Secure and Privacy-Preserving Data Aggregation Against Malicious Gateway in RPL-Based Internet of Things

Image Bhattarai Cong Pu
Department of Computer Science, Oklahoma State University, United States
Email: image.bhattarai@okstate.edu, cong.pu@ieee.org

2025 IEEE International Performance, Computing, and Communications Conference (IPCCC) | 979-8-3315-5286-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/IPCCC66453.2025.11304627

***Abstract*—In the era of Industry 4.0 (4IR), the Internet of Things (IoT) drives the transformation of conventional operation mode into intelligent systems through interconnecting smart devices to monitor, analyze, and optimize the target applications. In order to achieve energy-saving data transmission, a routing protocol, called RPL, has been specified for resource-challenged IoT devices and networks. In the context of 4IR, the IoT technology is being widely used for mission-critical systems, and the data collected by IoT devices might contain privacy-sensitive information. In addition, the IoT gateway could be compromised due to the lack of necessary and persistent physical and/or logical security protection. Hence, the protection of data security and privacy becomes a crucial factor for RPL-based IoT systems to realize their quality of service requirements and objectives successfully. In this paper, we propose a secure and privacy-preserving data aggregation approach, called *SPARDA*, for IoT devices and networks running the RPL routing protocol. *SPARDA* is realized with physical unclonable function, homomorphic encryption, and trapdoor function, and is perfectly integrated with the RPL routing protocol to prevent the malicious IoT gateway from either accessing, falsificating, or corrupting the real-time data from IoT devices throughout the data gathering and summarization phase. We choose an automatic security protocol verification tool, widely known as AVISPA, to analyze and verify the security specification of *SPARDA*. We also conduct an experimental study to evaluate the performance of *SPARDA* by comparing with benchmark methods. The experimental results indicate that not only does *SPARDA* protect IoT networks from malicious gateway attacks, but it also outperforms existing schemes in terms of computation and storage overheads while satisfying all critical security and privacy criteria.**

***Index Terms*—RPL, Internet of Things (IoT), Data Aggregation, Secure and Privacy-Preserving, Malicious Gateway**

I. Introduction

With the latest breakthroughs in wireless communication, mobile/edge computing, and machine learning technologies, the Internet of Things (IoT) [1] has gotten into a new period of vigorous advancement in the 21st century's third decade. The inherent ability to interconnect thousands or even millions of smart electronic devices to the Internet has made the IoT become one of the most crucial technology of modern society. For instance, currently no corner of the earth is immune from the destructive consequences of environmental degradation [2]. The IoT sensors with remote monitoring capability can assist with announcing water level and flood advisories, and predict other natural disasters such as landslides and earthquakes in prone areas, helping the corresponding authorities take prompt action to save lives. Amid all the impactful emerging technologies (e.g., artificial intelligence, 5G and connectivity technologies), the future of IoT is undoubtedly promising and something to look forward to.

A typical IoT system is composed of networked intelligent devices that use equipped processing and sensing units to collect, analyze, and transmit data they acquire from the surrounding environment [3]. To ensure the success of various IoT applications, routing plays a key role in efficiently managing data acquisition and transmission. In recent years, developing appropriate and energy-saving routing protocols for resource-restricted IoT devices and networks remains an active field of research, and attracts significant attention from commercial enterprises and technical communities, e.g., Internet Engineering Task Force (IETF) [4], Cisco [5], etc. In 2012, the IETF published a standard of routing protocol for IoT networks, commonly referred to as RPL [6]. Following the release of RPL, it quickly became a de-facto routing protocol for IoT applications such as precision agriculture, waste management and control, marine organisms exploitation, etc [7].

The RPL routing protocol brings many new features, such as automatic pathway discovery, network topology adaptation, cycle-free routes establishment, independent routing structure construction, etc [8]. Although the standardization of RPL has continued to mature and improve over the past few years, its development is still in primary stage, and it is optional to enforce certain security controls as the standard mentions [9]. As you can easily imagine, the IoT networks running RPL without proper security mechanisms might be vulnerable to devastating routing-related cyberattacks, which can compromise data security and privacy [10]. First, the IoT devices are often used in critical applications such as military defense systems, and the observations from the devices should be protected from eavesdropping and falsification. Second, from the privacy point of view, the IoT data might be analyzed in detail to derive additional private/sensitive information. For instance, the IoT smart meters are deployed to record daily consumption of electric energy and power status of households. The electricity data of households, if it is exposed to unintended or unauthorized individuals, might be used for criminal activities [11]. With access to the electricity data of households, cyber attackers could conclude that nobody is at home if the daily electricity consumption of a household is comparatively low and stable. Thus, protecting data security and privacy has become the most crucial factor in determining the success of RPL-based IoT systems [12].

Recently, some scholars have chosen to apply end-to-end

cryptographic methods [13], [14] to preserve data privacy and protect data content from eavesdroppers and hackers. Unfortunately, the cryptographic methods will cause a rapid increase in the size of transmitted data and communication overhead. In order to keep data confidential while reducing communication overhead, researchers have thought about data aggregation techniques [15], [16] for IoT systems, where the IoT gateway combines the observations from IoT devices and sends the aggregated report to the back-end server. Obviously the implicit assumption of these data aggregation techniques is that the IoT gateway operates as a verified trust anchor who will act legitimately as specified in the protocol. However, this overly-idealistic assumption conflicts with the reality of the newly emerging cyber threat environment, where the IoT gateway could be compromised as well due to the lack of necessary and persistent physical and/or logical security protection [17], [18]. Consequently, the adversary IoT gateway has a high capacity to learn and manipulate the observations from IoT devices, and then compromise the entire application. Therefore, there is an extremely urgent need to not only provide necessary security and privacy for IoT device observations, but also protect IoT systems from malicious gateway attacks.

In this paper, we concentrate on the problem of malicious gateway to protect resource-restricted IoT networks running RPL routing protocol, and propose a secure and privacy-preserving data aggregation approach, called *SPARDA*. *SPARDA* is realized with physical unclonable function, homomorphic encryption, and trapdoor function, and is perfectly integrated with the RPL routing protocol to prevent the malicious IoT gateway from either accessing, falsificating, or corrupting the IoT device observations during the data collection and aggregation process. In order to show that the proposed security approach is capable of operating safely in the adversarial environment, we implement *SPARDA* in HLPSL formal language [19], and execute the HLPSL program with the automatic security protocol analysis and verification tool AVISPA [20]. We also implement *SPARDA* and a couple of benchmark methods in Python 3, and conduct an extensive simulation-based experimental study on a Raspberry Pi 4 Model B. Based on the results of security verification and performance evaluation, we conclude that not only does *SPARDA* protect IoT networks from malicious IoT gateway attacks, but it also outperforms existing schemes in terms of computation and storage overheads while meeting all salient security and privacy requirements.

