future internet

Article

Realization of Authenticated One-Pass Key Establishment on RISC-V Micro-Controller for IoT Applications

Tuan-Kiet Dang *, Khai-Duy Nguyen, Binh Kieu-Do-Nguyen, Trong-Thuc Hoang and Cong-Kha Pham

Department of Computer and Network Engineering, University of Electro-Communications (UEC), 1-5-1 Chofugaoka, Tokyo 182-8585, Japan; khaiduy@vlsilab.ee.uec.ac.jp (K.-D.N.); binh@vlsilab.ee.uec.ac.jp (B.K.-D.-N); hoangtt@uec.ac.jp (T.-T.H.); phamck@uec.ac.jp (C.-K.P.)

* Correspondence: tuankiet@vlsilab.ee.uec.ac.jp

Abstract: Internet-of-things networks consist of multiple sensor devices spread over a wide area. In order to protect the data from unauthorized access and tampering, it is essential to ensure secure communication between the sensor devices and the central server. This security measure aims to guarantee authenticity, confidentiality, and data integrity. Unlike traditional computing systems, sensor node devices are often limited regarding memory and computing power. Lightweight communication protocols, such as LoRaWAN, were introduced to overcome these limitations. However, despite the lightweight feature, the protocol is vulnerable to different types of attacks. This proposal presents a highly secure key establishment protocol that combines two cryptography schemes: Elliptic Curve Qu–Vanstone and signcryption key encapsulation. The protocol provides a method to establish a secure channel that inherits the security properties of the two schemes. Also, it allows for fast rekeying with only one exchange message, significantly reducing the handshake complexity in low-bandwidth communication. In addition, the selected schemes complement each other and share the same mathematical operations in elliptic curve cryptography. Moreover, with the rise of a community-friendly platform like RISC-V, we implemented the protocol on a RISC-V system to evaluate its overheads regarding the cycle count and execution time.

Keywords: IoT; secure communication; key establishment; signcryption; ECQV; RISC-V

Citation: Dang, T.-K.; Nguyen, K.-D.; Kieu-Do-Nguyen, B.; Hoang, T.-T.; Pham, C.-K. Realization of Authenticated One-Pass Key Establishment on RISC-V Micro-Controller for IoT Applications. *Future Internet* **2024**, *16*, 157. https://doi.org/10.3390/fi16050157

Academic Editors: Christos Tryfonopoulos and Nicholas Kolokotronis

Received: 26 March 2024
Revised: 21 April 2024
Accepted: 1 May 2024
Published: 3 May 2024

1. Introduction

In modern society, the utilization of smart devices and services has become more prevalent. These encompass a vast range of industries, including home automation, manufacturing, medicine, finance, and transportation. The Internet of things (IoT) refers to the billions of interconnected devices that are equipped with sensors and actuators that can be accessed from virtually any location in the world through the Internet. A wireless sensor network (WSN), which is a subset of an IoT network, is a network that incorporates a large number of wireless sensor nodes, wireless communication, and data processing functions to monitor, gather, and transmit information from the environment. In contrast to the Internet, the construction of wireless sensor networks relies on low-complexity communication protocols. Low-power and wide-area communication networks, such as Zigbee and LoRa, are preferable for the development of WSNs, as they satisfy the constrained conditions of small sensor nodes. As the wireless network continues to expand, these nodes are scattered over a large area and are prone not only to physical attacks but also to cyberattacks. Therefore, it is crucial to provide security measures for WSNs just as much as the security measures for the Internet.

