RESEARCH PAPER
Contrib. Sci. & Tech Eng, 2025, 2(1)
Homepage: https://cste.journals.umz.ac.ir/
DOI: 10.22080/cste.2025.28688.1013

A Novel Identity based Internet of Things Routing Protocol based on RPL

Hamid Shabanipour [1], Reza Ebrahimi Atani [1]*, Arian Arabnouri [1]

[1] *Department of Computer Engineering, University of Guilan, P.O. Box 3756, Rasht, Iran.*

Article Info
Received 02 March 2025
Accepted 08 April 2025
Available online 09 April 2025

Keywords:
Internet of Things;
RPL protocol;
Security;
Digital signature;
IBS cryptography.

Abstract:
The Internet of Things (IoT) is one of the most transformative technologies of the modern era, enabling seamless connectivity and data exchange across a wide range of applications, including smart cities, healthcare, agriculture, and industrial automation. However, the rapid growth of IoT has introduced significant challenges, particularly in terms of security. Among these challenges, securing routing protocols in low-power and lossy networks (LLNs) is critical, as they are vulnerable to various attacks, such as rank spoofing and version number attacks, which can disrupt network topology and compromise data integrity. This paper proposes a novel identity-based routing protocol for IoT, built on the RPL (Routing Protocol for Low-Power and Lossy Networks) framework. Our approach utilizes Identity-Based Signature (IBS) cryptography to enhance the security of RPL against rank and version number attacks. By utilizing a lightweight digital signature scheme, our protocol ensures that only legitimate nodes can modify the network topology, preventing malicious actors from forging rankings or version numbers. The proposed scheme is designed to be computationally efficient, making it suitable for resource-constrained IoT devices. We provide a comprehensive security analysis, demonstrating that our protocol offers robust resistance to forging attacks. Additionally, we evaluate the scheme's performance in terms of time and energy consumption, showing that it is both efficient and scalable for large-scale IoT deployments. Our results indicate that the proposed identity-based routing protocol not only enhances the security of RPL but also maintains low overhead, making it a practical solution for securing IoT networks in real-world applications.

***Corresponding Author:** rebrahimi@guilan.ac.ir

Supplementary information: Supplementary information for this article is available at https://cste.journals.umz.ac.ir/

Please cite this paper as: Ebrahimi Atani, R., Shabanipour, H., & Arabnouri, A. (2025). A Novel Identity based Internet of Things Routing protocol based on RPL. Contributions of Science and Technology for Engineering, 2(1), 19-27. doi:10.22080/cste.2025.28688.1013.

1. Introduction

The rapid advancement of technology has led to the emergence of the Internet of Things (IoT), a paradigm that connects physical and virtual objects through the Internet, enabling seamless communication and data exchange. IoT has become a cornerstone of modern information and communication technology, with applications spanning diverse fields such as healthcare, agriculture, smart cities, industrial automation, and transportation. IoT systems collect and manage vast amounts of data by integrating sensors, actuators, and communication protocols, transforming how we interact with the world around us.

However, the proliferation of IoT devices has introduced significant challenges, particularly in terms of security. IoT networks often consist of heterogeneous nodes with limited computational power, memory, and energy resources. These constraints make them vulnerable to various attacks, especially at the network layer, where routing protocols play a critical role in ensuring reliable communication [1]. Among these protocols, the Routing Protocol for Low-Power and Lossy Networks (RPL) has been widely adopted due to its efficiency in managing resource-constrained environments [2]. RPL is designed to optimize network communicationwith limited bandwidth, high packet loss, and low-power devices, making it ideal for IoT applications. RPL is based on a routing source and operates on top of link-layer mechanisms such as MAC and IEEE 802.15.4 layers. This is a reactive protocol in which paths are found and created in the required time. To check different needs and various applications, the ROLL Working Group defines a set of indexes and constraints for links and nodes suitable for low-power and lossy networks on the RPL protocol [2].

Despite its advantages, RPL is susceptible to several internal attacks, such as rank spoofing and version number attacks. In a rank spoofing attack, a malicious node manipulates its rank to attract traffic, creating non-optimal routes, loops, and increased end-to-end delays. Similarly, in a version number attack, an attacker broadcasts a fake version number, forcing nodes to rebuild the network topology, which leads to increased control overhead, energy consumption, and packet loss. These attacks exploit the inherent trust mechanisms in RPL, undermining the stability and efficiency of the network [3].

*ISSN **3060-6578***

To address these vulnerabilities, various security mechanisms have been proposed, including intrusion detection systems, hash functions, and digital signatures [4-6]. However, many of these solutions are either computationally expensive or fail to provide comprehensive protection against internal attacks [7-13]. This paper proposes a novel identity-based routing protocol for IoT built on the RPL framework. Our approach utilizes Identity-Based Signature (IBS) cryptography to secure the network against rank spoofing and version number attacks. Using lightweight digital signatures, our protocol ensures that only legitimate nodes can modify the network topology, preventing malicious actors from disrupting the network.

The remainder of this paper is organized as follows: Section 2 provides background information on identity-based cryptography and RPL. Section 3 reviews related work on securing RPL against rank and version number attacks. Section 4 presents the proposed signature scheme and its architecture. Section 5 analyzes the security and performance of the proposed scheme. Finally, Section 6 concludes the paper and discusses future work.

2. Preliminaries

2.1. Identity-Based Cryptography

The Identity-Based Encryption scheme (IBE) consists of four phases: setup, private key generation, encryption, and decryption [14], which are briefly described below:

Setup: In this phase, the PKG provides everyone with the public key after generating the public and private key pairs.

Private Key generation: In this phase, the receiver node receives its private key corresponding to its personal information after authentication for the PKG.

