Adversarial Attacks Resilient Machine Learning-Enabled Access Control Scheme for Consumer Electronics in IoT-Based Applications

Basudeb Bera, Abhishek Bisht, Ashok Kumar Das, *Senior Member, IEEE*,
Sandip Roy, Sachin Shetty, *Senior Member, IEEE*, Biplab Sikdar, *Fellow, IEEE*

***Abstract*—Consumer Electronics (CE) have become integral to the evolving Internet of Things (IoT) ecosystem, providing enhanced security, interoperability, and intelligence through machine learning (ML) to improve IoT functionality and user experiences. However, as CE devices manage sensitive data, security vulnerabilities emerge, particularly due to the use of wireless communication channels, which are often insecure. The lack of security in ML training and testing on CE-related datasets heightens risks such as data poisoning and adversarial attacks. In this article, a resilient machine learning-based access control scheme (RMACE-IoT) is designed to address these critical security challenges in IoT applications. RMACE-IoT not only strengthens the security of data exchanges but also defends against adversarial attacks that could undermine ML model predictions. The proposed RMACE-IoT is validated through a real-time testbed experiment, showcasing its effectiveness in enhancing resilience and mitigating adversarial threats. The proposed scheme (Case 1) achieves communication cost reductions of up to 40.00%, 60.42%, 61.49%, 65.87%, and 81.73% when compared with the schemes of Luo et al., Ali and Pal, Wang et al., Ever, and Zhang et al., respectively. In terms of computational cost, the proposed scheme demonstrates improvements of 99.90%, 52.76%, 52.29%, 99.95%, and 45.70% over the corresponding schemes. These results demonstrate the practical applicability and advantages of RMACE-IoT in real-time IoT scenarios. This work significantly advances the field by addressing pressing security concerns in CE systems, paving the way for safer, more reliable, and intelligent consumer devices.**

***Index Terms*—Consumer electronics, Internet of Things (IoT), access control, machine learning, testbed, security.**

I. Introduction

CONSUMER Electronics (CE) encompass cutting-edge and inventive devices tailored to meet the evolving requirements of the upcoming Internet of Things (IoT) environment [1]. These devices are specifically designed to meet the requirements of IoT applications across various industries. IoT applications involve interconnecting diverse physical devices, objects, and systems through the Internet, enabling them to collect and exchange data [2]. CE devices are expected to offer improved connectivity, interoperability, and intelligence. Such devices may include smart home appliances, smart drones solutions, smart wearables like smartwatches, fitness trackers, and health monitors, connected cars, smart speakers, and other IoT-enabled devices. These devices will provide enhanced automation, control, and integration capabilities with other devices and systems. The primary goal of CE is to enhance the user experience and provide improved functionality for IoT applications. This is achieved by leveraging advanced technologies such as advanced connectivity, edge computing, and artificial intelligence or machine learning (AI/ML) capabilities, including voice recognition, natural language processing, and intelligent assistants. CE devices also incorporate enhanced sensors and energy-efficient features to deliver improved functionality, connectivity, and user experience within the context of IoT applications. These advanced connectivity options offer benefits such as faster data transmission, lower latency, increased network capacity, improved coverage, and enhanced security [3], [4], [5].

It is worth noting that CE devices encompass advanced sensors designed to gather precise and diverse data from various IoT applications. The incorporation of the environmental sensors enhanced sensor capabilities enables more accurate monitoring and tracking within IoT applications. Furthermore, CE devices have the potential to incorporate AI and ML capabilities. These technologies empower IoT devices to analyze and interpret data, make intelligent decisions, and provide personalized experiences to users. Examples of such experiences include predictive maintenance, anomaly detection, and smart automation. The integration of AI and ML within CE devices enhances their ability to process data and derive valuable insights [3], [6].

Since CE devices in IoT applications collect confidential and private data, ensuring the security of this data becomes a significant concern [2], [7], [8]. IoT devices are typically connected through insecure wireless media or public channels, further exacerbating the security risks associated with data collection in IoT applications. Additionally, when adopting AI/ML approaches on the collected data over such insecure channels, concerns regarding performance arise due to the

Manuscript received xxx; revised xxx; accepted xxx. *(Corresponding authors: Ashok Kumar Das, Basudeb Bera)*

Basudeb Bera and Biplab Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: b.bera26@nus.edu.sg; bsikdar@nus.edu.sg).

Abhishek Bisht is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India, and also with the Department of Computer Science and Engineering, Indiana University, Bloomington, IN 47405, USA (e-mail: abhishek.bisht@research.iiit.ac.in).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India (e-mail: iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in).

Sandip Roy and Sachin Shetty are with the Virginia Modeling, Analysis and Simulation Center, Department of Computational Modeling and Simulation Engineering, Old Dominion University, Suffolk, VA 23435, USA (e-mail: sroy@odu.edu; sshetty@odu.edu).

absence of security measures on the ML training/testing dataset. This lack of security measures can ultimately reduce the accuracy and performance of the ML model. Numerous studies [9], [10], [11] have reported various attacks on ML execution. These attacks include data poisoning attacks and adversarial attacks, both of which involve deliberate and malicious attempts to manipulate or deceive machine learning models [12]. Adversaries exploit vulnerabilities or weaknesses in the model's decision-making process to generate misleading inputs or perturbations that can mislead the model's predictions [9], [13].

A. Motivation

Traditional machine learning models are typically trained and tested on clean and static dataset, assuming a stationary data distribution. However, in real-world IoT scenarios, data can be subjected to various forms of perturbations and changes over time, including the presence of heterogeneous data types. For CE in IoT applications, we have proposed a resilient machine learning (ReML) based access control scheme. This scheme aims not only to enhance the security of data exchanges but also to safeguard against misleading the predictions of the ML model. ReML is particularly relevant in our proposed scheme due to the criticality of security and integrity in machine learning systems within the context of IoT applications. In this context, ReML refers to the development and deployment of machine learning models that exhibit robustness and the ability to maintain their performance and accuracy in the face of various challenges. These challenges include adversarial attacks, noisy or incomplete data, and concept drift. The objective of ReML is to ensure that machine learning models can effectively handle unexpected situations and resist adversarial attempts to manipulate or exploit the dataset [14], [15].

B. Main Contributions

The major contributions of this work are as follows:

- The proposed protocol offers a mechanism for secure mutual authentication to establish session keys among the communicating parties (smart devices, gateway nodes, and cloud servers).
- The proposed scheme provides a resilient machine learning approach, which is a novel approach that identifying the scheme's uniqueness, where adversarial attacks may be possible during the training and testing processes.
- Security analysis and formal security verification under Scyther Tool is used to demonstrate resistance against various active and passive attacks.
- The proposed scheme offers a comprehensive access control mechanism, incorporating two levels of access control. The first level manages access between a smart device and a gateway node, while the second level oversees access between the gateway node and the associated cloud server.
- The proposed scheme introduces a new node addition phase, demonstrating the scheme's scalability and highlighting the differences from previous research works in consumer electronics.
- The proposed scheme establishes a secure data exchange phase to counter adversarial attacks during ML model training. This is achieved through collaboration with a trusted but curious cloud server.

In IoT-based applications for consumer electronics, access control is crucial for securing devices like smart home systems, wearables, voice assistants, appliances, and healthcare devices. For instance, smart locks and thermostats use biometrics or passwords for user authentication, while machine learning can detect adversarial attacks like spoofing or unusual usage patterns. Similarly, in smart homes, where devices like security cameras, smart locks, and sensors are interconnected, adversarial attacks can attempt to bypass traditional access controls by manipulating the device's behavior or input data. A machine learning-enabled access control system, resilient to such attacks, can continuously adapt and recognize abnormal patterns or malicious inputs, preventing unauthorized access even when attackers attempt to exploit vulnerabilities. Similarly, in wearable devices, which store sensitive personal health data, adversarial attacks could target access protocols to steal or tamper with information. By using a robust machine learning model that can identify subtle changes or patterns in access requests, the scheme ensures that only legitimate users gain access, effectively preventing both unauthorized access and data breaches.

C. Article Structure

In Section II, we delve into the existing literature on access control schemes for IoT applications. Section III introduces an integrated system model encompassing network and threat models. Section IV explores the various phases of the proposed scheme, while Section V delves into the security analysis. In Section VI, we provide insights into testbed experiments conducted for the proposed scheme and ML simulation results. Section VII offers a comprehensive comparative study, highlighting the scalability and feasibility of our proposed scheme in comparison to other existing schemes. Finally, Section VIII presents key concluding remarks regarding the proposed scheme.

II. Related Work

Luo et al. [16] proposed an access control scheme for the cross-domain context of the IoT environment. Their scheme enhances security and efficiency in WSNs, enabling cross-domain communication between an Internet user in a certificateless cryptography (CLC) environment and a sensor node in an identity-based cryptography (IBC) environment, despite their differing system parameters.However, their scheme is vulnerable to device physical capture attacks and Ephemeral Secret Leakage (ESL) attacks under the CK-adversary model. Chaudhry et al. [17] presented a device access control protocol for IoT applications integrated with a cloud system. In their scheme, two smart devices establish a session key by exchanging their real identities over an insecure public channel. However, this approach does not protect anonymity for the devices. Masud et al. [18] suggested an authentication scheme for healthcare applications based on IoT. Their scheme

creates a secure session for authorized users while preventing unauthorized individuals from accessing the IoT sensor nodes. However, their scheme is vulnerable to privileged-insider attacks. Additionally, their scheme lacks support for the new smart device addition phase.

Li et al. [19] proposed an access control scheme for industrial IoT applications with cloud assistance. However, their scheme, which relies on bilinear pairing, incurs significant computational costs, rendering it inefficient for practical applications. A similar access control scheme was proposed by Xiong et al. [20] for an IoT-based cloud storage system. However, their scheme also utilizes a bilinear pairing-based approach for information exchange, resulting in significant computation overhead and inefficiency. Ali and Pal [21] proposed a user authentication protocol that utilizes a remote server in a multi-server environment. Their scheme involves three network entities: users, a registration center (RC), and servers, which communicate wirelessly. Users authenticate themselves to servers using smart cards and establish session keys before accessing services. However, Luo et al. [22] highlighted several security vulnerabilities in Ali and Pal's solution. These vulnerabilities include user and server impersonation, smart card theft, and replay attacks, user anonymity leakage, insider attacks, and a lack of forward secrecy.