A typical wireless sensor node is equipped with a sensing device, a wireless transceiver, a power management unit, and a central computing unit. The node devices in a WSN are inherently resource constrained, especially in batteryless and energy-harvesting applications: they have limited processing speed, storage capacity, and communication bandwidth. In a typical network, the sensor nodes communicate between themselves using radio signals. The constraints limit the integration of the high-security standard protocols used among modern computers, like SSL (secure sockets layer) or TLS (transport layer security). As mentioned, long range (LoRa), which is a widely adopted network in low-power applications, is a radio modulation technology that is capable of offering long-range, low-power, and secure data transmission for IoT devices. The LoRaWAN MAC protocol specifies the regulations for LoRa and defines the standards for LoRa networking. However, despite the lightweight feature, the LoraWAN protocol still faces significant threats from malicious parties [1–3]. Han and Wang [4] indicated that the flaws of LoRaWAN lie in the key management and data confidentiality. Lacking a mechanism for session key establishment enables long-term leakage of the root key due to side-channel attacks in encryption and decryption with the AES cipher block. Such potential weaknesses of LoRaWAN outweigh the lightweight feature of applying the symmetric AES-128 cipher. Several attempts were proposed to improve LoRaWAN's security measures. Naoui et al. [5] proposed applying the concept of the session key to update the secret key every session of communication, thus enhancing the secure communication channels between the end node and the network server. Han and Wang [4] proposed applying a lightweight Rabbit cipher in the key derivation function to the root key update scheme, and hence, strengthening the security of session keys.

The lightweight feature of symmetric key cryptography is an essential property that a key establishment (KE) protocol for WSNs should possess. Other proposals also tried to benefit from symmetric key cryptography and add more security properties to establish secure communication channels. For example, Pu et al. [6] proposed a lightweight KE protocol that provides an authentication property. The protocol applied a Tinkerbell chaotic map, a physical unclonable function (PUF), and a one-way hash function to provide an authentication session key agreement. Similarly, Zheng and Chang [7] proposed a mutual authentication protocol that also relies on a hash and a PUF primitive for the key exchange between IoT endpoints. These methods and the LoRaWAN share a common principle that exchanging messages in the public domain requires a cryptographic hash function with optimal collision resistance properties. In other words, the auxiliary materials for the key establishment are hashed and then sent to other parties through the handshake. The confidentiality of the materials is guaranteed by the hash function. Researchers have also tried to achieve a lightweight feature for secure communication of IoT devices with stronger cryptography.

Highly secure cryptography, such as public key cryptography (PKC), has been known to be expensive due to the limited computing resources of sensor nodes. The traditional Diffie–Hellman key exchange used in modern computers requires an authentication feature to prevent man-in-the-middle attacks. Torres et al. [8] indicated that IoT devices cannot afford the significant computational cost. Additionally, node devices need to update their public and private keys to prevent ciphertext-only attacks similar to the root key revelation of LoRaWAN. On the other hand, Keoh et al. [9] suggested that a careful selection of settings for public-key-based ciphersuites would ensure the security of the IoT network and reduce overheads in the handshake. Sciancalepore et al. [10] claimed that public key cryptography has become feasible on the latest generation of constrained devices and supported this statement by proposing an elliptic curve cryptography (ECC)-based protocol for IoT devices. Moreover, other than LoRaWAN, recent works for highly sensitive data transmission (healthcare, e-payment) applied PKC with various IoT communication technologies, like ZigBee, RFID, and NFC. A. Rehman et al. [11] utilized the Zigbee wireless technology to establish a reliable network management (RNM-SC) for a smart healthcare system. A group of sensor nodes are managed by a local coordinator and connected to a ZigBee router. The system offers an optimization in the transmission path, which is handled by the router, and a node table comprising information on close-distance nodes. The underlying key establishment to ensure authenticity relies on the station-to-station algorithm, which is based on the classic Diffie–Hellman method. Furthermore, RNM-SC applies Blum–Goldwasser asymmetric cryptography to encrypt and decrypt sensitive data. An extensive review by I. E. Gaabouri [12] addressed the security challenges and threats of systems based on RFID and NFC technologies. It is suggested that the deployment of a strong cryptography scheme is a feasible solution to reinforce RFID's communication security. The review also summarized the most recent studies on cryptography schemes for RFID and NFC and showed a diversity of encryption methodologies for solutions implemented for different applications. Based on the statistical data, the author suggested that cryptography systems like ECC are preferable for semi-active and active tag devices. Meanwhile, a lightweight scheme is favorable for passive tag devices where resource limitations are more strict. In addition, many recent works shared the same methodology of integrating strong cryptography, like ECC, to secure communication in IoT networks [13–15].