The remainder of this paper is structured as follows. In Section II, the existing data aggregation schemes are reviewed. Section III outlines the system and adversary models, along with the associated security and performance requirements. In Section IV, we introduce the proposed security approach. The security verification and analysis are provided in Section V. We describe the experimental study in Section VI, followed by the conclusion in Section VII.

II. Related Work

In [21], the authors attempt to address issues of illegal data analysis that might compromise data confidentiality in the IoT-assisted smart grid network by proposing a data aggregation scheme with privacy preservation. The basic idea of their solution is to assign a private key to each user so that s/he can encrypt her/his private information. According to the evaluation, the proposed approach shows promising results in terms of security and performance. The authors in [22] propose an user characteristics based electricity data aggregation scheme for outsourced smart grid networks, where the fog server conducts data aggregation operation based on the pre-defined rules. They argue that the proposed approach can realize the fine-grained data analysis and aggregation. However, the protection of fog server is overlooked in their approach. Once the fog server is compromised by attackers, the entire smart grid system will fail. In addition, the aforementioned data aggregation schemes are not designed with consideration of any routing mechanism, thus, they might not be integrated with RPL routing protocol properly.

In [23], a message aggregation protocol is proposed for vehicular ad hoc networks, where the aggregation operation is executed at the cluster head as well as the road side unit. First, the cluster head combines traffic messages based on their category and sends the combined report to the road side unit. Second, the road side unit aggregates all received reports from cluster heads within its region and produces a global aggregated report. However, two major drawbacks require further discussion. First, it is very challenging to maintain the stable clusters as the vehicles have high mobility. Second, forming and maintaining clusters will incur high communication overhead in vehicular ad hoc networks. The authors in [24] design a data authentication and aggregation scheme for intelligent healthcare systems. In their approach, the patient and the aggregator first perform mutual authentication and negotiate a confidential session key. After that, the aggregator combines all patients' health data based on information prioritization. Based on their findings and assessment, although it fulfills the necessary security requirements, the proposed scheme results in high computational overhead because of the adoption of elliptic curve cryptography and identity-based encryption.

In [25], the authors focus on generic resource-constrained devices and networks, and propose a privacy-preserving data aggregation protocol. Their approach relies on a trusted execution environment to perform heavyweight operations. Here, the trusted execution environment is regarded as a computer executing code on the processor. A primary limitation of their approach lies in the fact that the data aggregator has to be equipped with specific processor so that it can provide data aggregation service. In [26], a homomorphic encryption based data aggregation mechanism is proposed for IoT systems. The basic idea is that the user will decide whether to enable privacy encryption feature or not before submitting his/her data. With the assistance of fog devices, the centralized server is able to obtain the aggregated data of users.

In [27], an data aggregation mechanism, called One-Short, provides different chirps to distinctive LoRa devices for the encryption of their data. The fundamental idea of the One-Short is to investigate the periodicity of superimposed chirp

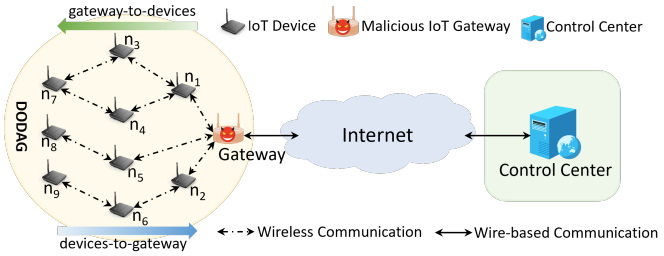


Fig. 1. System model.

signals and calculate the aggregate data using various aggregation functions. The aggregate data calculated by the gateway will provide a brief summary of sensor data across an extensive monitoring region. As there may be a potential suitability issue between LoRa and RPL [28], thus, the One-Short which is designed based on LoRa might not be suitable for RPL routing protocol. A fog computing assisted data aggregation approach is proposed for medical IoT applications in [29], where the fog node collects the encrypted data (homomorphic encryption) from medical devices and combines them using proxy re-encryption. The proposed approach has two major drawbacks. First, it does not offer physical security protection in the system, where the compromised medical devices can send false data to contaminate the aggregated report at the fog node. Second, the fog node is considered to be completely trustworthy in the system, which might not be true in the present cyber threat landscape.

In summary, one aspect missing in contemporary theory is a data aggregation approach that is capable of collecting and consolidating IoT devices' observation with cryptographically sound, resource-efficient, and privacy-aware operations as well as thwarting malicious gateway attacks within IoT systems employing the RPL routing protocol. This study's key contribution is the development of a novel data aggregation protocol against malicious IoT gateway, which opens up a promising research path within the field of IoT.

III. System and Adversary Models

A. System Model

With the emergence of Internet of Things (IoT) technology, heterogeneous IoT devices are becoming increasingly common because of their small size and low cost. Since IoT devices operate under strict limitations regarding computation, storage, and power supply, the communication links in the IoT networks exhibit reduced data transmission rates and high probabilities of channel errors. Considering the inherent limitations in computational and communication resources within IoT networks, the RPL routing protocol [6] has emerged as a robust solution for efficient data collection and dissemination.

As shown in Fig. 1, the RPL routing protocol organizes IoT devices into a multi-level branching architecture, and this architecture is formally referred to destination oriented directed acyclic graph, also abbreviated as DODAG. A DODAG includes an IoT gateway along with a group of IoT devices. Here, the major responsibilities of IoT gateway are to collect the observations from IoT devices, combine them into an aggregated observation report, and submit the report to the control center through the Internet. For certain IoT applications such as building automation system [30], the total number of IoT devices (e.g., at least 2,000) can be organized into several DODAGs (e.g., the IoT devices on the same floor will be organized into one DODAG). For simplicity, one DODAG is shown in Fig. 1. To facilitate efficient IoT device observation collection and IoT gateway control message dissemination, the RPL routing protocol supports three communication paradigms: device-to-device, gateway-to-device, and device-to-gateway. Using the device-to-gateway communication paradigm as an illustrative example, if there is a direct communication channel between an IoT device (e.g., n_1) and the IoT gateway, the IoT device can wirelessly transmit its observation directly to the IoT gateway. If the direct communication channel is absent, the IoT devices (e.g., n_4 and n_9) can choose to send observations to their preferred parents (e.g., n_1 is the preferred parent of n_4, n_6 is the preferred parent of n_9). The observation recipients will continue relaying observations to their preferred parents until the observations arrive at the IoT gateway. For instance, as shown in Fig. 1 a set of IoT devices $D = \{n_1, n_2, \cdots, n_9\}$ are deployed in an area of interest to frequently collect and send observations to the IoT gateway. Upon receiving the observations from all IoT devices within its DODAG, the IoT gateway compresses the observations into an aggregated report. Subsequently, the IoT gateway sends the aggregated report to the control center through wire-based communication for further analysis. We assume that time synchronization has been achieved between IoT devices and gateway.