Ma et al. [23] suggested an authentication scheme designed specifically for vehicular ad-hoc networks. Their scheme involves the mutual authentication of vehicle users, fog servers, and cloud servers, enabling the establishment of a session key for secure information sharing. However, Eftekhari et al. [24] discovered that the security solution proposed by Ma et al. [23] is susceptible to various attacks, including insider attacks, known session specific temporary information attacks, and stolen smart card attacks. Ever [25] proposed an authentication scheme specifically designed for Unmanned Aerial Vehicles (UAVs) acting as mobile sinks in an Internet of Drones (IoD) environment. The scheme assumes the presence of sensor nodes, cluster heads (CHs), mobile sinks (UAVs), and a base station (or server) as the communicating entities in the IoD network. Sensor nodes are connected to the nearest CH, which can establish connections with other nodes and UAVs. The UAVs collect sensing data from the CHs and transmit it to the base station. The communication between network entities relies on a session key established through bilinear pairing, which demands significant computational resources. However, the scheme proposed by Ever [25] lacks security against ESL attacks under the CK-adversary model, as well as anonymity and dynamic node addition functionality features.

Ren et al. [26] proposed an authentication and key agreement protocol designed for 5G networks. Their protocol incorporates an anonymous ticket generation mechanism utilizing the trapdoor collision property of chameleon hash functions. To reduce computational overhead during the authentication phase, the scheme employs a ticket validation process that utilizes a ring signature. However, a significant drawback of their scheme is its vulnerability to replay attacks, which undermines its overall security. Fan et al. [27] proposed an authentication protocol specifically designed for cross-network slicing in mobile communication. Their protocol enables end users to access various services provided by network slicing, with authentication taking place through the 5G core network, acting as a decentralized edge cloud, to minimize latency. The communication involves the user, slice base station, and operator to establish session keys. However, their proposed session key between the user and operator suffers from vulnerabilities, including an ESL attack under the CK-adversary model and a replay attack. Additionally, the model lacks support for entity anonymity.

Wang et al. [28] designed a key agreement protocol tailored for smart healthcare applications. This protocol involves users, functioning as patients equipped with resource-limited smart medical devices, gathering and transmitting health data to the connected edge server through the negotiation of a session key. However, Chang et al. [29] pointed out that Wang et al.'s scheme lacks privacy assurance and is susceptible to Denial-of-Service (DoS) attacks. In 2021, Yuanbing et al. [30] proposed a key agreement and authentication protocol for IoT-based smart healthcare through a wireless medical sensor network. Their scheme protects information privacy and security from being stolen by unauthorized users, safeguards against privileged insider attacks, prevents user anonymity invalidation, and defends against offline password guessing attacks. However, security concerns arise as their scheme generates the session key using public information, random numbers, and identities. Furthermore, Lee et al. [31] identified security vulnerabilities, including susceptibility to smart card stolen attacks, offline ID/password guessing attacks, user impersonation attacks, sensor node impersonation attacks, man-in-the-middle attacks, sensor node capture attacks, and a lack of user anonymity assurance. In 2023, Wang et al. [32] developed an authentication and key agreement framework designed for IoT applications assisted by the cloud. In their proposed system, IoT devices, users, gateway nodes, and cloud centers undergo mutual authentication before initiating a session key. The session key is generated using a combination of a hash function and ECC, incorporating random numbers and public parameters. However, this approach exposes vulnerabilities to ESL attacks under the CK-adversary model. Furthermore, their scheme lacks resistance against replay attacks.

In 2022, Meher et al. [33] proposed an authentication scheme relying on the elliptic curve discrete logarithm problem (ECDLP) between a passive radio-frequency identification (RFID) tag and a RFID reader for a consumer warehouse management system. Their scheme uses the Elliptic Curve Discrete Logarithm Problem (ECDLP) for a key-less authentication system in Warehouse Management Systems (WMS), optimizing memory usage in tags and servers. However, their proposed model, which incorporates random nonce and identities, is susceptible to potential risks. Notably, an adversary can reveal these identities at any time, rendering the scheme vulnerable to ESL attacks under the CK-adversary model. Furthermore, the proposed scheme lacks protection against replay attacks. In 2023, Mahmood et al. [34] proposed an access control scheme for AI-driven intelligent flying vehicles in consumer electronics. Their scheme introduces a neural computing-based access control protocol designed to protect AI-driven flying vehicles from potential security threats. Unfortunately, their

scheme lacks the ability to resist forward secrecy and having issue in the session key as it is not verified before utilization.

Zhang et al. [35] developed a user authentication protocol for the Internet of Drones (IoD), leveraging FourQ curves and BPV pre-calculation techniques to enhance data confidentiality. However, as pointed out by Park et al. [36], the protocol faces significant drawbacks, including high computational and communication overhead, susceptibility to replay attacks, and a lack of user anonymity. Mahmood et al. [37] proposed an authentication and key agreement protocol that allows two smart grid users to establish a session key, relying on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) for security. While the protocol is resistant to Denial of Service (DoS) attacks, it is vulnerable to ESL attacks [38]. Moreover, Mahmood et al.'s protocol does not ensure user anonymity or perfect forward secrecy [38]. In 2023, Zhang et al. [39] designed a three-factor authentication and key agreement scheme for an IoT-based e-healthcare application. However, their scheme is susceptible to replay attacks and does not support a dynamic addition of nodes.

In 2024, Aldosary et al. [40] proposed an authentication mechanism for mobile crowdsourcing networks (MCNs) that utilizes a combination of symmetric key encryption, ECC, and one-way hash functions. In their scheme, the service provider verifies the user's authenticity before allowing them to join the MCN and establishes a secure session key for encrypted data communication. However, their scheme does not support dynamic node addition, which raises concerns about scalability in real-time applications.

In 2024, Fan et al. [41] developed an authentication and key exchange framework for smart industrial devices based on physically unclonable functions (PUFs). Their scheme uses a one-way hash function, ECC, and PUFs to establish a Diffie-Hellman-type session key between smart devices and the server. However, this session key is constructed relying solely on random numbers. As a result, the key is vulnerable to ESL attacks under the CK-adversary model. Furthermore, their scheme lacks resistance to replay attacks and MiTM attacks, and it cannot preserve anonymity or untraceability due to the leakage of the device's identity.

The summarized key differences between RMACE-IoT and other existing access control schemes, in terms of cryptographic methods, advantages, and limitations, are provided in Table I of the Supplementary Material.

III. System Model

In this section, we introduce a comprehensive framework that combines the network model and security model, two fundamental elements of our innovative approach. The subsequent explanation provides an overview of the distinct responsibilities and capabilities associated with each component in our proposed framework.

A. Network Model

The network model employed in our RMACE-IoT scheme is depicted in Fig. 1, showcasing the architecture and interactions among the key components. Within this model, we incorporate smart devices (SD), gateway nodes (GN), and cloud servers (CS) as the primary entities. Before their integration into the IoT application, a trusted authority (TA) registers these components, assigning them unique private and public parameters. The smart devices are responsible for collecting sensor data using their built-in capabilities and transmitting it to the nearest GN. To ensure data integrity and authenticity, the smart devices include a digital signature along with the transmitted data. The GN verifies the data integrity by validating the signature, and subsequently forwards the data to the associated CS. The CS acts as the repository, where it collects and stores the received data from the smart devices and performs future ReML tasks. The data security is maintained through the utilization of signatures, ensuring that only authentic data is used for ReML purposes. In this network model, the CS is considered trusted but curious, implying that it may have access to the data but is expected to handle it with confidentiality. The gateway node (GN) is semi-trusted, while the trusted authority (TA) is fully trusted, serving as a reliable entity throughout the system.

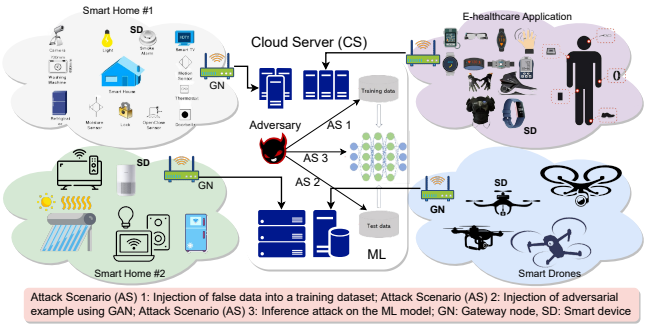

Fig. 1. Network model for ReML based access control for CE in IoT applications.

B. Threat Model

In the proposed RMACE-IoT scheme, the network components engage in communication via an insecure public channel or wireless media. A SD is responsible for collecting sensing data and transmitting it to the nearest GN by establishing a session key through access control. Subsequently, the GN verifies the integrity of the received data by examining the signature provided by the SD and forwards it to the associated CS using a similar session key through access control. However, since all these communications occur over an insecure channel, a security threat arises. Within this model, we take into consideration two well-established security threat models: Dolev-Yao (DY) [42] and Canetti and Krawczyk (CK) [43]. Under the DY-threat model, an adversary (denoted as $\mathcal{A}$) possesses the capability to eavesdrop on communication messages transmitted through the insecure channel. Furthermore, $\mathcal{A}$ can manipulate, delete, or introduce malicious content into the communication channel. Moreover, the CK-adversary threat model grants $\mathcal{A}$ additional advantages beyond those provided by the DY-threat model. In this threat model, $\mathcal{A}$ not only has the ability to modify, intercept, delete, or inject malicious information into the communicated messages but can also

seize control of short-term and long-term secrets, as well as session states. Additionally, we consider a hostile scenario where $\mathcal{A}$ can physically capture a SD and launch side-channel attacks, such as power analysis attacks [44], to extract stored information from the compromised device SD's memory.

The extended CK (eCK) adversary model is rooted in the CK adversary model, it surpasses the CK model by conferring additional powers and capabilities to $\mathcal{A}$, transforming it into a more formidable adversary. These supplementary capabilities may include actively executing diverse query sequences, like a Session Key Reveal query aimed at a specific session ID (e.g., sid), thereby compromising the freshness of the session. In the event that a session such as sid or its corresponding session sid' is compromised in the eCK model, it is considered exposed by $\mathcal{A}$. In the Honest-But-Curious adversary threat model, $\mathcal{A}$ is presumed to adhere to the protocol instructions honestly, abstaining from any deviation from the specified protocol steps or engagement in malicious activities, such as attempting to compromise encryption or tamper with transmitted data. However, the $\mathcal{A}$ can be characterized as "Curious," signifying their interest in acquiring as much information as legitimately accessible within the protocol's rules. $\mathcal{A}$ exhibits curiosity regarding the exchanged data and endeavors to gather information within the protocol-defined boundaries.

It is important to acknowledge potential attacks on the machine learning execution within this model. Adversaries may attempt to inject malicious data during the training process, manipulate the testing model, or launch inference attacks on the machine learning model itself.