Our proposal applies two schemes that can be constructed by using ECC. When constructing a protocol that uses ECC-based handshakes for node devices with small processors and low-bandwidth transceivers, there are several important factors to consider. First, a key establishment protocol that is proposed under a specific operating assumption must ensure security properties, such as confidentiality and authentication. Second, the number of computations executed at the node side is a critical issue that should be addressed. Third, the size and number of exchange messages should be reduced for low-bandwidth communication. With limited power and energy on a batteryless or self-charging system on an IoT node, the second and the third factors should be carefully addressed to improve the power allocation between the processing unit and the transceivers, as well as lengthen the operation of the whole system under limited power conditions.

Based on these speculations, this paper proposes three major contributions:

- This research was the first to propose utilizing a signcryption key encapsulation mechanism in composition with an Elliptic Curve Qu–Vanstone (ECQV) implicit certificate scheme [16] to construct a two-level key establishment protocol. By applying a hybrid cryptosystem, the proposed protocol preserves the security characteristics of both schemes and takes advantage of both public key cryptography and symmetric cryptography. Depending on the demand of the security level, the protocol, at the very least, can enable a fast rekeying method of session keys derived from a generated key pair, allowing for sending session key generation materials and encrypted sensor data in one message exchange. At a higher level of security, the protocol allows for updating the public and private key pair; hence, only several sessions share a key pair to derive the session keys. At the highest security level, every communication session can issue a fresh key pair and session key to communicate.
- Under a two-party communication model, our approach integrates a physical unclonable function to the ECQV scheme to provide additional authentication data to the handshake. This lightweight solution enhances the ECQV scheme and allows for only authenticated node devices to request a key pair. Thus, the modified ECQV scheme can protect against common attacks, such as impersonation attacks and replay attacks.
- We implemented a 32-bit RISC-V system that is suitable for a resource-constrained sensor node to assess the computational overheads of the proposed protocol. The software version of the protocol was tested on different configurations of the RISC-V processor to evaluate the timing cost. Moreover, instead of using software-based or pseudo-security primitives, the RISC-V system is equipped with standard physical security primitives, including a true random number generator (TRNG) and a PUF.

The remainder of this paper is organized as follows: Section 2 presents the background of the paradigm of a hybrid cryptosystem and the selected schemes. It also summarizes the mathematical notions used in the proposed protocol. Section 3 introduces the assumption of the system model for an IoT network and the combination of the ECQV implicit certificate scheme and the one-pass signcryption key exchange mechanism. This section also describes step-by-step operations in the proposed protocol. Section 4 analyzes the protocol's security features at different levels of security. Section 5 compares the communication and computation cost of the proposed protocol with related works that also support authentication property. Section 6 discusses the protocol's advantages and limitations and outlines the directions for future work. Finally, section 7 concludes the paper.

2. Preliminary Background

This section briefly reviews the ideas of a hybrid cryptosystem, ECQV implicit certificate scheme, and signcryption key establishment mechanism. In addition, a summary of the mathematical notions used in the proposed protocol is provided.

2.1. Hybrid Cryptosystem

A key establishment protocol aims to allow two participants who have their own source of randomness and need to establish a common secret. The output shared secret must be the same value for the two participants, and it has to be unknown to any outsider. The proposed key establishment protocol is designed based on a hybrid cryptosystem, which was formalized by Cramer and Shoup [17]. The hybrid construction paradigm, also known as the KEM-DEM construction, consists of two lower-level building blocks: a key encapsulation mechanism (KEM) and a data encapsulation mechanism (DEM). Generally, a KEM is a method by which two participants exchange their random contributions to the generation of the shared secret in a protected manner. Specifically, a KEM takes a private and public key pair to generate a random symmetric key and its encapsulation for the handshake. Meanwhile, a DEM properly encrypts a message payload using the generated symmetric key (the shared secret) with a symmetric block cipher. Fundamentally, a KEM consists of three operations:

- A key generation operation *KEM.KeyGen* that outputs a public and private key pair. The generation of a key pair depends on the chosen underlying algorithm.
- An encapsulation operation *KEM.Encrypt* that takes a key pair of the sender, and a public key of the receiver, and outputs a shared key K and an encapsulated text $C1$.
- A decapsulation operation *KEM.Decrypt* that takes a key pair of the receiver, a public key of the sender, and an encapsulated text $C1$, and extracts the shared key K if the operation is valid.