Encryption: In this phase, the sender node encrypts its message using the receiver node specification and the public key PKG and receives the encrypted message.

Decryption: In this phase, the receiver node decrypts the encrypted message using its private key and retrieves the original message.

An Identity-Based Signature (IBS) will be as follows steps, and vice versa cryptography:

Setup: In this step, the PKG generates a public-private key pair and distributes the public key to all participants while securing the private key.

Private Key Generation: In this phase, the sender node (signer) requests its private key from the PKG. After authenticating the sender's identity, the PKG generates and provides the private key corresponding to the sender's specification.

Signature Generation: In this phase, the sender node uses its private key to generate a digital signature for the message. The signed message is then sent to the receiver.

Signature Verification: In this phase, the receiver node verifies the authenticity of the signature using the sender's public key and identity. The message is accepted if the signature is valid; otherwise, it is rejected.

2.2. Bilinear Pairing

Bilinear pairings are a widely used function in asymmetric cryptography [15]. They are also used in several other cryptography contexts, such as digital signatures, searchable encryption, and IBE.

Consider three cyclic groups $(G_1,.)$, $(G_2,.)$ and $(G_3,.)$ of prime order q. A bilinear pairing is defined as a function $e: G_1 \times G_1 \rightarrow G_2$ with the following properties:

Bilinearity: The following Equation holds for

$$\forall a, b \in Z_q, \forall g_1 \in G_1, \forall g_2 \in G_2: e({g_1}^a, {g_2}^b) = e(g_1, g_2)^{ab} \quad (1)$$

This property can also be expressed as:

$$\forall a, b, c \in Z_q, \forall g_1, g_2 \in G_1, g_3 \in G_2: e({g_1}^a . {g_2}^b, {g_3}^c) = e({g_1}^a, {g_3}^c) . e({g_2}^b, {g_3}^c) \quad (2)$$

Non-degeneracy: Considering g_1 to be the generator of G_1 and g_2 to be the generator of G_2, $e(g_1, g_2)$ is the generator of G_3.

Computability: An algorithm exists to calculate $e(g_1, g_2)$ where $g_1 \in G_1, g_2 \in G_2$. Weil pairing and Tate pairing can be mentioned as some methods for calculating bilinear pairings on an elliptic curve. Bilinear pairing is categorized into symmetric and asymmetric types. In the case $G_1 = G_2$, the pairing is symmetric type. In addition to the aforementioned characteristics of bilinear pairings, the symmetric bilinear pairing has commutative properties as well $(e(a, b) = e(b, a))$. Our proposed scheme uses a symmetric type of bilinear pairings.

2.3. Routing Mechanism in RPL

Due to scalability and resource constraints in LoWPAN6 environments, selecting and designing an appropriate routing protocol for such networks is challenging. In these environments, nodes are connected through multi-hop wireless communication to a root node, which is a gateway to connect the LoWPAN6 network to other networks. The root node is also responsible for collecting and controlling data from other nodes [2, 3, 16].

The RPL protocol constructs Directional Acyclic Graphs (DAGs) without cycles, known as DODAGs (Destination-Oriented Directed Acyclic Graphs). A DODAG is a DAG with a single root node as its destination. The root node of a DODAG has no outgoing edges, meaning it does not forward data outside the network. A DODAG is uniquely identified by a set of parameters, including RPLInstanceID, DODAGID, and the version number. The version number represents the latest version of the DODAG graph created by the root. When the version number is incremented, the root initiates the creation of a new version of the DODAG [3, 5].

In the DODAG graph, each node calculates its rank based on the DIO (DODAG Information Object) message sent by the root. The DIO message contains information that enables a node to identify an RPL Instance, learn its

configuration parameters, select its DODAG parent set, and maintain the DODAG structure. The root node broadcasts the DIO message using a propagation radius mechanism. When joining the graph, nodes within the message's propagation radius select the root node as their parent. Nodes that receive the DIO message then forward it to their children, ensuring that the message reaches nodes outside the root's immediate range.

Upon receiving the DIO message, nodes identify their parent based on the information contained in the message and select the preferred parent—typically the node with the lowest cost for forwarding traffic to the root. The rank of a node determines its position relative to other nodes in the DODAG, with the root having the lowest rank. The rank increases as nodes move further away from the root. The exact calculation of a node's rank depends on the Objective Function (OF), which may consider factors such as topological distance, link metrics, or other features [5, 6].

The RPL protocol is primarily designed to optimize multi-point to single-point traffic flow, which is common in set-axis networks. According to standard specifications, RPL routes are optimized for traffic directed to or from one or more root nodes, which act as sinks in the topology. While RPL can also support point-to-point and multi-point to single-point traffic, these functionalities are less developed. For multi-point to single-point traffic, RPL requires only DIO and DIS (DODAG Information Solicitation) control messages. However, for the other two types of traffic, additional control messages—DAO (Destination Advertisement Object) and DAO-ACK (optional)—are also required.

In RPL, upward routing is established from each node to the root to enable multi-point to single-point communication. This communication paradigm is particularly important in set-axis networks, where multiple sensors send their data to a common point (in this case, the root of the DODAG). The root node can act as a boundary router, forwarding the collected data to time-series data storage systems or other networks.

By default, upward routing in RPL is performed through each node's preferred DIO parent. Each node maintains a set of one-hop neighbors, known as the candidate neighbor set. In the RPL protocol, a node selects a set of candidate parents from nodes with a lower rank than itself. Typically, each node selects one parent (or multiple parents, if supported) to forward packets toward the root of the DODAG. When a node needs to send data to the root, it transmits the data to its preferred parent, which then forwards the data to its own parent, eventually reaching the root. If the preferred parent is unavailable, the node can use an alternative parent from its candidate parent set to forward the data [3, 16]. For downward routing, which is required for single-point to multi-point and single-point to single-point communications, RPL supports optional features implemented through DAO (Destination Advertisement Object) and DAO-ACK messages. The DAO message propagates destination information upward in the DODAG. By default, point-to-point communication is achieved through upward routing, where a sensor node sends data to the root via its preferred parent. Upon receiving the data, the root forwards it to the intended destination.

