_{pm} + 9T_h + 3T_{sym}$	159.454 ms
[36]	$3T_{pm} + 10T_h + T_{asym}$	$T_{pm} + 8T_h$	$2T_{pm} + 8T_h$	$6T_{pm} + 22T_h + T_{asym}$	86.528 ms

tions 116 (2021) 1217–1229.

[19] D. Sethia, D. Gupta, H. Saran, Nfc secure element-based mutual authentication and attestation for iot access, IEEE Transactions on Consumer Electronics 64 (2018) 470–479. doi:10.1109/TCE.2018.2873181.

[20] C. Patel, D. Joshi, N. Doshi, A. Veeramuthu, R. Jhaveri, An enhanced approach for three factor remote user authentication in multi-server environment, Journal of Intelligent & Fuzzy Systems (????) 1–12.

[21] V. S. Miller, Use of elliptic curves in cryptography, in: Conference on the theory and application of cryptographic techniques, Springer, 1985, pp. 417–426.

[22] N. Koblitz, Elliptic curve cryptosystems, Mathematics of computation 48 (1987) 203–209.

[23] P. K. Dhillon, S. Kalra, Secure and efficient ecc based sip authentication scheme for voip communications in internet of things, Multimedia Tools and Applications 78 (2019) 22199–22222.

[24] D. Kumar, H. S. Grover, et al., A secure authentication protocol for wearable devices environment using ecc, Journal of Information Security and Applications 47 (2019) 8–15.

[25] A. Lohachab, et al., Ecc based inter-device authentication and authorization scheme using mqtt for iot networks, Journal of Information Security and Applications 46 (2019) 1–12.

[26] M. Qi, J. Chen, Y. Chen, A secure authentication with key agreement scheme using ecc for satellite communication systems, International Journal of Satellite Communications and Networking 37 (2019) 234–244.

[27] S. Garg, K. Kaur, G. Kaddoum, K.-K. R. Choo, Towards secure and provable authentication for internet of things: Realizing industry 4.0, IEEE Internet of Things Journal (2019).

[28] M. Dammak, O. R. M. Boudia, M. A. Messous, S. M. Senouci, C. Gransart, Token-based lightweight authentication to secure iot networks, in: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2019, pp. 1–4.

[29] T. K. Dang, C. D. Pham, T. L. Nguyen, A pragmatic elliptic curve cryptography-based extension for energy-efficient device-to-device communications in smart cities, Sustainable Cities and Society 56 (2020) 102097.

[30] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, S. Kumari, A robust ecc-based provable secure authentication protocol with privacy preserving for industrial internet of things, IEEE Transactions on Industrial Informatics 14 (2017) 3599–3609.

[31] P. Kocher, J. Jaffe, B. Jun, Differential power analysis, in: Annual international cryptology conference, Springer, 1999, pp. 388–397.

[32] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, B. Stiller, Landscape of iot security, Computer Science Review 44 (2022) 100467.

[33] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E.-J. Yoon, K.-Y. Yoo, Secure signature-based authenticated key establishment scheme for future iot applications, IEEE Access 5 (2017) 3028–3043.

[34] L. Wu, J. Wang, K.-K. R. Choo, D. He, Secure key agreement and key protection for mobile device user authentication, IEEE Transactions on Information Forensics and Security 14 (2018) 319–330.

[35] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Yliantila, B. Stiller, Group key establishment for enabling secure multicast communication in wireless sensor networks deployed for iot applications, IEEE Access 3 (2015) 1503–1511.

[36] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, C. Chen, A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems, IEEE Systems Journal 14 (2019) 39–50.

Conflict of interest:

We authors confirm that there is no conflict of interest associated with research work carried out with any means.

Further, we confirm, the article not submitted elsewhere to any journal or conference, solemnly set for, DCN Journal for review process.