A data encapsulation mechanism (DEM) is either an encryption or a decryption operation of a chosen block cipher. Figure 1 describes the model of a KEM and DEM in a hybrid cryptosystem.

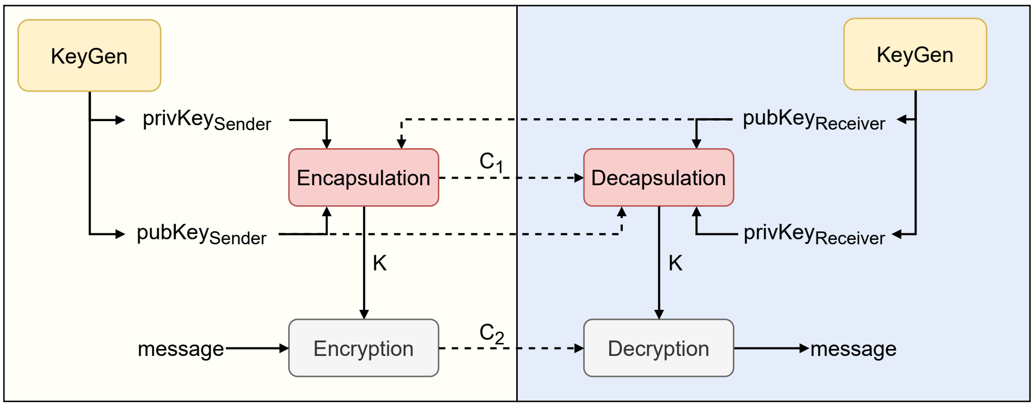

Figure 1. Hybrid KEM-DEM paradigm.

This hybrid paradigm takes advantage of the speed and efficiency of symmetric cryptography in the encryption and decryption of an arbitrary length payload. Moreover, the hybrid model relies on highly secure public key cryptography to generate a unique symmetric key for communication. This modular design approach also has a primary benefit in that the security requirements of the asymmetric and symmetric parts of the scheme can be completely independent of each other. Thus, it allows for freedom in the selection of the underlying algorithms for each part to construct a protocol that suits a specific application [18,19]. An example of a cryptosystem based on the Diffie–Hellman (DH) method that applies an AES cipher block as the data encapsulation can be found in [20]'s specification. Note that these selections for a hybrid cryptosystem offer security to an extent under a particular security assumption; thus, it is still vulnerable to some attacks, such as key compromise impersonation attacks. This research aimed to utilize the modular characteristic of this hybrid model to construct a key establishment protocol. The proposed protocol applies a modified version of a KEM with signcryption cryptography. Also, the ECQV implicit signcryption scheme is used as the $KEM.KeyGen$ operation to reduce the cost of public key cryptography as opposed to a DH-based KEM in the specification [20].

2.2. ECQV Implicit Certificate Scheme

As mentioned above, the KEM requires a public–private key pair before the encapsulation operation. In public key cryptography, the Diffie–Hellman key exchange is the common method for two parties to generate their key pairs and exchange the public parts. This key exchange requires explicit signatures from both participants to authenticate the exchange materials. However, generating these certificates can be computationally intensive, which is not ideal for IoT node devices. The ECQV scheme leverages the mathematical properties of ECC to associate IoT nodes' identities with their public keys, forming implicit certificates. This characteristic reduces the computational effort on authentication from one party in the communication channels in contrast to traditional public key cryptography methods, such as RSA, ECDSA, or EdDSA, which require both parties to compute and directly verify each other's certificates. In the context of a WSN, the ECQV scheme provides a passive approach, ensuring that only the deployed sensor nodes can derive the correct key pair from the implicit certificate issued by the trusted server. This scheme effectively mitigates the computational overhead associated with the key pair generation and authentication on the node devices. Moreover, the scheme reduces the quantity and size of the exchange messages in the network.