When a node joins the network, it can either send a DIS (DODAG Information Solicitation) message to request a DIO message or wait for periodic DIO messages broadcast by other nodes. Each node in the DODAG periodically sends DIO messages at intervals determined by the trickle scheduler [17]. A node that has not yet joined the DODAG selects the sender of the DIO message as its temporary parent.

In this paper, we focus on two types of internal attacks that target the DODAG structure: rank attacks and version number attacks. These attacks exploit vulnerabilities in RPL's graph structure. The following sections will briefly review each attack type.

2.3.1. Rank Attack

At RPL, the rank value increases from root to child. This attack can lead to creating non-optimal routes, creating loops, not allowing to use optimal routes in the topology, reducing the rate of packets received through end-to-end delay, and increasing the number of attacker nodes. The topology around the attacker node is constantly changing, and as a result, the neighbor’s topology around this node is also updated, causing greater control over on the nodes.

2.3.2. Version Number Attack

The attack occurs by releasing a higher version number of the DODAG tree. When the nodes receive a higher version number from the DIO message, they start creating a new DODAG tree. This case can lead to the creation of a new non-optimized topology that is incompatible and inconsistent within itself. This attack leads to creating a loop and incompatibility in the neighbor’s rank of the attacker node. Attack version number 18 times increases the control overhead and affects energy consumption and channel access. It also reduces the packet-receive ratio by more than 30% and doubles end-to-end delay in the network. An attacker who is at a distance greater than the root also causes the maximum amount of overhead and packet loss.

3. Related Works

3.1. Related Schemes

So far, research on the security of IoT routing protocols has mostly focused on the design of intrusion detection systems and hash functions, as well as the use of digital signatures and the principles of cryptographic or hybrid algorithms. In these secure designs, it should be noted that the node structure in wireless networks, especially low-power and lossy networks, should also be taken into account. The node's structure in such networks has limited memory, low-capacity power, and batteries that cannot store and keep a large volume of data. Therefore, special attention should be paid to the structural features of such networks when designing protocols. In this paper, we review several defense schemes introduced against fake rank and fake version number. In the following, three major defense plans

will be introduced and the weaknesses of each will be examined.

Dvir et al. [18] was introduced as a security solution (by adding a reverse hash chain to the DIO message) against these two types of vulnerabilities. The VeRA scheme prevents inappropriate behavior of nodes by impersonating the root node of DODAG and sending DIO messages with an illegally increased version number, as well as inappropriate behavior in the nodes in order to propagate a reduced rank illegally. VeRA has two phases: the launch phase and the version number update phase.

At the startup phase, the root of the DODAG generates a random number r and computes a hash chain called the version number hash chain, whose size is n + 1: $V_n, V_{n-1},\ldots, V_1, V_0 \mid V_n = h(r)$, $V_i = h(V_{i+1})$. For each element of V_i, the root generates a new random number xi and computes a new hash chain called the rank hash chain, whose size is $l+1$[1]: $R_{i,l}, R_{i,l-1}, \ldots, R_{i,1}, R_{i,0} \mid R_{i,0} = h(x_i)$, $R_{i,j} = h(R_{i,j-1})$. The root node then attaches the digital signature $\{\{V_0, MAC_{V_1}(R_{mrh})\}_{sign}$sing to the DIO message, where MAC means a message verification code on the highest-ranking hash function that is provided by the root DODAG is calculated ($R_{mrh} = R_{1,l}$), where $R_{1,l}$ is the next element V_1. Finally, the root of DODAG sends this DIO message to the nodes in multicast. This submission can continue until the version number is updated time. At the same time as receiving this DIO message, each intermediate node receives and verifies its version number by data verifying the validity from the signed message and sending this message to its neighbors in multicast by Drop timer expiration. At the version number update phase, in order to update a version of a DODAG graph, from VN_{i-1} to VN_i, the root node of a DIO message containing $\{VN_i, V_i, MAC_{V_{i+1}}(R_{i+1,l}), R_{i,Rank_{sender}}\}$ sends where $Rank_{sender}$ is the new root rank. Each intermediate node receives a message confirming that the value of this new version is higher than the old version or no if $VN_i > VN_{i-1}$ is the next step for verification and chek $V_0 = h^{(VN_i - VN_0)}(V_i) = h^i(V_i)$. Whenever any of these confirmations are rejected by the middle nodes, the process of updating the version number is not performed.

The main goal and purpose of TRAIL by Landsmann et al. [19] is path validation confirmation to upward and root using a Turnover period message. A child node that receives a rank notification from its parent sends a test message with a random variable η to its parent. The parent also adds its rating (j) to the test message (j, η) and sends it to the root. At each middle node receiving this message, two items must be confirmed: 1. The test message rank must be higher and more than the middle receiving node. 2. The rank of the sender node must be between the rank of the middle node of the recipient of the message and η. While occurring, the violation of the above items means a rank violation, and the test message is discarded. In this case, the corresponding sub-DODAG connection is interrupted or begins a local repair process.

When this test message reaches the root, the root adds the version number of the test message driver and sends it down. Each middle node verifies whether it contains a rank j greater than its own or no before sending it. Any violation in this case cause stops the sending of the message. At the last node (driver), the signature confirmation is done and X adapts itself and receives its version number and rank from its parent. As the rank notification is done consistently towards the root, no ranking violation is done by valid nodes, and the upward flow will be completely valid. The highest-ranking node that doesn't succeed in performing this test and process will, in fact, be the largest sub-DODAG affected by the rank spoofing attack. It should be noted that although a chain of k Malicious nodes attached are able to repeat $k-1$ times the rank values secretly at different times, they are counted in the test message as valid nodes. However, the attack does not reduce the attacker's rank amount.