B. Adversary Model

In this paper, we select a well-known Dolev–Yao (DY) adversary model [31] to specify the behaviors of attackers. Within the framework of the DY adversary model, the attackers are able to perform man-in-the-middle attacks, where wireless communication channels are no loner safe and could be eavesdropped on. In addition, the attackers have the ability to transmit a corrupted, altered, replayed, or duplicated message to either IoT device or gateway. Since the IoT devices are usually deployed in a hostile and unattended area, the attackers might find an opportunity to approach and gain access to IoT devices. However, if the attackers plan to physically compromise IoT devices, e.g., probing integrated circuits of IoT devices, their evil intention will not succeed. This is because the malicious probing operations will inevitably change the physical properties of integrated circuits of IoT devices as well as the PUF challenge-response mapping relationship. As a result, the IoT devices are unable to restore the same PUF response with the same PUF challenge, and the cryptographic keys built upon the PUF responses cannot be properly recovered. Moreover, the focus of this paper is to defend against malicious IoT gateway. Thus, we assume that the IoT devices are honest and trusted. As for the IoT gateways, they might be compromised due to the lack of necessary and persistent physical and/or logical security protection. Consequently, the attackers can learn and falsify IoT device observations for malicious purposes.

C. Security Requirements

SPARDA is designed to meet the heart objectives of IoT systems [32]: authentication, confidentiality, and integrity. From the legitimate IoT gateway point of view, only authenticated IoT devices are allowed to submit their observations along with valid identity verification information. The attackers cannot pretend to be any legitimate IoT device. Since the IoT device observations are transmitted over wireless medium, they should be protected from malicious eavesdropping and falsification. Even though the attackers can compromise the IoT gateway, they cannot illegally manipulate either IoT device observations or aggregated observation report. Any manipulation on individual IoT device observation or aggregated observation report should be detected by the control center.

IV. The Proposed Secure and Privacy-Preserving Data Aggregation Approach

In this section, we provide the design details of our secure and privacy-preserving data aggregation approach, called *SPARDA*, for IoT devices and networks running the RPL routing protocol. *SPARDA* is realized with the integration of physical unclonable function [33], homomorphic encryption [34], and trapdoor function [35]. The basic idea of *SPARDA* is that the control center initializes the IoT system through establishing and releasing key system variables and their respective functions. After that, the IoT devices and gateway register themselves with the control center to obtain their cryptographic credentials. Finally, the IoT devices submit their observations to the IoT gateway that will combine all observations into an aggregated observation report and send it to the control center. To put it succinctly, *SPARDA* consists of four phases: (A) system initialization; (B) device registration; (C) observation aggregation; and (D) aggregation verification.

A. System Initialization

The objective of system initialization phase is that the control center $\hat{C}$ finalizes the system parameters and functions, and shares them with IoT devices and gateway in the DODAG. The procedure is outlined below:

1) $\hat{C}$ arbitrarily selects three large prime numbers, p, q, and u.
2) $\hat{C}$ generates a cyclic group $\mathbb{G}$ with the prime order u and two generators g and v ($g \in \mathbb{G}$ and $v \in \mathbb{G}$).
3) $\hat{C}$ calculates $n = pq$ and $\gamma = lcm(p-1, q-1)$, and selects another generator $z \in \mathbb{Z}^*_{n^2}$. Here, *lcm* is the least common multiple function, which finds the smallest positive integer that is evenly divisible by $p-1$ and $q-1$.
4) $\hat{C}$ defines a secure hash function H: $\{0, 1\}^* \rightarrow \mathbb{G}$, a random number function f_a: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{Z}^*_u$, and a function $f_b(x) = \frac{x-1}{n}$. Here, $f_b(x)$ is a function commonly used in paillier cryptosystem and some other homomorphic encryption schemes.
5) $\hat{C}$ shares $\{u, \mathbb{G}, g, v, H, f_a, n, z\}$ with IoT devices and gateway, but saves $\{p, q, \gamma\}$ securely.

B. Device Registration

The objective of device registration phase is that the IoT devices register with the control center $\hat{C}$ to obtain their cryptographic credentials. Without loss of generality, we denote an IoT device as d_i. The following are the specific steps involved:

1) d_i arbitrarily selects a random number $r_i \in \mathbb{Z}^*_u$ and a PUF challenge che_i.
2) d_i calculates its PUF response $res_i = f^i_{puf}(che_i)$, where f^i_{puf} is the PUF of d_i.
3) d_i computes its private key and public key as $pr_i = f_a(r_i, res_i)$ and $pu_i = g^{pr_i} \in \mathbb{G}$, respectively.
4) d_i randomly chooses two numbers e_i, $t_i \in \mathbb{Z}^*_u$, and calculates a trapdoor hash value $h^{trap}_i = g^{e_i} pu_i{}^{t_i}$.
5) d_i deletes pr_i and res_i but stores r_i, che_i, e_i and t_i in the memory, and then sends $\{d_i, pu_i, h^{trap}_i\}$ to $\hat{C}$ over a secure channel.
6) $\hat{C}$ arbitrarily selects a random number $d^*_i \in \mathbb{G}$ and a random number $k_i \in \mathbb{Z}^*_u$. d^*_i is the secret identifier of d_i.
7) $\hat{C}$ calculates $\beta_i = g^{k_i}$ and $\beta^*_i = (d^*_i \,||\, DAG^*) \cdot pu_i{}^{k_i}$. Here, DAG^* is the secret DODAG identifier and shared with all IoT devices securely during the system initialization phase.
8) $\hat{C}$ sends $\{\beta_i, \beta^*_i\}$ to d_i. d_i can decrypt $\{\beta_i, \beta^*_i\}$ as $d^*_i \,||\, DAG^* = \beta^*_i \cdot (\beta_i)^{-pr_i}$ with pr_i and DAG^*.
$$\begin{aligned} d^*_i \,||\, DAG^* &= \beta^*_i \cdot (\beta_i)^{-pr_i} \\ &= (d^*_i \,||\, DAG^*) \cdot pu_i^{k_i} \cdot g^{k_i \cdot (-pr_i)} \\ &= (d^*_i \,||\, DAG^*) \cdot g^{pr_i \cdot k_i} \cdot g^{k_i \cdot (-pr_i)} \\ &= (d^*_i \,||\, DAG^*) \cdot g^{k_i \cdot pr_i} \cdot g^{-(k_i \cdot pr_i)} \\ &= (d^*_i \,||\, DAG^*) \cdot g^0 \\ &= d^*_i \,||\, DAG^*. \end{aligned}$$
9) $\hat{C}$ calculates $w = g^{DAG^*}$ and stores the entry of d_i, $\{d_i, pu_i, h^{trap}_i, d^*_i, w\}$, in the database.