TABLE I
NOTATIONS AND THEIR MEANINGS

Notation	Significance
TA, SD, GN, CS	Trusted authority, smart device, gateway node, and cloud server
$E_n(p,q)$	A non-singular elliptic curve of the form: "$y^2 = x^3 + px + q \pmod n$, $p, q \in Z_n$, and $4p^3 + 27q^2 \neq 0 \pmod n$"
G	A base point in $E_q(a,b)$ whose order is α as big as n
ID_i, TID_i, SID_i	i^{th} SD's real identity, temporal identity, and pseudo-identity
s_i, Pub_{s_i}	$s_i \in Z_n^*$ and $Pub_{s_i} = s_i \cdot G$ are SD's private and public keys
m_t	$m_t \in Z_n^*$ is a master private key of TA
$f_(x,y)$	A bivariate polynomial of the form: $f_(x,y) = \sum_{u=0}^{t}\sum_{v=0}^{t} a_{u,v}x^u y^v \pmod n$ over Galois field $GF(n)$, and $t \gg \#GN$s
ID_j, TID_j, SID_j	j^{th} GN's real identity, temporal identity, and pseudo-identity
RTS_i, RTS_j, RTS_l	Registration timestamp of SD_i, GN_j, and CS_l
TS_i	"Current timestamps for i = 1, 2, 3"
ΔT	"Maximum message transmission delay"
$h(\cdot)$	"Collision-resistant cryptographic one-way hash function"
$SK_{ij}(= SK_{ji})$, SKV	Session key between SD_i and GN_j and corresponding session key verifier
$SK_{jl}(= SK_{lj})$	Session key between GN_j and CS_l

IV. THE PROPOSED FRAMEWORK

This section introduces our proposed access control scheme, RMACE-IoT, which leverages machine learning in consumer electronics for IoT applications. Notations and their meanings are presented in Table I, which is utilized to describe the proposed scheme. The scheme comprises of several phases, which are outlined below.

A. Initial Setup Phase

A TA is responsible for selecting initial system parameters for registering the IoT entities, as follows.

- The TA selects a finite field as Galois field $GF(n)$ randomly, where n is a sufficiently large odd prime. This choice ensures that computational problems involving elliptic curves, such as the elliptic curve discrete logarithm problem (ECDLP) and the elliptic curve decisional Diffie-Hellman problem (ECDDHP), become intractable. It is recommended that n be at least 160 bits in size to achieve this level of security.
- The TA proceeds by selecting a non-singular elliptic curve, denoted as $E_n(p,q)$ of the form $E_n(p,q) : y^2 = x^3 + px + q \pmod n$. The constants p and q are chosen from the set Z_n with the condition: $4p^3 + 27q^2 \neq 0 \pmod n$. Furthermore, TA selects its own identity as ID_t and a secret master key, denoted as m_t which belongs to the set $Z_n^* = \{1, 2, \cdots, n-1\}$. Additionally, TA designates G on $E_n(p,q)$ as a base point with an order denoted as α, which is at least as large as n.
- The TA selects a collision-resistant one-way cryptographic hash function, denoted as $h(\cdot)$. In this case, to ensure sufficient security, Secure Hash Algorithm (SHA-256) is considered. Finally, TA keeps m_t as secret and publishes the parameters $\{E_n(p,q),\ G,\ h(\cdot)\}$ as public.

B. Registration Phase

In this section, the TA registers all the communicating entities for consumer electronics in IoT applications as follows.

1) Smart Devices Registration Phase: Before the deployment of SD, they are registered by the TA with private and public parameters.

- The TA selects an unique identity, ID_i, and a temporal identity, TID_i, for a smart device, SD_i. The TA then calculates a pseudo-identity, SID_i, as $SID_i = h(ID_i \,||RTS_i\, ||m_t\, ||ID_t)$, where RTS_i denotes a registration timestamp.
- The TA picks a secret signature key, $s_i \in Z_n^*$, and computes its corresponding public key, $Pub_{s_i} = s_i \cdot G$. TA then stores the parameters $\{TID_i,\ SID_i,\ s_i,\ Pub_{s_i}\}$ into SD_i's memory. TA sends $(TID_i,\ SID_i)$ to GN_j during its registration process. Subsequently, TA deletes s_i from its memory.

2) Gateway Node Registration Phase: A gateway node, GN_j, registers with the TA before the deployment of an IoT application using the following process.

- For a GN_j, the TA picks an unique identity ID_j, a temporal identity TID_j, and computes a pseudo-identity, SID_j, as $SID_j = h(ID_j \,||RTS_j\, ||m_t\, ||ID_t)$, where RTS_j denotes a registration timestamp. The TA selects a secret key $s_j \in Z_n^*$ for GN_j, and calculates its corresponding public key, $Pub_{s_j} = s_j \cdot G$.
- The TA selects a distinct t-degree symmetric bivariate polynomial of the form $f_(x,y) = \sum_{u=0}^{t}\sum_{v=0}^{t} a_{u,v}x^u y^v \pmod n$ over $GF(n)$, such that $t \gg$ number of GNs deployed in the IoT applications. Next, the TA derives the polynomial share $f_(SID_j, y)$ for GN_j and sends $(TID_j,\ SID_j)$ to cloud server CS_l.

Finally, The TA stores the registration parameters $\{ID_t, TID_j, SID_j, s_j, f_(SID_j, y)\}$ into GN_j's memory. The TA deletes s_j from its own memory.

3) Cloud Server Registration Phase: The TA registers the CS_l for smooth communication with gateway nodes (GNs) using the following process.

• The TA selects an unique identity for a CS_l as ID_l and a pseudo-identity, SID_l, as $SID_l = h(ID_l \,||ID_t \,||RTS_l \,||m_t)$, where RTS_l denotes a registration timestamp.

• The TA picks a secret key $s_l \in Z_n^*$ for CS_l, calculates the polynomial share $f_(SID_l, y)$. Finally, the TA stores $\{s_l, SID_l, f_(SID_l, y)\}$ into CS_l's memory and deletes s_l from its own database.

After the successful completion of the registration process, the communicating entities are deployed and can start communicating with each other using the pre-loaded registration parameters.

C. Access Control Phase

During this phase, we introduce two distinct access control mechanisms: a) access control for communication between a SD and a GN and b) access control for communication between a GN and the CS. Upon completion of the access control process, both communicating entities will have the capability to establish a confidential key for secure communication. The subsequent subsections elaborate on the specifics of these access control mechanisms. The communicated messages are sent via public channels.

1) Access control between SD and GN: In this section, a secure session key is established between a SD_i and a GN_j using the following steps.

Step ACCSG1: A smart device, SD_i, initiates the access control process by generating a random number, $x_1 \in Z_n^*$, and a timestamp, TS_1. SD_i calculates $a_1 = h(s_i|| \, x_1|| \, TS_1|| \, SID_i|| \, TID_i)$, $A = a_1 \cdot G$, and generates a signature Sig_i as $Sig_i = a_1 + h(TID_i \,||A \,||Pub_{s_i} \,||TS_1) * s_i \pmod n$. Next, SD_i constructs a message, msg_1, consisting of $\{TID_i, Sig_i, A, TS_1\}$ and sends it to GN_j.

Step ACCSG2: Upon receiving the message msg_1 from SD_i at a timestamp TS_1^*, GN_j checks the freshness of TS_1 by the condition $|TS_1 - TS_1^*| < \Delta T$, where ΔT is the "maximum transmission delay". If it is valid, GN_j checks the existence of TID_i. If it exists, then GN_j fetches SID_i corresponding to TID_i. Next, GN_j verifies the signature Sig_i as $Sig_i \cdot G = A + h(TID_i \,||A \,||Pub_{s_i} \,||TS_1) \cdot Pub_{s_i}$. If it is verified, GN_j selects a random number $x_2 \in Z_n^*$, and a new timestamp TS_2. GN_j calculates $b_1 = h(TS_2 \,||x_2 \,||s_j \,||SID_j)$, $B = b_1 \cdot G$, $DH_{ji} = b_1 \cdot A$, and generates a session key SK_{ji} as $SK_{ji} = h(DH_{ji} \,||SID_i \,||TS_1 \,||TS_2 \,||Sig_i)$. GN_j then generates a new temporal identity TID_i^n, and calculates $TID_i^* = TID_i^n \oplus h(SK_{ji} \,||TID_i \,||TS_2)$, along with a verifier $SKV = h(TID_i^* \,||SK_{ji} \,||B \,||TS_2 \,||TS_1)$. Subsequently, GN_j constructs a reply message msg_2 as $msg_2 = \{SKV, TS_2, B, TID_i^*\}$ and sends it to SD_i.

Step ACCSG3: Upon receiving the reply message msg_2 from GN_j at a timestamp TS_2^*, SD_i verifies its freshness using the condition $|TS_2 - TS_2^*| < \Delta T$. If it is valid, SD_i calculates $DH_{ij} = a_1 \cdot B$ and computes a session key $SK_{ij} = h(DH_{ij} \,||SID_i \,||TS_1 \,||TS_2 \,||Sig_i)$. Next, SD_i derives $TID_i^n = TID_i^* \oplus h(SK_{ij} \,||TID_i \,||TS_2)$ and computes $SKV' = h(TID_i^* \,||SK_{ij} \,||B \,||TS_2 \,||TS_1)$. SD_i then verifies $SKV' = SKV$. If the verification is successful, SD_i updates TID_i in its memory with the new value TID_i^n, and selects a new timestamp TS_3, and calculates $Ack = h(SK_{ij} \,||TID_i^n \,||TS_2 \,||TS_3)$. SD_i constructs an acknowledgment message msg_3 as $msg_3 = \{Ack, TS_3\}$ and sends it to GN_j.

Step ACCSG4: After receiving the message msg_3 from SD_i at a timestamp TS_3^*, GN_j checks its freshness by $|TS_3 - TS_3^*| < \Delta T$. If it is valid, GN_j calculates $Ack' = h(SK_{ji} \,||TID_i^n \,||TS_2 \,||TS_3)$ and verifies $Ack = Ack'$. If the verification is successful, GN_j believes that SD_i has successfully established the same session key $SK_{ji} (= SK_{ij})$, and updates TID_i^n in its memory.

At the end of this phase, SD_i and GN_j negotiate a secure session key $SK_{ji} (= SK_{ij})$ shows in Fig. 2.