The malicious node that receives the $< \eta, A >$ test message from its child or children has the authority to put or not put the child or children in the array. It can also rearrange the array or place its children in an invalid and incorrect position. In addition, the malicious node may try to disassociate itself from authentication in the hierarchical structure by not sending its variable value to its parent.

Stephen and Arockiam [20] proposed an algorithm called RIADRPL to avoid a loop being created by an adversary node with an increased rank value.

Tandon and Srivastava [21] designed a trust-based mechanism, which is a defense scheme against Sybil and Rank attacks. In this scheme, parameters such as rank, energy depletion, behavior, etc. will be used to compute a trust value for both the parent and child nodes to discover the adversary behavior of the attacker node.

DCTM-RPL], proposed by Hashemi and Aliee [22], is a dynamic and comprehensive trust model for RPL that ensures secure communication by maintaining a confidence value that exceeds a threshold.

LEADER Scheme: Karmakar et al. [23] developed the Low-overhead Rank Attack Detection scheme for securing RPL, using a lightweight message authentication code (HMAC-LOCHA) to maintain message integrity and authenticity.

SMTrust: This scheme, proposed by Muzammal et al., [24] ensures that only reliable nodes participate in the routing process. It uses mobility-based trust metrics to defend against various attacks, including version number, rank, blackhole, and greyhole attacks.

Hybrid RPL Protocol: Jhanjhi et al. [25] introduced a hybrid RPL protocol that utilizes machine learning techniques to detect attacks and select influential parameters to mitigate their impact.

PCC-RPL:], proposed by Pishdaret al. [26] , is a trust-based mechanism that enables parent nodes to scan the conditions and activities of their children, alerting the root about any suspicious child nodes.

Intrusion Detection Using Machine Learning: Belavagi and Muniyal [27] used machine learning algorithms like K-

means and random forest to detect multiple types of intrusions and build predictive models for attack detection.

IBS Scheme [28]: This scheme provides a method for reducing the impact of rank and version number spoofing, employing five algorithms to safeguard the DODAG graph from these attacks.

Setup: In this phase, it receives a security parameter k as input. PKG selects groups G_1 and G_2 from P; G_1 is a generated incremental group by P and G_2 is a multiplier group. Also, PKG has a bilinear pairing of $e = G_1 \times G_1 \rightarrow G_2$ and selects the $H_1: \{0,1\}^* \rightarrow Z_P^*$ and $H_2: \{0,1\}^n \times G_2 \times G_1 \rightarrow Z_P^*$ hash functions. The size of the version number or rank of a node is n bits. Then PKG randomly selects a main secret key $msk \in Z_P^*$ and sets the main public key $P_{pub} = msk.P$ and calculates $g = e(P,P)$. Finally, PKG publishes the general parameters of the system and holds the main secret key of msk.Params $= \{G_1, G_2, n, e, P, P_{pub}, g, H_1, H_2\}$. It should be noted that in this scheme, the root of DODAG acts as a PKG.

Extraction Algorithm: A node sends its ID to PKG, and PKG generates the private key of the node $S_{ID} = \frac{1}{H_1(ID)+msk}$ and sends this private key from a secure channel to the node. The node's public key is according to calculated operation $Q_{ID} = H_1(ID).P + P_{pub}$ and the signer ID is represented as ID_s.

Offline Signature Generation Algorithm: takes the private key of the S_{ID_s} signer as input. The steps of the algorithm are as follows: Selecting x and λ from Z_P^* at random. Calculating $r = g^x$ and $S = \lambda^{-1}(S_{ID_S} + P)$.Offline signature is $\pi = (x, r, \lambda, S)$.

Online Signature Generation Algorithm: A message m takes the offline signature π as input. The steps of the algorithm are as follows: Calculating $h = H_2(m, r, S)$ and $\varphi = \lambda(x + h) mod\ P$. The complete signature is $\sigma = (h, \varphi, S)$.

Signature Verification Algorithm: Receives signature 1 and signature IDs as input. The steps of the algorithm are as follows: Calculating $L = \varphi * S$ and $r = e\left(L, \frac{P.Q_{ID_S}}{P+H_1(ID_S).P+P_{pub}}\right).g^{-h}$. If $h = H_2(m, r, S)$ the signature is accepted and otherwise returns $\perp$.

To protect against version number attacks, the root of DODAG first executes the startup and extraction algorithms, obtains its private key $S_{ID_{root}}$ and params system generic parameters, and publishes params, but it holds the mask master secret key confidential. Then, it executes the OffSign algorithm and obtains its offline signature. This step is performed only once. When the root of DODAG intends to increase the version number of DODAG, it must first sign and confirm the new version number through the OnSign algorithm. It then broadcasts the signed version number through the DIO message. As we know, the DIO message contains the sender ID, so the receiving nodes are able to execute the UnSign algorithm and authenticate signatures. At the same time as receiving the DIO message with an enhanced version number and the corresponding signature, the receiver node first confirms the signature. If this confirmation succeeds, this node updates the version number and releases the DIO message unchanged (including the signed version number); otherwise, the receiving node will understand that this is not a root update and will discard the DIO message. Therefore, if a malicious node intends to impersonate the root and illegally publish the increased version number because it does not know the private key of the root, it cannot sign the version number, and invade in this mode is not successful.