2.3. Signcryption

Signcryption is a public key authenticated encryption introduced by Zheng [21]. The key idea behind signcryption is to integrate encryption and signature operations into a single computation, thereby reducing the computational overhead and improving the efficiency compared with performing the operations separately. The cryptography primitive aims to simultaneously achieve confidentiality, integrity, and authentication at a lower cost. Dent [22–24] extended the principles of signcryption to the key encapsulation mechanism and provided security definitions for the integration. The resulting signcryption KEM (SKEM) became a variant of the signcryption primitive capable of generating authenticated symmetric keys. The characteristics of SKEM motivated the exploration of the connection between SKEM and one-pass key establishment (OPKE) by Gorantla et al. [25]. Based on the one-pass HQMV scheme [26], Gorantla et al. proposed SKEM-based OPKE, which provides a highly efficient method to generate ephemeral symmetric keys with confidentiality and authenticity. This study aimed to realize the practicality of OPKE based on SKEM to meet the demands of wireless sensor networks.

The proposed protocol applies elliptic curve cryptography as the mathematical foundation of both the ECQV and SKEM schemes. Therefore, the parameters of a selected curve can be shared for the two schemes. For a specific curve, let $\mathcal{G}$ be an elliptic curve group, and let $G \in \mathcal{G}$ be a generator of order (prime) q of the group $\mathcal{G}$. The ECQV requires a hash modulo function $\mathcal{H}_1$, which computes the hash and then performs modulo to the order q. The function hash without modulo is denoted as $\mathcal{H}_2$. In addition, the ECQV algorithm is added with a function of a PUF with a challenge–response pair (CRP) denoted as $(\mathcal{C}^t, \mathcal{R}^t)$. Table 1 shows the notions used in the two schemes.

Table 1. Symbols and corresponding meanings.

	Symbols	Corresponding Meaning
Parameters and variables	ID	The identity
	G	The base point of ECC
	q	A prime order of a specific elliptic curve
	$(\mathcal{C}^t, \mathcal{R}^t)$	A challenge $\mathcal{C}^t$ and its corresponding response $\mathcal{R}^t$ at time t
	t, t_n, t_s	Generated random values in $[1, q-1]$
	R, R_n, R_s	Curve points generated from random values
	$Cert_N$	Implicit certificate designated for a node with ID_N
Keys	(k_S, K_S)	A private–public key pair of the server
	(k_N, K_N)	A private–public key pair of a node with ID_N
	K	A symmetric session key
Functions	$\mathcal{H}_1()$	Hash modulo function
	$\mathcal{H}_2()$	Hash function
	$F_{PUF}()$	Function of a PUF
	$Enc()$	Encryption algorithm using the symmetric key K
	$Dec()$	Decryption algorithm using the symmetric key K

3. Proposed Key Exchange Protocol

3.1. Security Model Assumption

The proposed protocol was designed for a wireless sensor network model consisting of an array of IoT sensor nodes attempting to communicate with a trusted server. Each sensor node is assigned an identity ID. The server is also assigned with an ID_S. Some IoT applications, such as medical wearable sensor devices, are required to provide an anonymity property (hiding the true ID of a node device) to the protocol. However, the anonymity property was out of the scope of this study. The server is a legitimate entity responsible for gathering sensor data and distributing materials for the generation of public–private key pairs. In the trusted manufacture domain, the server generates its asymmetric key pair (k_S, K_S). The server's private key k_S must be securely stored on the server side. The server's public key K_S and its identity ID_S are distributed to all sensor nodes in the network before deployment. The proposed protocol implements a physical unclonable function entity in each sensor node. Therefore, the server is required to store the challenge–response pairs $(\mathcal{C}^t, \mathcal{R}^t)$ of all sensor nodes in its database.

3.2. Practical Application of ECQV and One-Pass SKEM Methodologies

Wireless sensor networks are subject to constraints, particularly regarding power and energy consumption, which can range from hundreds of milliwatts to the microwatt scale. We aimed to construct a protocol that supports low-power WSNs while providing sufficient security properties. To achieve this goal, the optimal approach is to reduce the computation on the node devices by applying symmetric block ciphers, such as AES, ASCON, and TEA, which offer a lightweight encryption and decryption process. Furthermore, the limitation in computation was also considered in the key generation process, and SKEM is a suitable scheme to generate and transport the key generation material at the least cost. By definition, the SKEM scheme requires asymmetric key pairs to generate symmetric session keys. Instead of applying the classic Diffie–Hellman method to generate and exchange generation materials for key pairs, the ECQV scheme significantly reduces the computing cost of authentication. As a result, a combination of ECQV and SKEM provided an optimal solution to reduce the computing effort at the node side while still applying strong asymmetric cryptography to establish secure communication.