Smart Device as SD_i	**Gateway Node** as GN_j
Stored: $\{(TID_i, SID_i), s_i, Pub_{s_i}\}$	$\{(TID_j, SID_j), s_j, ID_t, (TID_i, SID_i)\}$
Pick random $x_1 \in Z_n^*$, timestamp TS_1 Calculate $a_1 = h(s_i\|\| \, x_1\|\| \, TS_1\|\| \, SID_i\|\| \, TID_i)$, $A = a_1 \cdot G$. Generate $Sig_i = a_1 + h(TID_i$ $\|\|A \,\|\|Pub_{s_i} \,\|\|TS_1) * s_i \pmod n$ $msg_1 : \{TID_i, Sig_i, A, TS_1\}$ →	
	Check freshness of TS_1, validate TID_i, and fetch SID_i Verify $Sig_i \cdot G = A + h(TID_i \,\|\|A \,\|\|Pub_{s_i} \,\|\|TS_1) \cdot Pub_{s_i}$. If so, pick random $x_2 \in Z_n^*$, timestamp TS_2, and calculate $b_1 = h(TS_2 \,\|\|x_2 \,\|\|s_j \,\|\|SID_j)$, $B = b_1 \cdot G$ $DH_{ji} = b_1 \cdot A$, $SK_{ji} = h(DH_{ji} \,\|\|SID_i \,\|\|TS_1 \,\|\|TS_2 \,\|\|Sig_i)$ Pick a new TID_i^n. Calculate $TID_i^* = TID_i^n \oplus h(SK_{ji} \,\|\|TID_i \,\|\|TS_2)$ $SKV = h(TID_i^* \,\|\|SK_{ji} \,\|\|B \,\|\|TS_2 \,\|\|TS_1)$ $msg_2 : \{SKV, TS_2, B, TID_i^*\}$ ←
Check freashness of TS_2. If yes, calculate $DH_{ij} = a_1 \cdot B$, $SK_{ij} = h(DH_{ij} \,\|\|SID_i \,\|\|TS_1 \,\|\|TS_2 \,\|\|Sig_i)$ Calculate $TID_i^n = TID_i^* \oplus h(SK_{ij} \,\|\|TID_i \,\|\|TS_2)$, $SKV' = h(TID_i^* \,\|\|SK_{ij} \,\|\|B \,\|\|TS_2 \,\|\|TS_1)$ Verify $SKV' = SKV$. If yes, pick new timestamp TS_3 Update TID_i with new TID_i^n into its memory Calculate $Ack = h(SK_{ij} \,\|\|TID_i^n \,\|\|TS_2 \,\|\|TS_3)$ $msg_3 : \{Ack, TS_3\}$ →	
	Check TS_3. If yes, calculate $Ack' = h(SK_{ji} \,\|\|TID_i^n \,\|\|TS_2 \,\|\|TS_3)$ Verify $Ack = Ack'$, if yes, update TID_i^n into its memory

Fig. 2. Summary of access control phase between SD_i and GN_j.

2) Access control between GN and CS: Within this section, a secure session key between a GN_j and the CS_l is established to facilitate secure communication between the two entities. The subsequent steps outline the process involved in establishing this secure session key.

Step ACCGC1: A gateway node GN_j initiates the process to facilitate a session key by generating a random number $y_1 \in Z_n^*$, a timestamp T_1, and calculates $c_1 = h(y_1 \,||T_1 \,||SID_j \,||s_j \,||TID_i)$ and $C = c_1 \cdot G$. Next, GN_j computes $D = h(C \,||TID_j \,||SID_j \,||T_1)$ and constructs an access request message MG_1 as $MG_1 = \{C, T_1, TID_j, D\}$. GN_j then sends the message MG_1 to CS_l.

Step ACCGC2: Upon receiving the message MG_1 from GN_j at a timestamp T_1^*, CS_l verifies the timestamp using the condition $|T_1 - T_1^*| < \Delta T$. If it is valid, CS_l fetches SID_j by checking the existence of TID_j. Next, CS_l calculates $D' = h(C \,||TID_j \,||SID_j \,||T_1)$ and checks $D' = D$. If the condition is valid, CS_l selects a puzzle, PZ_1, which has a corresponding solution, PS_1. For example, this puzzle could involve the last five digits of SID_j or the last five digits of the previous session key, among other possibilities. Next, CS_l selects a random number r_1, a timestamp T_2, and

calculates $e_1 = h(r_1 \,||s_l\, ||SID_l\, ||T_2\, ||ID_t)$, $E = e_1 \cdot G$, and $DH_{lj} = e_1 \cdot C$. CS_l then generates a session key SK_{lj} as $SK_{lj} = h(DH_{lj}\, ||f(SID_l, SID_j)\, ||PS_1\, ||T_2\, ||T_1)$ and derives $PZ_1^* = PZ_1 \oplus h(SID_j\, ||TID_j\, ||T_2\, ||T_1)$. Next, CS_l computes $SID_l^* = SID_l \oplus h(SID_j\, ||PS_1\, ||PZ_1\, ||T_2\, ||T_1)$ and $TID_j^* = TID_j^n \oplus h(SK_{lj}\, ||SID_j\, ||TID_j\, ||T_2\, ||T_1)$ by picking a new TID_j^n. CS_l computes a session key verifier $SKV = h(SK_{lj}\, ||PZ_1^*\, ||E\, ||TID_j^*\, ||SID_l^*\, ||T_2)$ and creates a reply message MG_2 as $MG_2 = \{E,\, PZ_1^*,\, SID_l^*,\, SKV,\, T_2, TID_j^*\}$. Next, CS_l sends it to GN_j over a public channel.

Step ACCGC3: Upon receipt of the message MG_2 at a timestamp T_2^* from CS_l, GN_j validates it using the condition $|T_2 - T_2^*| < \Delta T$. If the validation is successful, GN_j derives $PZ_1 = PZ_1^* \oplus h(SID_j\, ||TID_j\, ||T_2\, ||T_1)$ and solves it, obtaining the solution PS_1'. Next, GN_j computes $SID_l = SID_l^* \oplus h(SID_j\, ||PS_1'\, ||PZ_1\, ||T_2\, ||T_1)$, $DH_{jl} = c_1 \cdot E$, and a session key $SK_{jl} = h(DH_{jl}\, ||f(SID_j, SID_l)\, ||PS_1'\, ||T_2\, ||T_1)$. GN_j then derives $TID_j^n = TID_j^* \oplus h(SK_{jl}\, ||SID_j\, ||TID_j\, ||T_2\, ||T_1)$ and calculates the verifier $SKV' = h(SK_{jl}\, ||PZ_1^*\, ||E\, ||TID_j^*\, ||SID_l^*\, ||T_2)$. Next, GN_j verifies if SKV' matches SKV. If the verification is successful, GN_j is confident that the puzzle PZ_1 is correctly solved and that the session key SK_{jl} is generated accurately. GN_j updates TID_j in its memory with the new value TID_j^n and picks a new timestamp T_3. Next, GN_j calculates the acknowledgment Ack as $Ack = h(SK_{jl}\, ||PS_1'\, ||PZ_1\, ||T_3\, ||T_2)$. GN_j generates a message $MG_3 = \{Ack,\, T_3\}$, and sends it to CS_l.

Step ACCGC4: Upon receiving the acknowledgment message MG_3 from GN_j at a timestamp T_3^*, CS_l verifies its freshness using the condition $|T_3 - T_3^*| < \Delta T$. If the condition is valid, CS_l derives $Ack' = h(SK_{lj}\, ||PS_1\, ||PZ_1\, ||T_3\, ||T_2)$ and checks whether Ack' matches Ack. If there is a match, CS_l is confident that GN_j is authentic and has correctly created the same session key $SK_{lj} (= SK_{jl})$. Subsequently, CS_l updates TID_j in its memory with the new value TID_j^n.

The cloud server CS_l and gateway node GN_j successfully establish the same secure session key $SK_{lj} (= SK_{jl})$ for secure communication. Figure 3 presents the details of this phase. A detailed flow diagram illustrating the entire access control process is presented in Fig. 4.

D. Secure Data Exchange Phase

In this section, we focus on the exchange of sensing data between SD_i and GN_j, as well as between GN_j and CS_l. The process begins with SD_i encrypting the data, denoted as $DATA$, using the negotiated session key SK_{ij} (also equal to SK_{ji}) with GN_j. Additionally, SD_i generates a current timestamp TS_{ij} and creates a signature (Sig_{data}) using its private key s_i on the $DATA$. Subsequently, SD_i constructs a message $msg_{ij} = \{E_{SK_{ij}}(DATA,\, TS_{ij}),\, Sig_{data},\, Pub_{s_i},\, TS_{ij}\}$ and transmits it to GN_j via a public channel.

Upon receiving the message msg_{ij} from SD_i, GN_j verifies its freshness. If the message is deemed valid, GN_j decrypts the ciphertext using its session key SK_{ji} and verifies the signature using the public key Pub_{s_i}. If the verification process is successful, GN_j encrypts the data using another session key SK_{jl} and selects a current timestamp TS_{jl}. Subsequently,

Gateway Node as GN_j	**Cloud Server** as CS_l
Stored: $\{(TID_j,\, SID_j),\, s_j,\, f(SID_j, y)\}$	$\{(TID_j,\, SID_j),\, s_i,\, SID_l,\, f(SID_l, y),\, ID_t\}$
Pick random $y_1 \in Z_n^*$, timestamp T_1 Calculate $c_1 = h(y_1\, ||T_1\, ||SID_j\, ||s_j\, ||TID_j)$, $C = c_1 \cdot G$.	
Calculate $D = h(C\, ||TID_j\, ||SID_j\, ||T_1)$ $MG_1 : \{C, T_1, TID_j, D\}$ $\longrightarrow$	Check $\lvert T_1 - T_1^* \rvert < \Delta T$. If so, validate TID_j, and fetch SID_j Calculate $D' = h(C\, ||TID_j\, ||SID_j\, ||T_1)$, and check if $D' = D$? Pick a puzzle PZ_1 (corresponding solution is PS_1), random r_1, and timestamp T_2. Calculate $e_1 = h(r_1\, ||s_l\, ||SID_l\, ||T_2\, ||ID_t)$ $E = e_1 \cdot G$, $DH_{lj} = e_1 \cdot C$, and session key $SK_{lj} = h(DH_{lj}\, ||f(SID_l, SID_j)\, ||PS_1\, ||T_2\, ||T_1)$ $PZ_1^* = PZ_1 \oplus h(SID_j\, ||TID_j\, ||T_2\, ||T_1)$ $SID_l^* = SID_l \oplus h(SID_j\, ||PS_1\, ||PZ_1\, ||T_2\, ||T_1)$ Select TID_j^n, and calculate $TID_j^* = TID_j^n \oplus h(SK_{lj}\, ||SID_j\, ||TID_j\, ||T_2\, ||T_1)$ $SKV = h(SK_{lj}\, ||PZ_1^*\, ||E\, ||TID_j^*\, ||SID_l^*\, ||T_2)$ $MG_2 : \{E, PZ_1^*, SID_l^*, SKV, T_2, TID_j^*\}$ $\longleftarrow$
Check $\lvert T_2 - T_2^* \rvert < \Delta T$?, if yes, $PZ_1 = PZ_1^* \oplus h(SID_j\, ||TID_j\, ||T_2\, ||T_1)$ Solve PZ_1, lets solution is PS_1', and derive $SID_l = SID_l^* \oplus h(SID_j\, ||PS_1'\, ||PZ_1\, ||T_2\, ||T_1)$ Calculate $DH_{jl} = c_1 \cdot E$, and session key $SK_{jl} = h(DH_{jl}\, ||f(SID_j, SID_l)\, ||PS_1'\, ||T_2\, ||T_1)$ $TID_j^n = TID_j^* \oplus h(SK_{jl}\, ||SID_j\, ||TID_j\, ||T_2\, ||T_1)$ $SKV' = h(SK_{jl}\, ||PZ_1^*\, ||E\, ||TID_j^*\, ||SID_l^*\, ||T_2)$ Check $SKV' = SKV$, if yes, GN_j believed that it correctly solve PZ_1 and generate SK_{jl}	
Update TID_j with the new TID_j^n into its memory and pick new timestamp T_3 Calculate $Ack = h(SK_{jl}\, ||PS_1'\, ||PZ_1\, ||T_3\, ||T_2)$ $MG_3 : \{Ack, T_3\}$ $\longrightarrow$	Check $\lvert T_3 - T_3^* \rvert < \Delta T$?, if yes, derive $Ack' = h(SK_{lj}\, ||PS_1\, ||PZ_1\, ||T_3\, ||T_2)$ Verify if $Ack' = Ack$? If yes, CS_l believed that GN_j is authentic and generate the session key $SK_{lj} (= SK_{jl})$ correctly Update TID_j with the new TID_j^n into its memory