C. Observation Aggregation

The objective of observation submission is that the IoT devices encrypt their observations with cryptographic credentials and send them to the control center $\hat{C}$. Without loss of generality, we denote the IoT device d_i's observation at the time ts as o^{ts}_i. Here is a step-by-step breakdown:

1) d_i calculates $res_i = f^i_{puf}(che_i)$ with che_i and f^i_{puf}, and then restores $pr_i = f_a(r_i, res_i)$ with res_i and r_i.
2) d_i computes $t_i^{'} = f_a(d^*_i, ts)$ and $s_i = pr_i \cdot (t_i - t_i^{'}) + e_i \ (mod\ u)$.
3) d_i arbitrarily selects a random number $c_i \in \mathbb{Z}^*_{n^2}$ and calculates $y_i = z^{o^{ts}_i} \cdot c^n_i \ mod\ n^2$.
4) d_i calculates $\psi_i = \left(H(d_i \,||\, ts) \cdot g^{s_i \ mod\ u} \cdot v^{o^{ts}_i}\right)^{DAG^*}$.
5) d_i sends an observation packet $pkt^{ts}_i = \{d_i, s_i, y_i, \psi_i, ts\}$ to the IoT gateway $\hat{GW}$.
6) After receiving observations from all IoT devices in the DODAG, $\hat{GW}$ calculates $s^* = \sum_{i=1}^{N} s_i \ mod\ u$, $y^* = \prod_{i=1}^{N} y_i \ mod\ n^2$, and $\psi^* = \prod_{i=1}^{N} \psi_i$, where N is the total number of IoT devices in the DODAG.
7) $\hat{GW}$ sends the aggregated observation report $pkt^{ts}_{\hat{GW}} = \{\hat{GW}, s^*, y^*, \psi^*, ts\}$ to $\hat{C}$.

D. Aggregation Verification

The objective of aggregation verification phase is that the control center $\hat{C}$ verifies and accesses the aggregated observation report. Below is a detailed sequence of actions:

1) $\hat{C}$ use the secret identifier of an IoT device, e.g., d_i^* for IoT device d_i, to calculate $t_i^* = f_a(d_i^*, ts)$.
2) $\hat{C}$ validates all observations based on the following, $\prod_{i=1}^{N} h_i^{trap} \stackrel{?}{=} g^{s^*} \cdot \prod_{i=1}^{N} pu_i{}^{t_i^*}$. If the above validation succeeds, $\hat{C}$ proceeds to decrypt the encrypted aggregation observation report. Otherwise, it terminates the aggregation verification phase.

$$\begin{aligned}
\prod_{i=1}^{N} h_i^{trap} &= g^{s^*} \cdot \prod_{i=1}^{N} pu_i{}^{t_i^*} \\
&= g^{\sum_{i=1}^{N} s_i \bmod u} \cdot \prod_{i=1}^{N} pu_i{}^{t_i^*} \\
&= g^{\sum_{i=1}^{N} \left(pr_i \cdot t_i \, - \, pr_i \cdot t_i^{'} \, + \, e_i\right)} \cdot \prod_{i=1}^{N} pu_i{}^{t_i^*} \\
&= g^{\sum_{i=1}^{N} \left(pr_i \cdot t_i \, - \, pr_i \cdot t_i^{'} \, + \, e_i\right)} \cdot \prod_{i=1}^{N} g^{pr_i t_i^{'}} \\
&= g^{\sum_{i=1}^{N} \left(pr_i \cdot t_i \, + \, e_i\right)} \cdot \prod_{i=1}^{N} g^{-pr_i t_i^{'}} \cdot \prod_{i=1}^{N} g^{pr_i t_i^{'}} \\
&= g^{\sum_{i=1}^{N} \left(pr_i \cdot t_i \, + \, e_i\right)} \\
&= g^{\sum_{i=1}^{N} \left(pr_i \cdot t_i\right)} \cdot g^{\sum_{i=1}^{N} \left(e_i\right)} \\
&= \prod_{i=1}^{N} pu_i^{t_i} g^{e_i} = \prod_{i=1}^{N} h_i^{trap}.
\end{aligned}$$

3) $\hat{C}$ obtains the encrypted aggregation observation report y^* as $o^{ts} = \left(f_b\left(y^{*^{\gamma}} \bmod n^2\right) \, / \, f_b\left(z^{\gamma} \bmod n^2\right)\right) \bmod n$. Here, the received combined ciphertext is decrypted using f_b, which helps to map the result back to the plaintext space.

$$\begin{aligned}
o^{ts} &= \frac{f_b\left(\left(\prod_{i=1}^{N} y_i \bmod n^2\right)^{\gamma} \bmod n^2\right)}{f_b(z^{\gamma} \bmod n^2)} \bmod n \\
&= \frac{f_b\left(\left(\prod_{i=1}^{N} (z^{o_i^{ts}} \cdot c_i^n) \bmod n^2\right)^{\gamma} \bmod n^2\right)}{f_b(z^{\gamma} \bmod n^2)} \bmod n \\
&= \frac{f_b\left(\left(z^{o^{ts\gamma}} \cdot \prod_{i=1}^{N} c_i^{n\gamma}\right) \bmod n^2\right)}{f_b(z^{\gamma} \bmod n^2)} \bmod n \\
&= \frac{f_b\left(\left(z^{o^{ts\gamma}} \bmod n^2\right) \cdot \left(\prod_{i=1}^{N} c_i^{n\gamma} \bmod n^2\right)\right)}{f_b(z^{\gamma} \bmod n^2)} \bmod n \\
&= \frac{f_b(z^{o^{ts\gamma}} \bmod n^2)}{f_b(z^{\gamma} \bmod n^2)} \bmod n \\
&= o^{ts}
\end{aligned}$$

4) $\hat{C}$ validates the following, $\hat{e}(\psi^*, g) \stackrel{?}{=} \hat{e}\left(\prod_{i=1}^{N} H(d_i \,||\, ts) \cdot g^{s^*} v^{o^{ts}}, w\right)$ If the above validation succeeds, $\hat{C}$ accepts the aggregated observation report. Otherwise, it terminates the aggregation verification phase.