3.2. Weaknesses of Light Weight Defense Design

In this scheme, a bilinear pairing is e used to validate the message:

$$e\left(L, \frac{P}{P+P.H_1(ID)+P_{pub}}.Q_{ID}\right) \quad (3)$$

It should be noted that when using a bilinear pairing, both of its parameters must be a certain member of G1, which is an elliptic bending group here. However, it is not possible to calculate the second parameter as $\frac{P}{P+P.H_1(ID)+P_{pub}}.Q_{ID}$. In fact, in this regard, one point of the elliptical bend is divided into another point of the elliptical bend. It is impossible to Multiply and divide in the elliptical bending groups. In fact, the discrete logarithm theorem on the elliptic bending is based on this property. In addition, if there was a possibility of splitting into the elliptical bending group (as we know there is no possibility of splitting into an incremental group in the elliptical bending space), there is a possibility of the fake signature. In this case, the following method can be used to fake the signature:

The following will be part of the assumptions of the spoofing attack:

$$S_{ID} = \frac{P}{H_1(ID)+msk} \quad (4)$$

$$Q_{ID} = P.(H_1(ID) + msk) \quad (5)$$

$$e(S_{ID}.Q_{ID}) = e(P,P) \quad (6)$$

$$S = \frac{S_{ID}+P}{\lambda} \quad (7)$$

$$\Phi = \lambda.(x + h) \quad (8)$$

By proving the equality of the following bilinear pairing, we can start fake the signature and obtain the Φ and S parameters:

$$\begin{aligned} &e(S, \Phi.Q_{ID}) = e\left(\frac{S_{ID}+P}{\lambda}, \lambda.(x+h).Q_{ID}\right) \\ &= e((S_{ID} + P), (x+h).Q_{ID}) \\ &= e(S_{ID}, (x+h).Q_{ID}).e(P, (x+h).Q_{ID}) \\ &= e(P,P)^{(x+h)}.e\left(\frac{P}{\lambda}, \lambda.(x+h).Q_{ID}\right) e(S, \Phi.Q_{ID}) \\ &= e(P,P)^{(x+h)}.e\left(\frac{P}{\lambda}, \Phi.Q_{ID}\right) e(P,P)^{(x+h)} \end{aligned} \quad (9)$$

$$= \frac{e(S,\Phi \cdot Q_{ID})}{e\left(\frac{P}{\lambda},\Phi \cdot Q_{ID}\right)} = e\left(S - \frac{P}{\lambda}, \Phi \cdot Q_{ID}\right)$$

Now, according to how the signature is validated, the following equation is in place:

$$e(P,P)^{(x+h)} = e\left(L, \frac{P}{P+P.H_1(ID)+P_{pub}} \cdot Q_{ID}\right) = e\left(S - \frac{P}{\lambda}, \Phi \cdot Q_{ID}\right) \quad (10)$$

Now, with this equation, we can easily determine some values for Φ and S.

$$\Phi = \frac{P}{P+P.H_1(ID)+P_{pub}} \quad (11)$$

$$L = S.\Phi = S - \frac{P}{\lambda} \quad (12)$$

$$S = \frac{p}{\lambda \cdot (1-\Phi)} \quad (13)$$

It is enough for the intruder to produce two random numbers λ' and x'; then, calculate the two S' and Φ' given above equations. The value of r can also be calculated from the relation $r' = g^{x'}$. Finally, the value of $h = H_2(m', r', S')$ is generated for arbitrary text m' and sends triplex $\sigma = <h, \Phi', S'>$ to the other nodes. In addition, given the relation $P_{pub} = msk.P$ and the generality of the P_{pub} key, if it is possible to divide the elliptical bending group, it is possible to obtain msk, which is the main private key, there is a $msk = \frac{P_{pub}}{P}$. As a result, there will be no private parameters, and the signature will be easily forged. The following is a proposed algorithm for generating a new signature scheme that does not have the above design flaws and is resistant to spoofing attacks.

4. Proposed Signature Scheme

A digital signature scheme is commonly utilized used to authenticate messages on WSN and IoT networks. In a digital signature scheme, each node needs two key pairs (public and private) to sign and verify a message. This message can be in a protocol like the RPL DIO message that is sent from the root to other nodes and contains the location information and the corresponding DODAG version number. Identity-based cryptography is a primary approach for efficient key management, especially in large-scale networks such as wireless sensor networks and IoT devices with low computing power and energy. In an IBS scheme, the sender node utilizes his private key which he received from private key generator (PKG) to generating signature of the message. Then, he sends the message and its signature. Receivers retrieve the sender's public key by using his identity specifications. The proposed signature scheme is based on IBS. In addition, we present an approach to provide resistance against rank spoofing and version number attacks. Obviously, any leakage of key information compromises the overall security of the scheme. Therefore, it is important to note that the management and exchange of confidential keys are very important.

The proposed scheme consists of four phases: setup, key generation, signature generation, and validation:

Setup(k): This algorithm takes input from the security parameter k. In this phase two groups of the order q ($q > 2k$) denoted by G_1 and G_2 are generated. G_1 is an additive group and G_2 is a multiplicative group. Then, a bilinear pairing : $G_1 \times G_1 \rightarrow G_2$ and P, which is a generator of G_1, are chosen and computes $g = e(P,P)$, private key $msk \leftarrow Z_p^*$ and public key $P_{pub} = msk.P$. In addition, two hash functions $H_1: \{0,1\}^* \rightarrow Z_p^*$ and $H_2: \{0,1\}^* \times G_2 \rightarrow Z_p^*$ are chosen. Finally, this algorithm returns global parameter $gp = \{G_1, G_2, P, P_{pub}, e, H_1, H_2\}$.

Key generation ($KeyGen(gp, ID)$): The root of DODAG, known as the key generator, runs this algorithm. This algorithm takes the global parameter and the node's Identifier as input. The private key is calculated according to the formula $S_{ID} = \frac{P}{H_1(ID)+msk}$ and sent this private key to the node through a secure channel.