Fig. 3. Summary of access control phase between GN_j and CS_l.

GN_j constructs a message $msg_{jl} = \{E_{SK_{jl}}(DATA,\, TS_{jl}),\, Sig_{data},\, Pub_{s_i},\, TS_{jl}\}$ and forwards it to CS_l.

Similarly, upon receiving the message msg_{jl}, CS_l checks its freshness, performs decryption, and verifies the signature. If all checks pass successfully, CS_l utilizes the data for machine learning (ML) purposes. Since each data is accompanied by its own signature, only authentic data is used for ML operations.

E. New Smart Device (SD_k) Addition Phase

Before the deployment of a new smart device (SD_k), a TA registers it with private and public parameters.

- The TA picks a new unique identity, ID_k, and a new temporal identity, TID_k, for a smart device, SD_k. The TA then calculates a pseudo-identity, SID_k, as $SID_k = h(ID_k\, ||RTS_k\, ||m_t\, ||ID_t)$, where RTS_k denotes a registration timestamp.
- The TA selects a new secret signature key, $s_k \in Z_n^*$, and computes its corresponding public key, $Pub_{s_k} = s_k \cdot G$. The TA then stores the parameters $\{TID_k,\, SID_k,\, s_k,\, Pub_{s_k}\}$ into SD_k's memory. The TA sends $(TID_k,\, SID_k)$ to the associated gateway node GN_j during its addition process. Subsequently, the TA deletes s_k from its database.

V. Security Analysis

In this section, we provide an informal (heuristic) security analysis to showcase the robustness and resilience of the proposed RMACE-IoT scheme against a range of active and passive attacks.

A. Informal Security Analysis

In Propositions 1–7, we show the proposed scheme (RMACE-IoT) is resilient against various attacks. The detailed proofs of these propositions are provided in the supplementary material.

Proposition 1. *RMACE-IoT is resilient against replay attack.*

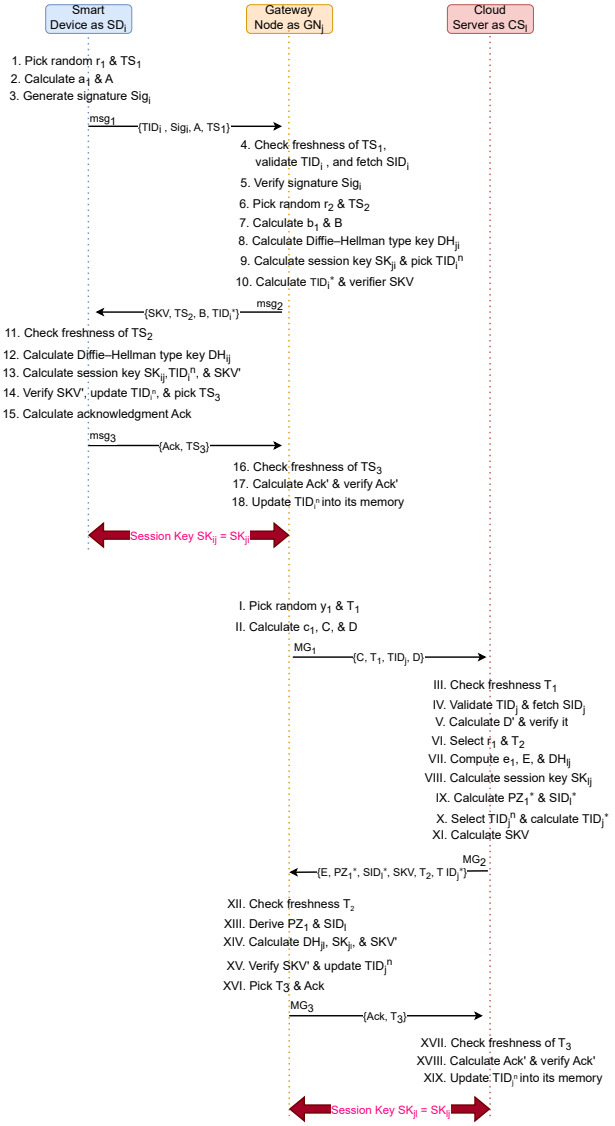

Fig. 4. A flow diagram of the entire access control process.

Proposition 2. *RMACE-IoT is secure against Man-in-the-Middle (MiTM) attack.*

Proposition 3. *RMACE-IoT is resilient against SD impersonation attack.*

Proposition 4. *RMACE-IoT is resilient against GN impersonation attack.*

Proposition 5. *RMACE-IoT is robust against privileged-insider attack.*

Proposition 6. *RMACE-IoT is resilient against physical device capture attack.*

Proposition 7. *RMACE-IoT is resilient against Ephemeral Secret Leakage (ESL) attack.*

Proposition 8. *RMACE-IoT preserves both anonymity and untraceability properties.*

B. Formal Security Verification under Scyther Tool

Within this section, we present a formal security verification of the proposed RMACE-IoT scheme, employing the widely adopted Scyther tool. Scyther is an automatic verification tool specifically designed for security protocols [45], [46]. It offers guarantees of termination and allows for proving correctness across an unlimited number of sessions. Additionally, it can generate a proof tree if desired. Unlike other tools, Scyther's graphical user interface provides an analysis based on classes of protocol behavior or attacks, rather than solely relying on individual attack traces. By utilizing the Dolev-Yao (DY) threat model [42], Scyther operates under a widely employed threat model for the analysis of security properties in cryptographic protocols. In the DY model, an assumed adversary possesses unlimited computational power and can intercept, modify, and generate messages at will. This threat model allows Scyther to effectively analyze and identify various security vulnerabilities in protocols, considering realistic adversary assumptions.

The Scyther tool utilizes its proprietary specification language, referred to as "spdl", to delineate the implementation of a security protocol. This language empowers users to define the distinct roles engaged in the protocol, wherein communication occurs. Moreover, it offers a range of functions, such as "sent" and "recv", to facilitate the transmission and reception of messages between these roles. These functions are denoted in the format of "sent_" or "recv_". For a comprehensive understanding, please refer to the Scyther manual [46].

Scyther results : verify

Claim				Status		Commer
RMACE_IoT	SD	RMACE_IoT,SD1	Alive	Ok	Verified	No attacks.
		RMACE_IoT,SD2	Nisynch	Ok	Verified	No attacks.
		RMACE_IoT,SD3	Niagree	Ok	Verified	No attacks.
		RMACE_IoT,SD4	Commit GN,skij	Ok	Verified	No attacks.
		RMACE_IoT,SD5	Secret si	Ok	Verified	No attacks.
		RMACE_IoT,SD6	Secret x1	Ok	Verified	No attacks.
		RMACE_IoT,SD7	Secret sidi	Ok	Verified	No attacks.
	GN	RMACE_IoT,GN1	Alive	Ok	Verified	No attacks.
		RMACE_IoT,GN2	Nisynch	Ok	Verified	No attacks.
		RMACE_IoT,GN3	Niagree	Ok	Verified	No attacks.
		RMACE_IoT,GN4	Commit SD,skji	Ok	Verified	No attacks.
		RMACE_IoT,GN5	Secret sj	Ok	Verified	No attacks.
		RMACE_IoT,GN6	Secret x2	Ok	Verified	No attacks.
		RMACE_IoT,GN7	Secret sidj	Ok	Verified	No attacks.

Done.

Fig. 5. Simulation results using Scyther tool.

The findings of this evaluation, performed employing the Scyther specification language, are showcased in Fig. 5. The simulation examines case 1, specifically the access control between SD_i and GN_j, with the roles defined as follows: a) SD, and b) GN. In the proposed scheme, random values are generated using the "fresh" or "var" keyword, and the

"match( )" function is utilized to assign values provided by the Scyther tool. The experiment was conducted under the following environmental parameters: verification parameters set at 5 (maximum number of runs), matching type designated as "typed matching," and advanced parameters configured with 10 (maximum number of patterns per claim). The searching pruning technique employed was "Find best attack."

Scyther leverages the concept of "claim" events to articulate security requisites. These claims can manifest in various forms, such as the designation of "secret," denoting that the value specified in the claim event is treated as confidential and verified within the bounds of the security model, considering potential adversaries. Moreover, Scyther employs Nisynch to denote non-injective synchronization and Niagree to represent non-injective agreement. Notably, the outcomes portrayed in Fig. 5 affirm that Scyther has identified no vulnerabilities or deficiencies in the proposed scheme.

VI. Experimental Setup and Results

In this section, we concentrate on performing testbed experiments using Raspberry Pi devices, which are considered as smart devices, along with server platforms such as GN or CS.