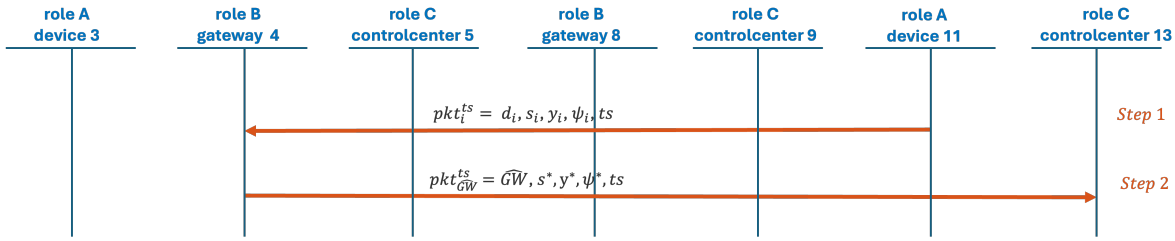
$$\begin{aligned}
\hat{e}(\psi^*, g) &= \hat{e}(\prod_{i=1}^{N} \psi_i, g) \\
&= \hat{e}(\prod_{i=1}^{N} \left(H(d_i||ts) \cdot g^{s_i \bmod u} \cdot v^{o_i^{ts}}\right)^{DAG^*}, g) \\
&= \hat{e}(\prod_{i=1}^{N} \left(H(d_i||ts) \cdot g^{s_i \bmod u} \cdot v^{o_i^{ts}}\right), g^{DAG^*}) \\
&= \hat{e}(\prod_{i=1}^{N} \left(H(d_i||ts) \cdot g^{s_i \bmod u} \cdot v^{o_i^{ts}}\right), w) \\
&= \hat{e}(\prod_{i=1}^{N} H(d_i||ts) \cdot \prod_{i=1}^{N} g^{s_i \bmod u} \cdot \prod_{i=1}^{N} v^{o_i^{ts}}, w) \\
&= \hat{e}(\prod_{i=1}^{N} H(d_i||ts) \cdot g^{\sum_{i=1}^{N} s_i \bmod u} \cdot v^{\sum_{i=1}^{N} o_i^{ts}}, w) \\
&= \hat{e}(\prod_{i=1}^{N} H(d_i||ts) \cdot g^{s^*} \cdot v^{o^{ts}}, w)
\end{aligned}$$

V. Security Verification and Analysis

In this section, we aim to prove that the design of *SPARDA* meets the pre-determined security specifications. First, we utilize AVISPA [36], which is an automated security verification tool, to examine *SPARDA* for any potential security design flaws and vulnerabilities. Second, we conduct an informal security analysis to demonstrate that *SPARDA* can securely exchange messages in the adversarial environment.

A. Security Verification using AVISPA

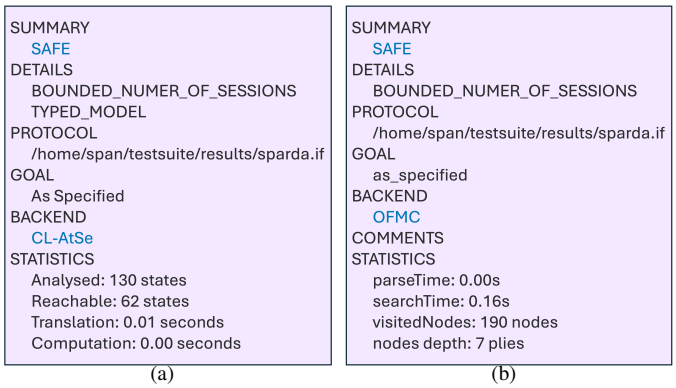
In this subsection, we verify our approach *SPARDA* using AVISPA. Here, AVISPA is a push-button security verification tool that is widely employed to examine the security requirements of communication protocols. The goal of this security verification is to show that *SPARDA* can safely operate in the adversarial settings and does not have any security design flaws and vulnerabilities. To achieve this goal, *SPARDA* is first implemented in High-Level Protocol Specification Language (HLPSL) to model the communication pattern of IoT devices, gateway, and control center. AVISPA employs two back-ends, On-the-Fly Model Checker (OFMC) and Constraint-Logic-based Attack Searcher (CL-AtSe), to analyze the operations of communication protocols. OFMC serves a purpose in inspecting the state space of a communication protocol and is suitable for detecting security vulnerabilities concerning integrity, confidentiality, and authentication. Speaking of CL-AtSe, it is useful for translating a communication protocol into a set of constraints for threat modeling. In the HLPSL implementation of *SPARDA*, we define three roles (or entities), i.e., IoT device, gateway, and control center, and message exchanges are modeled among these roles, as shown in Fig. 2. The values for pkt_i^{ts} and $pkt_{G\hat{W}}^{ts}$, as illustrated in the text, encapsulate nonces and exponents following AVISPA's



pkt_i^{ts} = nonce-2.inv(pubA.nonce-1.nonce-11).{exp(nonce-6,nonce-9).exp(nonce-7,nonce-8)}_inv(pubA).exp(h(nonce-2.nonce-3).exp(nonce-4,inv(pubA.nonce-1.nonce-11)).exp(nonce-5,nonce-9,nonce-10).nonce-3

$pkt_{\hat{GW}}^{ts}$ = nonce-13.inv(pub.nonce-12.nonce-21).{exp(nonce-16,nonce-19).exp(nonce-17,nonce-18)}_inv(pubB).exp(h(nonce-2.nonce-3.exp(nonce-14,inv(pubB.nonce-12.nonce-21)).exp(nonce-15,nonce-19),nonce-20).nonce-22)

Fig. 2. Communication sequence diagram of AVISPA security verification.



(a) (b)

Fig. 3. Security verification results using AVISPA.

formal syntax. Furthermore, auxiliary roles such as intruder, session, goal, and environment are also defined in the HLPSL program as they collectively assist with examining the security design of *SPARDA*. Finally, we execute the HLPSL program in AVISPA within Virtual Box [37] on Ubuntu 10.04. The results obtained from the security verification in AVISPA are given in Fig. 3. As shown here, *SPARDA* is a sound security protocol and does not exhibit any security design flaws and vulnerabilities that could be exploited by adversaries. The Python and HLPSL programs of *SPARDA* are available at https://github.com/congpu/SPARDA.

B. Resilience and Immunity Analysis Against Various Attacks

In this subsection, we analyze *SPARDA* against various cyber attacks such as device impersonation attack, replay attack, device capture attack, and observation modification attack.

1) Device Impersonation Attack: When the control center $\hat{C}$ receives the aggregated observation report $pkt_{G\hat{W}}^{ts}$, it first verifies the identity of IoT devices using the trapdoor function, i.e., $\prod_{i=1}^{N} h_i^{trap} \stackrel{?}{=} g^{s^*} \cdot \prod_{i=1}^{N} pu_i^{t_i^*}$. The adversary would never be able to generate s_i without the valid private key pr_i. In addition, the adversary would not be able to retrieve the secret s_i even though he/she gets a hold of pr_i. This is because s_i depends on the random number t_i. If the identity validation succeeds, $\hat{C}$ proceeds with decrypting the aggregated observation report, otherwise, it terminates the aggregation verification phase. Thus, *SPARDA* is resilient against device impersonation attack.

2) Replay Attack: In *SPARDA*, the aggregated observation report $pkt_{G\hat{W}}^{ts}$ is piggybacked with the current system time ts. When the control center $\hat{C}$ receives $pkt_{G\hat{W}}^{ts}$, It can verify whether $pkt_{G\hat{W}}^{ts}$ is fresh. If $pkt_{G\hat{W}}^{ts}$ is indeed obsolete, $\hat{C}$ will discard it. Otherwise, $\hat{C}$ will proceed with validating the identity of IoT devices. In summary, our protocol *SPARDA* is immune to replay attack.