Signature Generation ($sign(sk, m)$): The sender runs this algorithm to sign the message and provide message and sender authentication. This algorithm takes the message and the sender's private key as input. While only the sender has his private key, only he has permission to run this algorithm. This algorithm chooses a random number x from Z_p^* and calculates $r = g^x$ and $S = (x+h).S_{ID}$, and finally returns the triplex $\sigma = <m, S, r>$.

Signature verification ($\mathbf{verify(pk, m, \sigma)}$): After receiving the <m, S, r>, the triples receiver runs this algorithm to ensure the integrity of the message. This algorithm takes the message, the signature, and the sender's public key as input. Then, it calculates the hash value $h = H_2(m, r)$ and checks if $g^h . r == e(S, H_1(ID) . P + P_{pub})$ holds; if the equality is satisfied, then it returns true.

5. Proposed Architecture

The proposed scheme introduces two scenarios to provide resistance against rank spoofing attacks. The first scenario is defined at the time of the initial formation of the DODAG graph. This scenario consists of three sections:

DODAG root section: Root executes the setup and key generation algorithms and obtains its own S_{ID} secret key and general parameters. Root calculates its rank and signs it using the signature algorithm. Finally, it publishes the DIO message containing the signed-rank.

First root children (first gp and DIO receivers): The root is chosen as the preferred parent. The nodes send their IDs to the root. The root executes the key generation algorithm for each node, calculates the private key, and sends this key privately to the node. The nodes calculate their rank and sign it using the signature-generating algorithm. The DIO message releases its signed rank, root, root ID, and general system parameters.

Other nodes receiving gp and DIO: According to the mentioned phases, other nodes gp received neighbor node's signed rank and its parent's signed-rank. The parent of node B is called node A. Therefore, they act as follows: By executing the verify algorithm, they will confirm the signature and obtain the rank of nodes B and A. If the signature is successfully confirmed and the rank of node B

is higher than node A, the process goes to the next step; otherwise a rank spoofing attack has occurred, and the received ranks are ignored. They select one of the neighboring nodes as the preferred parent and, through it, request the private key from the root; they calculate their rank and sign it through the signature-generating algorithm. The DIO message publishes their signed rank, preferred parent's signed rank, preferred parent's ID, and general parameters of the gp system.

In the second scenario, after the DODAG graph was formed and created, all the nodes had already computed their signature. They received the general parameters of the gap system and the private keys. Therefore, when a node receives the signed rank of neighbor node A and the signed rank of parent node B (node A), it acts as follows:

The node, by executing the verify algorithm, verifies the signature and obtains the rank of two nodes, B and A. As mentioned, node A is the parent of node B. If the confirmation is successful and the node B rank is higher than A, the process goes to the next step; otherwise, the rank spoofing attack occurs, and the received ranks are discarded. The node may change its preferred parent, calculating its rank and signing it using the signature-generating algorithm. The node then releases a DIO message containing its signed rank, the preferred parent's signed rank, and the preferred parent ID.

6. Analysis of Scheme

6.1. Proposed Scheme Integrity

To check the Integrity of the proposed scheme, it is sufficient to prove that if a valid signature is given the verify function $\text{verify}(\text{pk}, \text{m}, \sigma)$, the comparison answer will be positive. This case is proved below:

$$
\begin{aligned}
&\text{e}\left(\text{S}, \text{H}_1(\text{ID}) \,.\, \text{P} + \text{P}_{\text{pub}}\right) \\
&= \text{e}\left((\text{x}+\text{h}) \,.\, \text{S}_{\text{ID}}, \text{H}_1(\text{ID}).\text{P} + \text{P}_{\text{pub}}\right) \\
&= \text{e}\left((\text{x}+\text{h}) \,.\, \frac{\text{P}}{\text{H}_1(\text{ID})+\text{msk}}, \text{H}_1(\text{ID}).\text{P} + \text{msk}.\text{P}\right) \\
&= \text{e}\left(\frac{(\text{x}+\text{h})}{\text{H}_1(\text{ID})+\text{msk}} \,.\, \text{P}, (\text{H}_1(\text{ID}) + \text{msk}).\, \text{P}\right) \\
&= \text{e}(\text{P}, \text{P})^{\left(\frac{(\text{x}+\text{h})}{\text{H}_1(\text{ID})+\text{msk}}\right) . (\text{H}_1(\text{ID}) + \text{msk})} \\
&= e(P, P)^{(x+h)} = g^{x+h} = g^x \,.\, g^h = r \,.\, g^h
\end{aligned}
\tag{14}
$$

6.2. Security Analysis of the Proposed Scheme Concerning the Signature Security

Previously, the security of the proposed signature was proved. Given this case, as correctly mentioned by Nikravan et al. [28], the proposed scheme is safe. Because if the attacker node attempts to change the network version since it cannot sign for a new message (fake version) without having a private key, other nodes will find their claim false by using sent signature confirmation from the node. Given that the version is only produced by the root and the root is responsible for its signature, it cannot be spoofed without the private key of the root, and other nodes will notice the rank spoofing. Therefore, if the root's private key is not disclosed, it will not be possible to spoof the version.

The digital signature is also used to prevent rank spoofing attacks. Each node sends its rank plus its signature and its parent's rank plus its signature to the new node. The new node approves two incoming signatures (the first is the responsive node signature that is verified with the node's public key, and the second is the parent signature of the node that is verified with the public key of the parent node). In this way, the third node cannot change the ranks with the attack of the middleman because it is unable to spoof the signatures of nodes. After verifying two signatures, the node checks that the node rank must be higher than its parent node rank. Therefore, since the node cannot forge its parent's signature, it must choose a rating higher than its own rating; otherwise, it will be specified with its rank spoof in this review. As a result, the scheme is also resistant to rank spoofing attacks if the node's private key is not disclosed to others (especially their children).