A. Testbed Experiment of the Cryptographic Primitives Using Raspberry PI

We conducted an analysis of the execution time for various cryptographic primitives using the widely recognized cryptography standard library, known as "cryptography 37.0.2". This library offers Python programmers access to a comprehensive set of cryptographic recipes and primitives. Let T_{eca}/T_{ecm}, T_{senc}/T_{sdec}, T_h, T_{mtp}, T_m, T_{me}, and T_{bp} denote the execution time for different operations: "elliptic curve point addition/multiplication", "Advanced Encryption Standard (AES-128) encryption/decryption", "one-way cryptographic hash function (specifically SHA-256)", "message to elliptic curve point function", "modular multiplications", "modular exponentiation", and "bilinear pairing operation", respectively. For the elliptic curve point operations, we utilized a non-singular elliptic curve called secp256r1, which follows the form $y^2 = x^3 + px + q \pmod n$ (for additional details, refer to RFC5480). To measure the timings for bilinear pairing operations, we employed the Tate bilinear pairing on a supersingular elliptic curve represented by $y^2 = x^3 - x + 1$ over the Galois field (GF) [47] and the T_{mtp} for message to point (MTP) function has been calculated using the Koblitz's method [48]. We conducted a total of 500 iterations for all experiments, and subsequently calculated the maximum, minimum, and average costs (in milliseconds) for each cryptographic primitive. The comparative analysis is based on the average time recorded for each primitive. The experimental results are showcased in Table II. The details of the testbed experiments, and results for Platform 1 (server side) and Platform 2 (device side using Raspberry Pi) are provided as follows.

Platform 1. In this platform, our attention is directed towards a dedicated computing platform tailored for server-based computations, which possesses the following characteristics: "Ubuntu 22.04 LTS, with memory: 16 GB, processor: Intel® Core™ i7-9750H CPU @ 2.60GHz processor with 6 cores; 12 threads, OS type: 64-bit and disk type: SSD 256 GB". We have provided here the specification, because the experimental results differ for other specifications under different configurations.

Platform 2. In this particular platform, we have assigned the user equipment to operate as a device, which is considered as a Raspberry Pi setup, employing the subsequent configuration: "Raspberry PI 4 Model B, with CPU: 64-bit, Processor: 1.4 GHz Quad-core, 4 cores, Memory (RAM): 1GB, and OS: Ubuntu 20.04 LTS, 64-bit". Note that we have provided here the specification of Raspberry Pi 4, because the experimental results differ for other speficications under different configurations.

TABLE II
Execution time (in milliseconds) for cryptographic primitives

Primitive	Average time (ms) for server	Average time (ms) for Raspberry PI 4
T_h	0.0424	0.3187
T_{senc}	0.0173	0.0926
T_{sdec}	0.0163	0.0945
T_{ecm}	0.1590	1.0712
T_{eca}	0.0229	0.1509
T_{bp}	746.0845	3938.8026
T_{mtp}	0.6627	7.7039
T_m	0.001	0.008
T_{me}	0.339	2.249

B. Testbed Implementation of Proposed Scheme

We implemented the proposed RMACE-IoT scheme using Python and the Cryptography Library. The operating system we used was Ubuntu 22.04 LTS. The hardware specifications of the machine we utilized were as follows: Processor: Intel Core i7-9750H CPU @ 2.60 GHz, 16 GB RAM, and a 256 GB SSD. Furthermore, we employed the HTTP protocol to establish a communication channel between the devices, including the smart device, gateway node, and cloud server. The detail of this experiment is provided in the supplementary material.

C. Machine Learning Setup and Results

To investigate the influence of adversarial attacks, specifically data poisoning attacks, on the machine learning (ML) training process, we utilized a dataset available at https://aka.ms/diabetes-data and trained a Random Forest Classifier on it. It is worth noticing that during the secure data exchange phase discussed in Section IV-D, the collected data from the smart devices SD_i through the gateway nodes GN_j is stored securely to the cloud servers CS_l. So, in between the communications among SD_i, GN_j and CS_l, the data corruption by an attacker is not possible. In this case, it is assumed a 0% attack scenario, where the adversary cannot successfully launch data poisoning attack due to the presence of unique data signatures in the proposed scheme. Thus, in the absense of the proposed access control mechanism, there is always data poisoning attack possibility by the adversary.

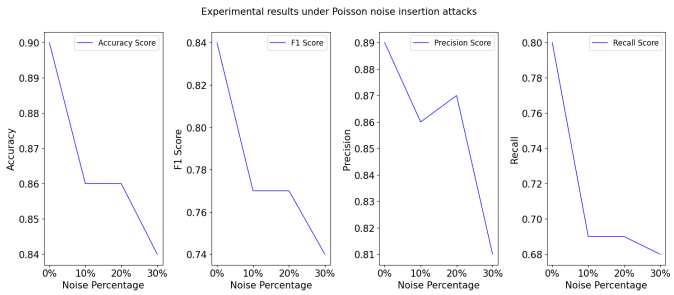

Fig. 6. Impact of adversarial attacks on machine learning (ML) training.

In our machine learning experimental scenario, we examine the impact of adversarial attacks by initiating data poisoning attacks on the training dataset. These attacks involve introducing noise at various percentages during training the ML model. To generate the noisy data, we employ Poisson noise, a commonly used method. Let us assume that original dataset size is denoted as x. The adversary randomly selects a percentage, y, of the data from this dataset and poisons it by inserting noisy data. Consequently, a fraction of the original data, equivalent to $x/y\%$, is replaced by noisy data. The ML model is subsequently trained using this noisy dataset. We conducted a simulation of a data poisoning attack using a Diabetes dataset. A Random Forest Classifier was trained on this dataset, which was split in a $7:3$ ratio, with 70% used for training and 30% for testing.

The results of our experiments are presented in Fig. 6, which illustrates the performance of the machine learning model trained on the original data compared to the model trained with the poisoned data resulting from the noise attack. In our experiments, the ML model exhibits robust resilience against adversarial attacks during the training process. This is particularly evident in the case of the 0% attack scenario, where the adversary cannot successfully launch an attack due to the presence of unique data signatures. Here, the data signatures represent the authentic signatures of sensor data. The data, along with their corresponding signatures, is uploaded to a cloud server for the utilization of ML models. Before incorporating these real data into the ML model's training process, the authenticity of the data is verified by checking their corresponding signatures. Only data with successfully verified signatures is accepted for the ML process. This is crucial because the integrity of the data depends on their signatures; thus, verified signatures ensure the preservation of the data's integrity. Consequently, signature verification during the training process ensures the integrity of the dataset, resulting in notably high accuracy in the output of the ML process, as the dataset is authentic. However, when we permit the adversary to launch attacks at rates of 10%, 20%, and 30%, the model's accuracy noticeably decreases (see Fig. 6).

We employed Google Colaboratory to train the ML model. Fig. 6 displays the results, illustrating the resilience of the ML model (at 0%) and the impact of adversarial attacks on metrics, such as Accuracy, F1 Score, Precision, and Recall. As depicted in this figure, an increase in the noise percentage correlates with a decrease in all of these metrics.

VII. Comparative Analysis

Within this section, we assess and contrast the performance of the proposed scheme (RMACE-IoT) with other pertinent competitive schemes, including those proposed by Luo et al. [16], Ali and Pal [21], Ma et al. [23], Ever [25], Ren et al. [26], Fan et al. [27], Wang et al. [28], Yuanbing et al. [30], Huang et al. [49], Wang et al. [32], Meher et al. [33], Mahmood et al. [34], Zhang et al. [35], [50], [37], Aldosary et al. [40] and Fan et al. [41].

TABLE III
Comparative analysis on communication costs

Scheme	No. of messages	Total cost (in bits)
Luo et al. [16]	2	3040
Ali and Pal [21]	4	4608
Ma et al. [23]	4	5664
Ever [25]	6	5344
Ren et al. [26]	4	3520
Fan et al. [27]	6	6016
Wang et al. [28]	3	1568
Yuanbing et al. [30]	4	4608
Huang et al. [49]	2	1632
Wang et al. [32]	6	4736
Meher et al. [33]	3	1536
Mahmood et al. [34]	2	3008
Chaudhry et al. [50]	2	2368
Mahmood et al. [37]	2	2176
Zhang et al. [35]	4	9984
Aldosary et al. [40]	2	1920
Fan et al. [41]	5	1600
RMACE-IoT (Case 1)	3	1824
RMACE-IoT (Case 2)	3	2432

A. Communication Costs Analysis

To compute the communication expense, we took into account the access control phases between SD and GN, as well as between GN and CS, as elucidated in Sec. IV-C. In this examination, we made certain assumptions regarding the dimensions of various data constituents, namely: identity or temporal identity, timestamp, random nonce, hash digest (utilizing the SHA-256 hashing algorithm), puzzle, and elliptic curve point (such as A and B). We assumed these components to be 160 bits, 32 bits, 160 bits, 256 bits, 32 bits, and 320 bits, respectively. In Wi-Fi networks commonly used for IoT, channel bandwidths can range from 20 MHz to 160 MHz, with 20 MHz and 40 MHz being more typical for many IoT applications, whereas NB-IoT has a channel bandwidth of 200 kHz but occupies only 180 kHz.

In the proposed RMACE-IoT, two cases are considered, such as case 1: access control between SD_i and GN_j and case 2: access control between GN_j and CS_l. In case 1, three messages are exchanged over a public channel: $msg_1 = \{TID_i, Sig_i, A, TS_1\}$, $msg_2 = \{SKV, TS_2, B, TID_i^*\}$, and $msg_3 = \{Ack, TS_3\}$. These messages require 672 bits, 864 bits, and 288 bits, respectively, resulting in a total of 1824 bits. In case 2, three messages are exchanged over a public channel: $MG_1 = \{C, T_1, TID_j, D\}$, $MG_2 = \{E, PZ_1^*, SID_l^*, SKV, T_2, TID_j^*\}$, and $MG_3 = \{Ack, T_3\}$. These messages need 768 bits, 1376 bits, and 288 bits, respectively, resulting in a total of 2432 bits.

Upon reviewing Table III, it is evident that our proposed RMACE-IoT scheme demonstrates significantly reduced communication expenses in comparison to other relevant schemes, including those put forth by Ali and Pal [21], Ma et al. [23], Ever [25], Fan et al. [27], Yuanbing et al. [30], Wang et al. [32], Zhang et al. [35], [50], [37], and Aldosary et al. [40]. As depicted in Fig. 7 Case 1 (A) and Case 2 (A), the analysis reveals that as the number of smart devices/GNs or servers increases, our RMACE-IoT scheme necessitates lower communication costs (in kilo-bits) for both Case 1 and Case 2 compared to the alternative approaches.