The idea of the proposed protocol was to utilize the advantages that SKEM offers:

- The session key generation with signcryption gives assurance to the sender that the session key is available only to the recipient and assurance to the recipient that the key came from the sender.

- One-pass key establishment provides a very efficient construction of session keys, which reduces the complexity of the handshake and computation at the sensor nodes.
- The encrypted sensor data can be transmitted simultaneously with key generation materials, which favors the intermittent operation of the sensor nodes.

According to Okamoto [27], it is important for one-pass key establishment protocols to have the security goal of key compromise impersonation. This is also true for OPKE based on SKEM, where only the sender contributes ephemeral data to the symmetric key generation. An adversary who can expose the key pair of the sender can permanently impersonate the sender. Therefore, the practical application of the SKEM-based OPKE relies completely on the secrecy of the asymmetric key pair. Moreover, in the deployment domain, the SKEM scheme publicly exposes the sender's identity ID in the exchange message. Without the existence of a trusted third party (a public key infrastructure) to generate key pairs for the sender and the receiver, any adversary can claim to possess that same ID and establish a shared key with the recipient. In order to lower the complexity of a WSN, the server and the sensor nodes need a method to update the key pairs at the node to mitigate the risk of impersonation.

The ECQV implicit certificate scheme is a low-cost solution to this problem. The authenticity of a key pair yielded by the ECQV scheme adds another layer of assurance to the server that the generated session key is bound to not only the identity of a sensor node and the server but also the key pair designated for that sensor node. The sensor nodes can benefit from the low cost in the size of the exchange message and relieve themselves from the burden of signing and verifying explicit signatures. However, the deployment of an ECQV scheme faces the same issue due to the revelation of the sensor nodes' identities ID_N. In the deployment domain, the ECQV scheme can be triggered by a malicious party claiming to have ID_N. The implicit certificate is bound to the identity, not the physical device that has been assigned with ID_N. From the server's perspective, it can only recognize the identity that is trying to make a request for a certificate stored in the database. Therefore, the proposed protocol resolves this problem by binding the implicit certificate with the identity ID_N and a unique characteristic of an actual physical node device. A physical unclonable function implemented inside of the processor of a sensor node can provide distinctive responses that characterize a sensor node from each other. Furthermore, it is worth noticing that the ECQV scheme allows the server to compute a sensor node's public key without an additional exchange message from the sensor node. The modified ECQV with a PUF makes the public key somewhat "private" when using the response in the computation. In addition, from the standpoint of a sensor node, a matching challenge–response pair guarantees that it is communicating with the authentic server.

The realization of SKEM-based OPKE is beneficial for a constrained WSN. The proposed protocol provides the intermittent operation of batteryless or self-charging IoT nodes with higher secure communication channels. By applying the ECQV scheme as the key pair generation method, and a PUF as an enhancement to the ECQV scheme, a WSN can benefit from the advantages of SKEM-based OPKE with minimal risks.

3.3. Protocol Proposal

This study's primary contribution was the construction of a two-level security protocol when combining ECQV and SKEM schemes to establish secure communication channels in an IoT network. Figure 2 describes the proposed protocol. The two-level security notion implies the dependency of the generation of a session key on only a SKEM or both SKEM and ECQV scheme.

A level 1 secure communication channel, whose security property depends solely on the SKEM scheme, is established when a generated node's key pair (k_N, K_N) is used repetitively to create fresh session keys K. This mechanism enables quick session key generation at the cost of only one exchange message. A feasible application of SKEM requires assumptions on secure storage for ephemeral data and the key pair on the node device to avoid long-term compromises. To eliminate the risk of exposing or compromising the random source and the key pair stored on a node device, a level 2 secure channel is created with a unique and fresh session key using a newly generated node's key pair. At this level, the handshake to generate a session key requires three exchange messages to sequentially execute both ECQV and one-pass SKEM schemes. An alternative use case is that periodically updating the key pair of a node can minimize the risk of temporary leakage of a key pair when using only the SKEM scheme and benefit from the low exchange message count.