6.3. Performance Analysis of the Proposed Scheme

In order to analyze the efficiency of the proposed scheme, the time required and energy consumption for each step must be calculated. To perform the signature production step, a hash and a multiplication of G_1 are required. In order to perform the validation step, a hash, a multiplication on G_1, multiplication, and exponention on G_2 as well as a bilinear pairing are also required. In the proposed scheme, the size of the variables is suggested as follows:

G_1 group of 542 bits ($|G_1|$ = 542 bit), G_2 group of 1084 bits ($|G_1|$ = 1084 bit), 80 bit id length ($|ID|$ = 80 bit) and size p, 252 bits (|p| = 542 bits) is assumed. In this proposed scheme, heavy operations related to computation are performed on a MICA2 processor. The result of this test is the time required for the relevant calculations as described in the Table 1:

Table 1. Time and energy required for the proposed signature scheme

	Multiplication over G_1	exponentiation over G_2	Bilinear pairing	Usage Type	Total
Sign	1	0	0	Time	0.81 s
				Energy	19.44 mJ
Verify	1	1	1	Time	3.61 s
				Energy	86.64 mJ

Table 1 shows the number of operations required along with the sum of energy and total time at each step for the proposed scheme. As can be seen in this table, a little time and energy are needed to create the signature that makes it work easily and quickly. At the same time, signature validation and verification requires more time and energy. It should be noted, however, that validation is also easily accomplished and takes little time.

Table 2 shows the number of operations required along with the sum of energy and total time at each step for the scheme [28]. As can be seen in this table, all the costly operations have taken place during the process of creating an offline signature. As a result, any number of signatures to be done will be fixed cost (the total cost of signatures is about 1.71 (s)). Therefore, the cost of signing is better than the proposed scheme of this thesis. However, the cost of validation is much higher than the proposed scheme. Given that the number of validations for a signature is more than that of creating a signature, it can be concluded that the proposed scheme is more efficient.

Table 2. Time and energy required for [20] signature scheme

	Multiplication over G_1	exponentiation over G_2	Bilinear pairing	Usage Type	Total
Offline Sign	1	0	0	Time	1.71 s
				Energy	41. 4 mJ
Online Sign	0	0	0	Time	0 s
				Energy	0 mJ
Verify	3	1	1	Time	5.23 s
				Energy	125.52 mJ

For example, to protect the version, the root produces the signature. Assuming the signature is already manufactured offline, this procedure will take no time. Now, the root sends the message and signature to the other nodes of the DODAG graph. If, for example, there are 100 other nodes in the network, the total energy consumed for signature validation is 12552mJ. Also, the total time required to sign is 5.23s. So, in this scenario, the proposed scheme of this paper would be both time and energy-efficient.

7. Conclusions

Due to the inherent features of the RPL protocol, how the nodes are ranked, and the version number mechanism to prevent loop formation, both rank and version number attacks are the most important ones on this protocol. In this paper, due to the importance of these attacks and the need to pay attention to defense against them, a digital signature algorithm has been used to prevent rank attacks and send version numbers to nodes. This digital signature design's primary and first goal is to prevent rank attacks and version numbers, it has been introduced as a better and more secure defense scheme than the Light Weight Defense scheme. It should also be borne in mind that since the proposed algorithm is a type of IBE algorithm known as a lightweight algorithm, so in terms of the amount of time and energy consumed, with the increase in the scale of a network, it is more efficient and optimum than before (due to the differences in the signature generation step of the two IBOOS and IBE algorithms). Compared to the VeRA scheme, it should also be noted that the updates are interdependent, and the attacker is aware that they can launch an attack. Also, one of the major problems associated with the TRAIL scheme has been scalability, which has been largely addressed in our proposed scheme.

7.1. Funding

We gratefully acknowledge the financial support from the Iran National Science Foundation (INSF) [Research project 97008930].

7.2. Author Contributions

Hamid Shabanipour and Reza Ebrahimi Atani conceived of the presented idea. Hamid Shabanipour developed the theory and performed the computations. Arian Arabnouri verified the analytical methods. Reza Ebrahimi Atani encouraged Hamid Shabanipour to investigate and supervise the findings of this work. All authors discussed the results and contributed to the writing of the final manuscript.

8. References

[1] Marietta, J., & Chandra Mohan, B. (2020). A Review on Routing in Internet of Things. Wireless Personal Communications, 111(1), 209–233. doi:10.1007/s11277-019-06853-6.

[2] Zrelli, A. (2022). Hardware, Software Platforms, Operating Systems and Routing Protocols for Internet of Things Applications. Wireless Personal Communications, 122(4), 3889–3912. doi:10.1007/s11277-021-09116-5.

[3] Ahmid, M., Kazar, O., & Barka, E. (2024). Internet of Things Overview: Architecture, Technologies, Application, and Challenges. Decision Making and Security Risk Management for IoT Environments. Advances in Information Security, vol 106. Springer, Cham, Switzerland. doi:10.1007/978-3-031-47590-0_1.

[4] Pongle, P., & Chavan, G. (2015). A survey: Attacks on RPL and 6LoWPAN in IoT. 2015 International Conference on Pervasive Computing (ICPC). doi:10.1109/pervasive.2015.7087034.

[5] Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., & Alexander, R. (2012). RPL: IPv6 Routing Protocol for

Low-Power and Lossy Networks (T. Winter & P. Thubert, Eds.). RFC Editor. doi:10.17487/rfc6550.