3) Device Capture Attack: Assume that an adversary has successfully obtained a legitimate IoT device d_i. Through a probing attack, the adversary might be able to retrieve the critical information from d_i's memory such as random numbers r_i, e_i, and t_i, as well as d_i's PUF challenge che_i. With the above critical information, the adversary might attempt to generate d_i's PUF response res_i. However, this attempt would fail because even a slight change to the integrated circuit of d_i would destroy its PUF mapping. Furthermore, since the PUF response res_i is required to generate d_i's private key pr_i, the adversary would not be able to create a valid message to communicate with the IoT gateway $G\hat{W}$. Finally, since every IoT device has a unique challenge-response pair, capturing a single device would be a pointless endeavor. Thus, *SPARDA* is secured against device capture attack.

4) Observation Modification Attack: When the IoT gateway $G\hat{W}$ sends the encrypted observation report $pkt_{G\hat{W}}^{ts}$ to the control center $\hat{C}$, $\hat{C}$ validates $pkt_{G\hat{W}}^{ts}$ with $\hat{e}(\psi^*, g) \stackrel{?}{=} \hat{e}\big(\prod_{i=1}^{N} H(d_i \,||\, ts) \cdot g^{s^*} v^{o^{ts}}, w\big)$. Here, the properties of bilinear pairing ensure that both sides of the equation are equal only if the correct secret credential, i.e., s^*, is used. Since s^* is inherently tied to the IoT device d_i's private key, the adversary would not be able to generate the valid s^* without pr_i. Hence, when $\hat{C}$ validates $pkt_{G\hat{W}}^{ts}$ from $G\hat{W}$, it ensures that $pkt_{G\hat{W}}^{ts}$ has not been modified maliciously. Additionally, the observation report o_i^{ts} is hidden by the generator z in y_i, and the use of the random number c_i^n guarantees randomness. This design ensures that each encryption is unique, even for the same observation. Moreover, this randomness makes it extremely difficult for an adversary to predict or modify the message without d_i's private key pr_i. In a nutshell, *SPARDA* is resilient against observation modification attack.

Fig. 4. Experimental setup.

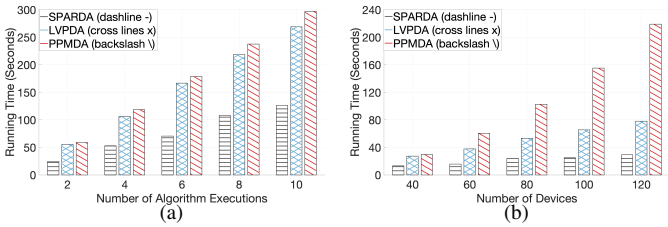


Fig. 5. Running time performance under varying numbers of algorithm executions and IoT devices.

VI. Performance Evaluation

To evaluate the performance of *SPARDA*, we carry out extensive simulated experiments on a Raspberry Pi 4 Model B which is used to mimic general-purpose IoT devices operating with constrained processing capabilities, restricted memory, and limited energy availability. The Raspberry Pi 4 Model B has a Quad core Cortex-A72 (ARM v8) 64-bit processor and 4GB LPDDR4-3200 memory, and runs Debian 10 operating system. Here, the experimental setup is shown in Fig. 4. For comparative performance analysis, we select two relevant and advanced data aggregation schemes, LVPDA [38] and PPMDA [39], as benchmark methods. Here, the LVPDA is a data aggregation method that integrates paillier homomorphic encryption with an online/offline signature mechanism. The basic idea of the PPMDA is to aggregate multidimensional data with a distributed decryption technique.

First, we obtain the running time of *SPARDA*, LVPDA, and PPMDA under varying numbers of algorithm executions and IoT devices and present the results in Fig. 5. In short, running time refers to the amount of time either *SPARDA*, LVPDA, or PPMDA takes to complete its execution, based on the size of its input (e.g., the numbers of algorithm executions or IoT devices). As can be clearly observed in Fig. 5(a), the overall running time for the three approaches exhibits growth as the number of algorithm executions is increased from 2 to 10. However, the running time of our protocol *SPARDA* is the least among all approaches. This is because *SPARDA* employs lightweight operations such as homomorphic encryption, trapdoor function, along with physical unclonable function to realize the verification of the aggregated observation report without the need to validate the identity and observation report of IoT devices separately. The LVPDA has a higher running time than our protocol *SPARDA* because it has each IoT device send its online/offline signatures to the edge server. The edge server then proceeds to verify the correctness of the received

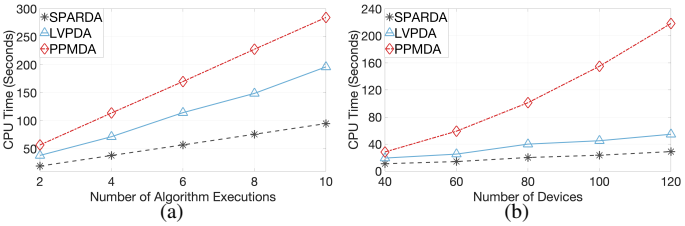


Fig. 6. CPU time performance under varying numbers of algorithm executions and IoT devices.

signatures, thereby increasing the overall running time. The PPMDA records the longest running time compared to the other two approaches because aggregation is performed on multidimensional data. With the multidimensional data, multiple measurements can be embedded into a single ciphertext. However, this approach increases the size of the ciphertext, which eventually leads to increased running time. Moreover, since the PPMDA does not employ a trusted authority (TA), each entity has to decrypt the ciphertext all by oneself, which causes an increase in running time. Likewise, the running time of *SPARDA*, LVPDA, and PPMDA with respect to the number of IoT devices ranging from 40 to 120 is shown in Fig. 5(b). As the IoT device count escalates, the PPMDA has more entities to cooperate in distrusted decryption and the LVPDA generates more individual signatures for verification, both of which contribute to a longer running time. Our protocol *SPARDA* continues to outperform both LVPDA and PPMDA.

Second, we investigate the CPU time of *SPARDA*, LVPDA, and PPMDA by changing the numbers of algorithm executions and IoT devices in Fig. 6. CPU time refers to the actual amount of time the CPU spends actively executing the operations of either *SPARDA*, LVPDA, or PPMDA. As illustrated in Fig. 6(a), the CPU time of the PPMDA is the highest among the three protocols. This is because the PPMDA operates with multidimensional ciphertexts and requires distributed decryption across entities. As a result, the highest CPU time is observed. The LVPDA has a relatively lower CPU time compared to the PPMDA. The rationale is that the LVPDA utilizes more lightweight operations than the PPMDA. However, the LVPDA still shows a higher CPU time than our protocol *SPARDA* because it requires each IoT device to sign the report and forward it to the edge server for verification. In our protocol *SPARDA*, IoT devices do not need to send their signatures to the IoT gateway. In addition, the identity and aggregated observation report verification occurs via trapdoor function and PUF. Thus, *SPARDA* shows the lowest CPU time. In Fig. 6(b), we measure the CPU time of *SPARDA*, LVPDA, and PPMDA while changing the number of IoT devices from 40 to 120. Consistent with prior results, all three approaches exhibit linear growth in CPU time while the number of IoT devices is increased. However, our protocol *SPARDA* still achieves the best performance due to the adoption of homomorphic encryption along with trapdoor function and PUFs.