B. Computation Costs Analysis

To determine the computation cost in the proposed RMACE-IoT scheme, two cases are considered, such as case 1: access control between SD_i and GN_j and case 2: access control between GN_j and CS_l. In case 1, a smart device requires the computation cost of $6T_h + 2T_{ecm} \approx 4.0546$ ms and GN_j requires the computation cost of $5T_h + 4T_{ecm} + T_{eca} \approx 0.8709$ ms. In case 2, GN_j requires the computation cost of $8T_h + 2T_{ecm} + T_{poly} \approx (0.6527 + 0.001t)$ ms and CS_l requires the computation cost of $8T_h + 2T_{ecm} + T_{poly} \approx (0.6527 + 0.001t)$ ms. Therefore, both of them together requires the computation costs of $16T_h + 4T_{ecm} + 2T_{poly} \approx (1.3054 + 0.002t)$ ms ≈ 1.5054 ms for $t = 100$, where t is the degree of a bivariate polynomial over finite field $GF(n)$ in the proposed RMACE-IoT scheme.

It is worth noting that the evaluation of a univariate polynomial of degree t necessitates t modular multiplications (T_m) and t modular additions (T_a), that is, $T_{poly} = tT_m + tT_a$. In other words, the total computation time for evaluating the polynomial, denoted as T_{poly}, can be approximated as tT_m by disregarding the modular addition operation. Here, T_m and T_a represent the time required for a single modular multiplication and modular addition, respectively. Table IV presents a comparison of computation costs between the proposed RMACE-IoT scheme and other existing schemes. The Table IV indicates that our scheme exhibits lower computation costs compared to Luo et al. [16], Ali and Pal [21], Ma et al. [23], Ever [25], Ren et al. [26], Wang et al. [28], Yuanbing et al. [30], Huang et al. [49], Wang et al. [32], Mahmood et al. [34], Zhang et al. [35], [50], [37], and Aldosary et al. [40] on the smart device side. Additionally, our scheme demonstrates reduced computation costs compared to Luo et al. [16], Ali and Pal [21], Ma et al. [23], Ever [25], Wang et al. [28] and Wang et al. [32] for case 1 on the server's side. As illustrated in Fig. 7 Case 1 (B) and Case 2 (B), the results demonstrates that as the number of smart devices/GNs or servers increases, our RMACE-IoT scheme requires lower computation costs in comparison to the alternative approaches.

C. Functionality and Security (FS) Attributes

Table V illustrates that the proposed RMACE-IoT scheme satisfies all the essential security and functionality requirements, providing a robust access control solution for consumer electronics in IoT applications. In contrast, other existing solutions do not completely meet the desired level of security.

TABLE IV
COMPARATIVE ANALYSIS ON COMPUTATION COSTS

Scheme	Smart device/IoT device	GN/Server/CS
Luo et al. [16]	$T_{bp} + T_h$ ≈ 3939.1213 ms	$3T_{ecm} + 3T_{bp} + 3T_h + T_{eca} + T_{me}$ ≈ 2239.2196 ms
Ali and Pal [21]	$4T_h + 6T_{ecm} + 4T_{eca}$ ≈ 8.3056 ms	$8T_h + 8T_{ecm} + 6T_{eca} + 4T_{senc}/T_{sdec}$ ≈ 1.8158 ms
Ma et al. [23]	$4T_h + 3T_{ecm}$ ≈ 4.4884 ms	$13T_h + 12T_{ecm}$ ≈ 2.4592 ms
Ever [25]	$9T_h + 2T_{bp} + 2T_{mtp} + 3T_{ecm}$ ≈ 7899.0949 ms	$6T_h + 3T_{bp} + 2T_{mtp} + 3T_{ecm}$ ≈ 2240.3103 ms
Ren et al. [26]	$4T_h + 3T_{ecm} + T_{eca} \approx 4.6393$ ms	$3T_h + 5T_{ecm} + 2T_{eca} \approx 0.968$ ms
Fan et al. [27]	$3T_h + 2T_{ecm} + 4T_{senc}/T_{sdec}$ ≈ 3.4729 ms	$T_h + T_{ecm} + 10T_{senc}/T_{sdec}$ ≈ 0.3694 ms
Wang et al. [28]	$5T_h + 4T_{ecm} + 3T_{eca}$ ≈ 6.331 ms	$5T_h + 6T_{ecm} + 3T_{eca}$ ≈ 1.2347 ms
Yuanbing et al. [30]	$15T_h + 4T_{ecm}$ ≈ 9.0653 ms	$11T_h$ ≈ 0.4664 ms
Huang et al. [49]	$5T_h + 5T_{ecm} + T_{sdec}$ ≈ 7.044 ms	$5T_h + 4T_{ecm} + T_{sdec}$ ≈ 0.8643 ms
Wang et al. [32]	$9T_h + 5T_{ecm}$ ≈ 8.2243 ms	$18T_h + T_{ecm}$ ≈ 0.9222 ms
Meher et al. [33]	$3T_{eca} + 2T_{ecm}$ ≈ 2.5951 ms	$3T_{eca} + 3T_{ecm} + T_{senc}$ ≈ 0.563 ms
Mahmood et al. [34]	$6T_h + T_{eca} + 3T_{ecm}$ ≈ 5.2767 ms	$7T_h + 2T_{eca} + 2T_{ecm} + 2T_{senc}/T_{sdec} \approx 0.6942$ ms
Zhang et al. [35]	$12T_h + 3T_{ecm} + T_{senc} + T_{sdec} \approx 7.2251$ ms	$7T_h + T_{ecm} + T_{senc} + T_{sdec} \approx 0.4894$ ms
Mahmood et al. [37]	$4T_h + 10T_{ecm} + 4T_{eca} \approx 12.5904$ ms	–
Chaudhry et al. [50]	$4T_h + 3T_{ecm} + T_{senc} + T_{sdec} \approx 4.6755$ ms	$6T_h + 4T_{ecm} + 2T_{senc} + 2T_{sdec} \approx 0.9576$ ms
Aldosary et al. [40]	$6T_h + 2T_{ecm} + 2T_{senc}/T_{sdec} \approx 4.2411$ ms	$5T_h + T_{ecm} + 2T_{senc}/T_{sdec} \approx 0.4046$ ms
Fan et al. [41]	$5T_h + 2T_{ecm} + 2T_{senc}/T_{sdec} \approx 3.923$ ms	$3T_h + 2T_{ecm} + 3T_{senc}/T_{sdec} \approx 0.4951$ ms
RMACE-IoT (Case 1)	$6T_h + 2T_{ecm}$ ≈ 4.0546 ms	$5T_h + 4T_{ecm} + T_{eca} \approx 0.8709$ ms
RMACE-IoT (Case 2)	–	$16T_h + 4T_{ecm} + 2T_{poly}$ $\approx (1.3054 + 0.002t)$ ms ≈ 1.5054

Note: t: degree of a bivariate polynomial over finite field $GF(n)$ in the proposed RMACE-IoT scheme. Here, $t = 100$.

TABLE V
COMPARATIVE ANALYSIS ON VARIOUS FS ATTRIBUTES

Attribute	[37]	[50]	[35]	[16]	[21]	[23]	[25]	[26]	[27]	[33]	[34]	[40]	[41]	Ours
$FUSA_1$	✓	✓	×	✓	×	✓	✓	×	×	✓	✓	✓	✓	✓
$FUSA_2$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$FUSA_3$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$FUSA_4$	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
$FUSA_5$	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$FUSA_6$	×	✓	✓	×	✓	×	×	✓	✓	×	✓	✓	×	✓
$FUSA_7$	×	✓	×	✓	✓	×	×	✓	×	✓	✓	✓	✓	✓
$FUSA_8$	✓	✓	✓	✓	✓	×	✓	✓	×	✓	✓	✓	✓	✓
$FUSA_9$	×	×	✓	×	×	×	×	×	✓	×	×	×	×	✓
$FUSA_{10}$	✓	×	×	✓	✓	×	×	✓	×	✓	×	✓	×	✓

$FUSA_1$: "replay attack"; $FUSA_2$: "man-in-the-middle attack"; $FUSA_3$: "key negotiation"; $FUSA_4$: "device impersonation attack"; $FUSA_5$: "resilience against device physical capture attack"; $FUSA_6$: "ESL attack under the CK-adversary model"; $FUSA_7$: "anonymity leakage"; $FUSA_8$: "privileged-insider attack"; $FUSA_9$: "support new device addition phase", $FUSA_{10}$: Formal security verification using AVISPA/Scyther/ProVerif; ✓: "a scheme is secure or it supports an attribute"; ×: "a scheme is insecure or it does not support an attribute".

For instance, Luo et al. [16] lacks support for resilience against device physical capture attacks and ESL attacks under the CK-adversary model. Similarly, Fan et al. [27] is vulnerable to replay attacks, ESL attacks under the CK-adversary model, and anonymity leakage.

VIII. CONCLUSION AND FUTURE WORKS

We have developed RMACE-IoT, a resilient machine learning-based access control system for consumer electronics in the IoT environment. This scheme ensures secure data transfer to the cloud server, where the data is utilized for

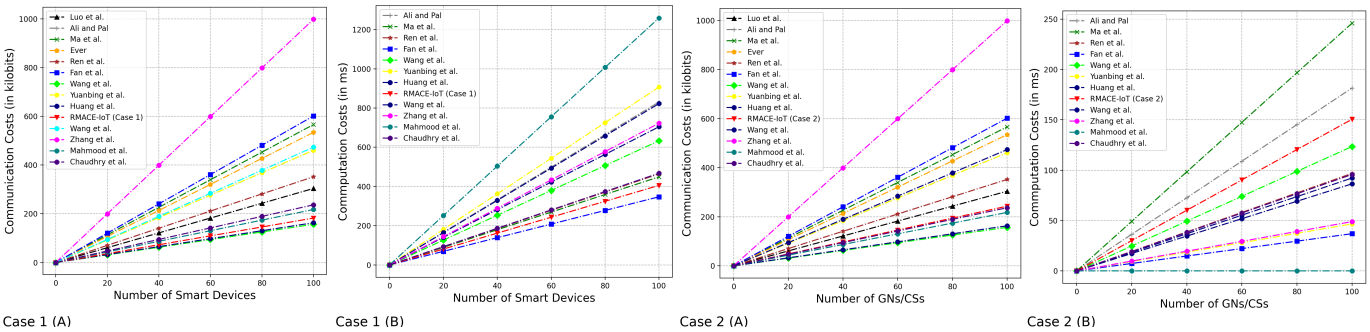

Fig. 7. Comparison of Case 1 and Case 2.

machine learning purposes. Our proposed system effectively defends against adversarial attacks during both the training and testing processes of the machine learning model. Each piece of data is assigned a unique signature, and only authenticated data is used in the machine learning process after signature verification. Real-time test-bed experiments were conducted to demonstrate the practical feasibility of our proposed access control scheme. Furthermore, a security analysis confirms the resilience of the RMACE-IoT scheme against various active and passive attacks, whereas the past consumer research works of Meher et al. [33] and Mahmood et al. [34] cannot resist such potential attacks. Finally, performance evaluation results illustrate the efficiency of our system compared to other competing security schemes as we as past CE research works.