[6] Gaddour, O., & Koubâa, A. (2012). RPL in a nutshell: A survey. Computer Networks, 56(14), 3163–3178. doi:10.1016/j.comnet.2012.06.016.

[7] Li, C., Liu, Y., Xiao, J., & Zhou, J. (2022). MCEAACO-QSRP: A Novel QoS-Secure Routing Protocol for Industrial Internet of Things. IEEE Internet of Things Journal, 9(19), 18760–18777. doi:10.1109/JIOT.2022.3162106.

[8] Ahmmad, B. A., & Alabady, S. A. (2023). Energy-efficient routing protocol developed for internet of things networks. IET Quantum Communication, 4(1), 25–38. doi:10.1049/qtc2.12051.

[9] Mohseni, M., Amirghafouri, F., & Pourghebleh, B. (2023). CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic. Peer-to-Peer Networking and Applications, 16(1), 189–209. doi:10.1007/s12083-022-01388-3.

[10] Sahay, R., Geethakumari, G., & Mitra, B. (2021). A novel Network Partitioning Attack against Routing Protocol in Internet of Things. Ad Hoc Networks, 121. doi:10.1016/j.adhoc.2021.102583.

[11] Gali, S., & Nidumolu, V. (2022). An intelligent trust sensing scheme with metaheuristic based secure routing protocol for Internet of Things. Cluster Computing, 25(3), 1779–1789. doi:10.1007/s10586-021-03473-3.

[12] Pishdad, F., & Ebrahimi Atani, R. (2024). Prevention and detection of botnet attacks in IoT using ensemble learning methods. Biannual Journal Monadi for Cyberspace Security (AFTA), 13(2), 45-55.

[13] Nia, M. A., Atani, R. E., & Haghi, A. K. (2014). Ubiquitous IoT structure via homogeneous data type modelling. 7'th International Symposium on Telecommunications (IST'2014), 283–288. doi:10.1109/istel.2014.7000715.

[14] Baek, J., Newmarch, J., Safavi-Naini, R., & Susilo, W. (2004). A survey of identity-based cryptography. Australian Unix Users Group Annual Conference, 1-3 September, 2004, Melbourne, Australia.

[15] Bösch, C., Hartel, P., Jonker, W., & Peter, A. (2014). A survey of provably secure searchable encryption. ACM Computing Surveys, 47(2), 1–51,. doi:10.1145/2636328.

[16] Ananna, T. N., & Saifuzzaman, M. (2024). Introduction to Internet of Things. Studies in Computational Intelligence, 1169, 1–49. doi:10.1007/978-981-97-5624-7_1.

[17] Levis, P., Clausen, T., Hui, J., Gnawali, O., & Ko, J. (2011). The Trickle Algorithm. RFC Editor. doi:10.17487/rfc6206.

[18] Dvir, A., Holczer, T., & Buttyan, L. (2011). VeRA - Version Number and Rank Authentication in RPL. 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems. doi:10.1109/mass.2011.76.

[19] Landsmann, M., Wahlisch, M., & Schmidt, T. (2013). Topology Authentication in RPL. 2013 IEEE Conference on Computer Communications Workshops (INFOCOM Wkshps). doi:10.1109/infcomw.2013.6970745.

[20] Stephen, R., & Arockiam, L. (2018). RIAIDRPL: Rank increased attack (RIA) identification algorithm for avoiding loop in the RPL DODAG. International Journal of Pure and Applied Mathematics, 119(16), 1203-1210.

[21] Tandon, A., & Srivastava, P. (2019). Trust-based Enhanced Secure Routing against Rank and Sybil Attacks in IoT. 2019 Twelfth International Conference on Contemporary Computing (IC3), 1–7. doi:10.1109/ic3.2019.8844935.

[22] Hashemi, S. Y., & Shams Aliee, F. (2019). Dynamic and comprehensive trust model for IoT and its integration into RPL. Journal of Supercomputing, 75(7), 3555–3584. doi:10.1007/s11227-018-2700-3.

[23] Karmakar, S., Sengupta, J., & Bit, S. Das. (2021). LEADER: Low Overhead Rank Attack Detection for Securing RPL based IoT. 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS) 2021, 429–437. doi:10.1109/COMSNETS51098.2021.9352937.

[24] Muzammal, S. M., Murugesan, R. K., Jhanjhi, N. Z., & Jung, L. T. (2020). SMTrust: Proposing Trust-Based Secure Routing Protocol for RPL Attacks for IoT Applications. 2020 International Conference on Computational Intelligence, ICCI 2020, 305–310. doi:10.1109/ICCI51257.2020.9247818.

[25] Fatima-tuz-Zahra, Jhanjhi, N., Brohi, S. N., Malik, N. A., & Humayun, M. (2020). Proposing a Hybrid RPL Protocol for Rank and Wormhole Attack Mitigation using Machine Learning. 2020 2nd International Conference on Computer and Information Sciences (ICCIS), 1–6. doi:10.1109/iccis49240.2020.9257607.

[26] Pishdar, M., Seifi, Y., Nasiri, M., & Bag-Mohammadi, M. (2022). PCC-RPL: An efficient trust-based security extension for RPL. Information Security Journal, 31(2), 168–178. doi:10.1080/19393555.2021.1887413.

[27] Belavagi, M. C., & Muniyal, B. (2020). Multiple intrusion detection in RPL based networks. International Journal of Electrical and Computer Engineering, 10(1), 467–476. doi:10.11591/ijece.v10i1.pp467-476.

[28] Nikravan, M., Movaghar, A., & Hosseinzadeh, M. (2018). A Lightweight Defense Approach to Mitigate Version Number and Rank Attacks in Low-Power and Lossy Networks. Wireless Personal Communications, 99(2), 1035–1059. doi:10.1007/s11277-017-5165-4.