Third, we examine the storage overhead of *SPARDA*, LVPDA, and PPMDA in Fig. 7, where the number of IoT devices is changed from from 40 to 120. Here, storage overhead refers to the memory space either *SPARDA*, LVPDA,

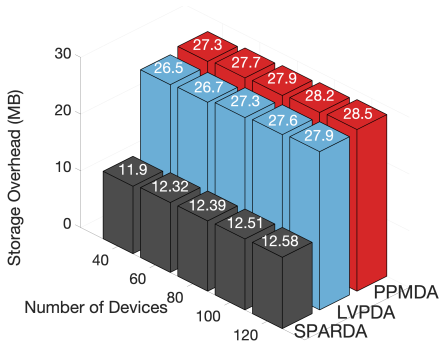


Fig. 7. Storage overhead performance under varying numbers of IoT devices.

or PPMDA requires in the Raspberry Pi 4. It can be seen from Fig. 7 that the PPMDA requires the largest amount of memory space to operate. This is because that the PPMDA stores large ElGamal ciphertexts, each of which comprises two group elements per value. Moreover, in the PPMDA distributed decryption requires each entity to store public keys as well as partially decrypted values for decryption, which causes an increase to storage overhead. The LVPDA also has a higher storage overhead than our protocol *SPARDA* due to the need to store offline signature states for each IoT device. In addition, the edge server needs to buffer and execute the offline verification algorithm for each IoT device after receiving the offline tags. This also causes an increase in memory usage. On the contrary, our protocol *SPARDA* utilizes a trapdoor function and a lightweight pairing check to verify the identity of IoT devices and aggregated observation report, instead of generating large ciphertexts or storing additional protocol states.

Lastly, we evaluate the energy consumption of *SPARDA*, LVPDA, and PPMDA by varying the numbers of algorithm executions and IoT devices in Fig. 8. Here, the solid bar area reflects the growth in energy consumption when the number of algorithm executions is increased by 1 and the number of IoT devices is increased by 10 in Fig. 8(a) and Fig. 8(b), respectively. General speaking, energy consumption refers to the amount of electrical energy (joule) used by the computational processes required to execute either *SPARDA*, LVPDA, or PPMDA. Compared to *SPARDA* and LVPDA, the PPMDA exhibits greater complexity due to its operational logic and multidimensional data, which causes an increase in the size of ciphertexts. As a result, executing the relevant operations would consume more energy. Moreover, the distributed decryption technique used across multiple entities without the assistance of a trusted authority (TA) leads to more energy consumption. The LVPDA shows a higher energy consumption than our approach *SPARDA* because of a greater number of data transmissions among IoT devices. In the LVPDA, since each IoT device transmits its report along with its digital signature, the edge server needs to retrieve the stored cryptographic information and verify each IoT device using its stored offline signature state. Our protocol *SPARDA* demonstrates minimal energy usage. This is because *SPARDA* uses lightweight operations like homomorphic encryption and trapdoor function along with PUFs, making it consume less

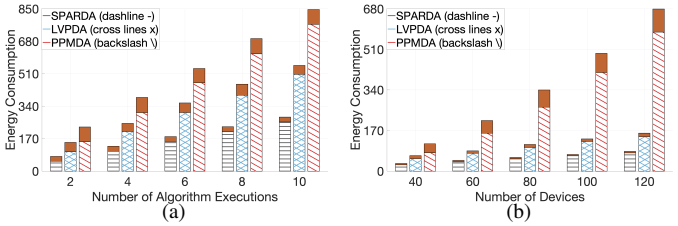


Fig. 8. Energy consumption performance under varying numbers of algorithm executions and IoT devices.

energy for secure data aggregation and verification.

VII. Conclusion

In this paper, we proposed a secure and privacy-preserving data aggregation scheme, called *SPARDA*, against malicious gateways in RPL-based Internet of Things (IoT) systems. In *SPARDA*, IoT devices send their observations to the IoT gateway which will aggregate all observations into an aggregated observation report and send it to the control center. *SPARDA* is constructed with homomorphic encryption, physical unclonable function, and trapdoor function and integrated with the RPL routing protocol to prevent the malicious IoT gateway from either accessing, falsifying, or corrupting the real-time data from IoT devices throughout the data gathering and summarization phase. We employed AVISPA to assess *SPARDA*'s security-related design features and conducted an informal security analysis. The security verification and analysis have proved that our protocol *SPARDA* can safely operate in the cyber-threat environments and is secure against various cyberattacks. In addition, we implemented *SPARDA* along with two benchmark methods and carried out extensive experiments on a Raspberry Pi 4. The comprehensive simulation results show that *SPARDA* outclasses its peer methods in terms of computation and storage overheads while meeting all salient security and privacy requirements.

For future work, we plan to integrate *SPARDA* with blockchain technique for trust evaluation. First, the trustworthiness of IoT devices will be evaluated based on their observations (e.g., data accuracy and uptime) using machine learning algorithms (e.g., random forest and long short-term memory), and then securely stored on a private blockchain (e.g., hyperledger fabric blockchain) to ensure data integrity and tamper resistance. Second, a smart contract is specifically designed to validate trust scores and identity suspected adversarial IoT devices. Finally, the adversarial IoT devices are added into the revocation list by revoking their cryptographic keys, which ensures that adversarial IoT devices are blocked from reentering the network. In this way, a decentralized and tamper-proof trust management framework can be built to protect RPL-based IoT systems.

Acknowledgment

This work was supported by the National Science Foundation (NSF) through SaTC under Award 2333777. The Python and HLPSL programs for *SPARDA* are publicly available at https://github.com/congpu/SPARDA.

REFERENCES

[1] O. Aouedi, T. Vu, A. Sacco, D. Nguyen, K. Piamrat, G. Marchetto, and Q. Pham, "A Survey on Intelligent Internet of Things: Applications, Security, Privacy, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 2, pp. 1238–1292, 2025.

[2] A. Mishra, "Internet of Things-based sensors for environmental monitoring," in *Emerging Sensors for Environmental Monitoring*, 2025, pp. 167–186.

[3] W. Jiang, Y. Zhang, H. Han, and J. Mu, "Generative AI for Consumer Internet of Things: Challenges and Opportunities," *IEEE Consumer Electronics Magazine (Early Access)*, pp. 1–10, 2025.

[4] *Internet Engineering Task Force*, Last accessed: June 24, 2025, https://www.ietf.org/.

[5] *Routing Protocol for LLN (RPL) Configuration Guide*, Last accessed: June 24, 2025, https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/rpl/configuration/15-mt/rpl-15-mt-book.html.

[6] T. Winter *et al.*, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, 2012.