In future work, we plan to explore several promising directions for extending the RMACE-IoT scheme. One key area is the adaptation of RMACE-IoT for a broader range of IoT applications beyond consumer electronics, including critical sectors such as healthcare, smart cities, and industrial IoT. These domains present unique challenges in terms of scalability, interoperability, and security, and RMACE-IoT could be further optimized to meet these specific needs. Additionally, we aim to investigate the integration of RMACE-IoT with emerging technologies like blockchain to introduce an additional layer of security and transparency. By leveraging blockchain's decentralized and immutable nature, we could enhance access control by ensuring tamper-proof records of user interactions and access attempts, thus reinforcing the resilience against adversarial attacks. Furthermore, we plan to explore the potential of machine learning techniques in dynamic and real-time adaptation of access control policies, enabling RMACE-IoT to more effectively respond to evolving security threats.

REFERENCES

[1] C. K. Wu, C.-T. Cheng, Y. Uwate, G. Chen, S. Mumtaz, and K. F. Tsang, "State-of-the-Art and Research Opportunities for Next-Generation Consumer Electronics," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 937–948, 2023.

[2] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2020.

[3] D. Javeed, M. S. Saeed, I. Ahmad, P. Kumar, A. Jolfaei, and M. Tahir, "An Intelligent Intrusion Detection System for Smart Consumer Electronics Network," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 906–913, 2023.

[4] C. Koverman, "Next-Generation Connected Support in the Age of IoT: It's time to get proactive about customer support," *IEEE Consumer Electronics Magazine*, vol. 5, no. 1, pp. 69–73, 2016.

[5] S. J. Nawaz, S. K. Sharma, M. N. Patwary, and M. Asaduzzaman, "Next-Generation Consumer Electronics for 6G Wireless Era," *IEEE Access*, vol. 9, pp. 143 198–143 211, 2021.

[6] S. K. Behera, P. Kumar, D. P. Dogra, and P. P. Roy, "A Robust Biometric Authentication System for Handheld Electronic Devices by Intelligently Combining 3D Finger Motions and Cerebral Responses," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 1, pp. 58–67, 2021.

[7] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021.

[8] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge Computing in Industrial Internet of Things: Architecture, Advances and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[9] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain," *ACM Comput. Surv.*, vol. 54, no. 5, 2021.

[10] J. Chen, X. Gao, R. Deng, Y. He, C. Fang, and P. Cheng, "Generating Adversarial Examples Against Machine Learning-Based Intrusion Detector in Industrial Control Systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1810–1825, 2022.

[11] N. Martins, J. M. Cruz, T. Cruz, and P. Henriques Abreu, "Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review," *IEEE Access*, vol. 8, pp. 35 403–35 419, 2020.

[12] A. Singh and B. Sikdar, "Adversarial Attack and Defence Strategies for Deep-Learning-Based IoT Device Classification Techniques," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2602–2613, 2022.

[13] Y. Zhou, M. Kantarcioglu, and B. Xi, "A survey of game theoretic approach for adversarial machine learning," *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 3, p. e1259, 2019.

[14] F. O. Olowononi, D. B. Rawat, and C. Liu, "Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for cps," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 524–552, 2021.

[15] N. Taimoor and S. Rehman, "Reliable and Resilient AI and IoT-Based Personalised Healthcare Services: A Survey," *IEEE Access*, vol. 10, pp. 535–563, 2022.

[16] M. Luo, Y. Luo, Y. Wan, and Z. Wang, "Secure and efficient access control scheme for wireless sensor networks in the cross-domain context of the IoT," *Security and Communication Networks*, vol. 2018, pp. 1–10, 2018.

[17] S. A. Chaudhry, K. Yahya, F. Al-Turjman, and M.-H. Yang, "A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems," *IEEE Access*, vol. 8, pp. 139 244–139 254, 2020.

[18] M. Masud, G. S. Gaba, K. Choudhary, M. S. Hossain, M. F. Alhamid, and G. Muhammad, "Lightweight and Anonymity-Preserving User Authentication Scheme for IoT-Based Healthcare," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2649–2656, 2022.

[19] Q. Li, Q. Zhang, H. Huang, W. Zhang, W. Chen, and H. Wang, "Secure, Efficient, and Weighted Access Control for Cloud-Assisted Industrial IoT," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 16 917–16 927, 2022.

[20] S. Xiong, Q. Ni, L. Wang, and Q. Wang, "SEM-ACSIT: Secure and Efficient Multiauthority Access Control for IoT Cloud Storage," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2914–2927, 2020.

[21] R. Ali and A. K. Pal, "An efficient three factor-based authentication scheme in multiserver environment using ECC," *International Journal of Communication Systems*, vol. 31, no. 4, p. e3484, 2018.

[22] M. Luo, A. Sun, D. He, and X. Li, "An efficient and secure 3-factor user-authentication protocol for multiserver environment," *International Journal of Communication Systems*, vol. 31, no. 14, p. e3734, 2018.

[23] M. Ma, D. He, H. Wang, N. Kumar, and K.-K. R. Choo, "An Efficient and Provably Secure Authenticated Key Agreement Protocol for Fog-Based Vehicular Ad-Hoc Networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8065–8075, 2019.

[24] S. A. Eftekhari, M. Nikooghadam, and M. Rafighi, "Security-enhanced three-party pairwise secret key agreement protocol for fog-based vehicular ad-hoc communications," *Vehicular Communications*, vol. 28, p. 100306, 2021.

[25] Y. K. Ever, "A secure authentication scheme framework for mobile-sinks used in the internet of drones applications," *Computer Communications*, vol. 155, pp. 143 – 149, 2020.

[26] Z. Ren, X. Li, Q. Jiang, Q. Cheng, and J. Ma, "Fast and Universal Inter-Slice Handover Authentication with Privacy Protection in 5G Network," *Security and Communication Networks*, vol. 2021, p. 6694058, 2021.

[27] C.-I. Fan, Y.-T. Shih, J.-J. Huang, and W.-R. Chiu, "Cross-Network-Slice Authentication Scheme for the 5th Generation Mobile Communication System," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 701–712, 2021.

[28] W. Wang, H. Huang, F. Xiao, Q. Li, L. Xue, and J. Jiang, "Computation-transferable authenticated key agreement protocol for smart healthcare," *Journal of Systems Architecture*, vol. 118, p. 102215, 2021.

[29] Y.-F. Chang, C.-Y. Tsai, and W.-L. Tai, "Comments on a Computation-Transferable Authenticated Key Agreement Protocol for Smart Healthcare," in *International Conference on Applied System Innovation (ICASI'23)*, Chiba, Japan, 2023, pp. 62–64.

[30] W. Yuanbing, L. Wanrong, and L. Bin, "An Improved Authentication Protocol for Smart Healthcare System Using Wireless Medical Sensor Network," *IEEE Access*, vol. 9, pp. 105 101–105 117, 2021.

[31] J. Lee, J. Oh, and Y. Park, "A Secure and Anonymous Authentication Protocol Based on Three-Factor Wireless Medical Sensor Networks," *Electronics*, vol. 12, no. 6, 2023.

[32] C. Wang, D. Wang, Y. Duan, and X. Tao, "Secure and Lightweight User Authentication Scheme for Cloud-Assisted Internet of Things," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2961–2976, 2023.

[33] B. K. Meher, R. Amin, A. K. Das, and M. K. Khan, "KL-RAP: An Efficient Key-Less RFID Authentication Protocol Based on ECDLP for Consumer Warehouse Management System," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3411–3420, 2022.

[34] K. Mahmood, T. Tariq, A. K. Sangaiah, Z. Ghaffar, M. A. Saleem, and S. Shamshad, "A Neural Computing-based Access Control Protocol for AI-driven Intelligent Flying Vehicles in Industry 5.0-assisted Consumer Electronics," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3573–3581, 2024.

[35] N. Zhang, Q. Jiang, L. Li, X. Ma, and J. Ma, "An efficient three-factor remote user authentication protocol based on BPV-FourQ for internet of drones," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 3319–3332, 2021.

[36] Y. Park, D. Ryu, D. Kwon, and Y. Park, "Provably Secure Mutual Authentication and Key Agreement Scheme Using PUF in Internet of Drones Deployments," *Sensors*, vol. 23, no. 4, 2023.

[37] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generation Computer Systems*, vol. 81, pp. 557–565, 2018.

[38] S. Hu, Y. Chen, Y. Zheng, B. Xing, Y. Li, L. Zhang, and L. Chen, "Provably Secure ECC-Based Authentication and Key Agreement Scheme for Advanced Metering Infrastructure in the Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 5985–5994, 2023.

[39] L. Zhang, Y. Zhu, W. Ren, Y. Zhang, and K.-K. R. Choo, "Privacy-Preserving Fast Three-Factor Authentication and Key Agreement for IoT-Based E-Health Systems," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 1324–1333, 2023.

[40] A. Aldosary, M. Tanveer, M. Ahmad, L. A. Maghrabi, E. A. Ahmed, A. Hussain, and A. A. A. El-Latif, "A Secure Authentication Framework for Consumer Mobile Crowdsourcing Networks," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2024, doi: 10.1109/TCE.2024.3473930.

[41] C.-I. Fan, C.-I. Lai, and D. Vishwasrao Medhane, "CAKE-PUF: A Collaborative Authentication and Key Exchange Protocol Based on Physically Unclonable Functions for Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 11, no. 24, pp. 39 709–39 720, 2024.

[42] 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.

[43] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02)*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[44] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[45] C. Cremers, "The Scyther Tool," 2006, https://people.cispa.io/cas.cremers/scyther/. Access on June 2023.

[46] C. J. F. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *Computer Aided Verification*, A. Gupta and S. Malik, Eds., Princeton, NJ, USA, 2008, pp. 414–418.

[47] B. Lynn, "The Tate Pairing," 2001, https://crypto.stanford.edu/pbc/notes/ep/tate.html. Access on June 2023.

[48] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[49] Y.-T. Huang, T.-S. Chen, and S.-D. Wang, "Authenticated Key Agreement Scheme for Fog Computing in a Health-Care Environment," *IEEE Access*, vol. 11, pp. 46 871–46 881, 2023.

[50] S. A. Chaudhry, K. Yahya, S. Garg, G. Kaddoum, M. M. Hassan, and Y. B. Zikria, "LAS-SG: An Elliptic Curve-Based Lightweight Authentication Scheme for Smart Grid Environments," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1504–1511, 2023.