[7] A. Zilberman, A. Dvir, and A. Stulman, "IPv6 Routing Protocol for Low-Power and Lossy Networks Security Vulnerabilities and Mitigation Techniques: A Survey," *ACM Computing Surveys*, vol. 57, no. 11, pp. 1–77, 2025.

[8] C. Pu, "Sybil Attack in RPL-Based Internet of Things: Analysis and Defenses," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4937–4949, 2020.

[9] H. Albinali and F. Azzedin, "Towards RPL Attacks and Mitigation Taxonomy: Systematic Literature Review Approach," *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 5215–5238, 2024.

[10] C. Pu, "Cyberattacks and Countermeasures in RPL-based Industrial Internet of Things," in *Intelligent Cyber-Physical Systems Security for Industry 4.0*, 2022, pp. 57–77.

[11] L. Xiao, H. Chen, S. Xu, Z. Lv, C. Wang, and Y. Xiao, "Reinforcement Learning-Based False Data Injection Attacks in Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 4, pp. 3475–3484, 2025.

[12] B. Groves and C. Pu, "A Gini Index-Based Countermeasure Against Sybil Attack in the Internet of Things," in *IEEE MILCOM*, 2019, pp. 1–6.

[13] R. Ponnuru, S. Kumar, M. Azab, and G. Alavalapati, "BAAP-FIoT: Blockchain Assisted Authentication Protocol for Fog-enabled Internet of Things Environment," *IEEE Internet of Things Journal*, vol. 12, no. 11, pp. 15 681–15 696, 2025.

[14] C. Pu, I. Ahmed, and S. Chakravarty, "Resource-efficient and data type-aware authentication protocol for Internet of Things Systems," in *IEEE TPS*, 2023, pp. 101–110.

[15] A. Chaudhary and S. Peddoju, "ADA2- IoT: An adaptive data aggregation algorithm for IoT infrastructure," *Internet of Things*, vol. 27, p. 101299, 2024.

[16] V. Vo, D. Le, S. Raza, M. Kim, and H. Choo, "Active Neighbor Exploitation for Fast Data Aggregation in IoT Sensor Networks," *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 13 199–13 216, 2024.

[17] J. Zhao, F. Huang, H. Hu, L. Liao, D. Wang, and L. Fan, "User security authentication protocol in multi gateway scenarios of the Internet of Things," *Ad Hoc Networks*, vol. 156, p. 103427, 2024.

[18] C. Pu and K. Choo, "Lightweight Sybil Attack Detection in IoT based on Bloom Filter and Physical Unclonable Function," *Computers & Security*, vol. 113, p. 102541, 2022.

[19] Y. Chevalier *et al.*, "A high level protocol specification language for industrial security-sensitive protocols," in *Proc. SAPS*, 2004, pp. 1–13.

[20] A. Armando *et al.*, "The AVISPA tool for the automated validation of internet security protocols and applications," in *Proc. CAV*, 2005, pp. 281–285.

[21] J. Wang, L. Wu, S. Zeadally, M. Khan, and D. He, "Privacy-preserving Data Aggregation against Malicious Data Mining Attack for IoT-enabled Smart Grid," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–25, 2021.

[22] H. Li, X. Li, and Q. Cheng, "A fine-grained privacy protection data aggregation scheme for outsourcing smart grid," *Frontiers of Computer Science*, vol. 17, no. 3, p. 173806, 2023.

[23] B. Bhabani and J. Mahapatro, "CluRMA: A cluster-based RSU-enabled message aggregation scheme for vehicular ad hoc networks," *Vehicular Communications*, vol. 39, p. 100564, 2023.

[24] P. Verma and D. Gupta, "A pairing-free data authentication and aggregation mechanism for Intelligent Healthcare System," *Computer Communications*, vol. 198, pp. 282–296, 2023.

[25] M. Wang, K. He, J. Chen, R. Du, B. Zhang, and Z. Li, "PANDA: Lightweight non-interactive privacy-preserving data aggregation for constrained devices," *Future Generation Computer Systems*, vol. 131, pp. 28–42, 2022.

[26] Z. Zeng, Y. Liu, and L. Chang, "A Robust and Optional Privacy Data Aggregation Scheme for Fog-Enhanced IoT Network," *IEEE Systems Journal*, vol. 17, no. 1, pp. 1110–1120, 2023.

[27] N. Hou, X. Xia, Y. Wang, and Y. Zheng, "One shot for all: Quick and accurate data aggregation for LPWANs," *IEEE/ACM Transactions on Networking*, vol. 32, no. 3, pp. 2285–2298, 2024.

[28] B. Sartori, S. Thielemans, M. Bezunartea, A. Braeken, and K. Steenhaut, "Enabling RPL Multihop Communications based on LoRa," in *Proc. IEEE WiMob*, 2017, pp. 1–8.

[29] C. Chakraborty, S. Othman, F. Almalki, and H. Sakli, "FC-SEEDA: Fog computing-based secure and energy efficient data aggregation scheme for Internet of healthcare Things," *Neural Computing and Applications*, vol. 36, no. 1, pp. 241–257, 2024.

[30] J. Martocci *et al.*, *Building Automation Routing Requirements in Low-Power and Lossy Networks*, 2010.

[31] Q. Do, B. Martini, and K. Choo, "The role of the adversary model in applied security research," *Computers & Security*, vol. 81, pp. 156–181, 2019.

[32] F. Lins, F. Freitas, O. Nóbrega, and G. Valença, "Security Requirements Engineering Approaches for IoT-Based Systems: A Comprehensive Review and Open Research Challenges," in *IEEE WF-IoT*, 2024, pp. 1–6.

[33] A. Al-Meer and S. Al-Kuwari, "Physical Unclonable Functions (PUF) for IoT Devices," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–31, 2023.

[34] T. Doan, M. Messai, G. Gavin, and J. Darmont, "A survey on implementations of homomorphic encryption schemes," *The Journal of Supercomputing*, vol. 79, no. 13, pp. 15 098–15 139, 2023.

[35] N. Döttling, S. Garg, Y. Ishai, G. Malavolta, T. Mour, and R. Ostrovsky, "Trapdoor Hash Functions and Their Applications," in *Proc. CRYPTO*, 2019, pp. 3–32.

[36] *Automated Validation of Internet Security Protocols and Applications*, Last accessed: June 17, 2025, http://www.avispa-project.org/.

[37] *VirtualBox*, Last accessed: June 17, 2025, https://www.virtualbox.org/.

[38] J. Zhang, Y. Zhao, J. Wu, and B. Chen, "LVPDA: A Lightweight and Verifiable Privacy-Preserving Data Aggregation Scheme for Edge-Enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4016–4027, 2020.

[39] X. Zuo, L. Li, H. Peng, S. Luo, and Y. Yang, "Privacy-Preserving Multidimensional Data Aggregation Scheme Without Trusted Authority in Smart Grid," *IEEE Systems Journal*, vol. 15, no. 1, pp. 395–406, 2021.