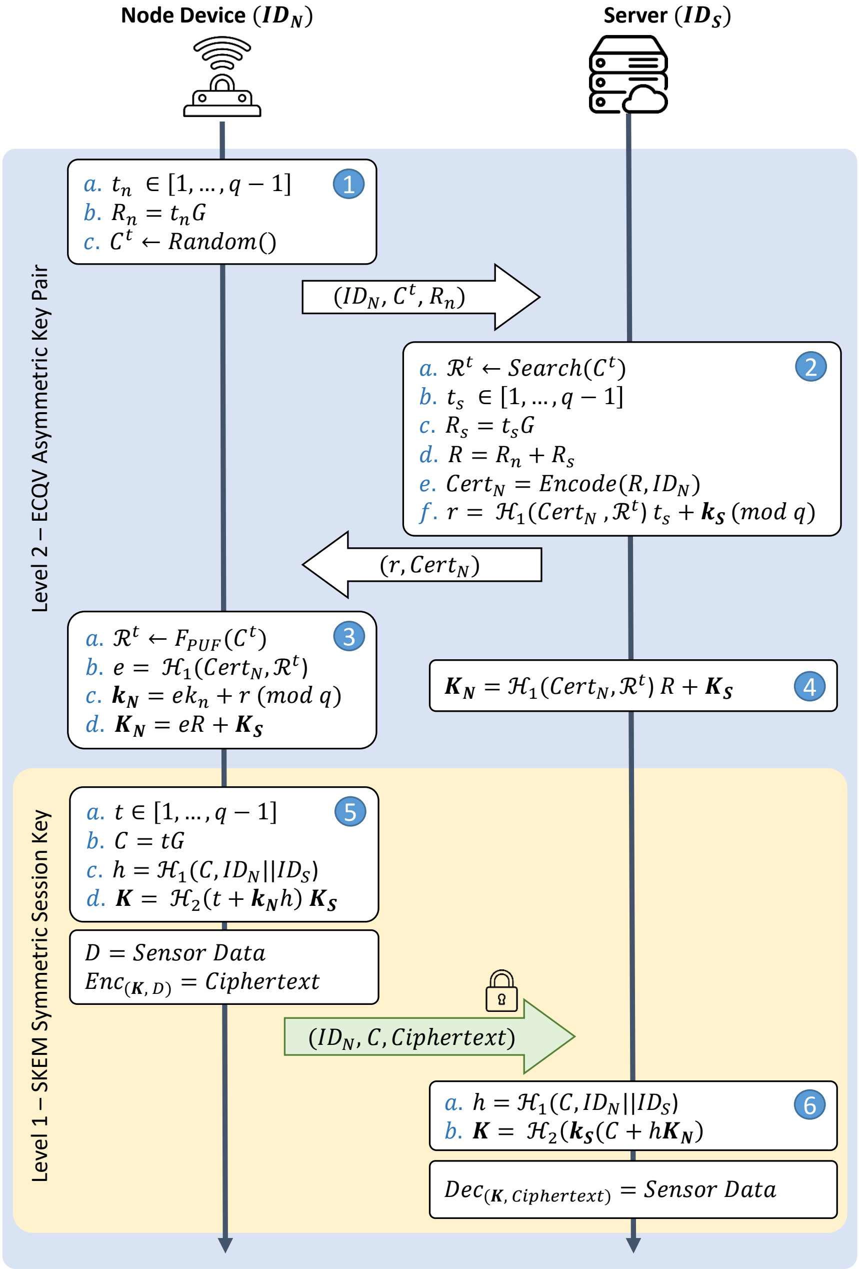
Node Device $(\boldsymbol{ID_N})$
Server $(\boldsymbol{ID_S})$
Level 2 – ECQV Asymmetric Key Pair
1
a. $t_n \in [1, \ldots, q-1]$
b. $R_n = t_n G$
c. $C^t \leftarrow Random()$
(ID_N, C^t, R_n)
2
a. $\mathcal{R}^t \leftarrow Search(C^t)$
b. $t_s \in [1, \ldots, q-1]$
c. $R_s = t_s G$
d. $R = R_n + R_s$
e. $Cert_N = Encode(R, ID_N)$
f. $r = \mathcal{H}_1(Cert_N, \mathcal{R}^t)\, t_s + \boldsymbol{k_S}\, (mod\ q)$
$(r, Cert_N)$
3
a. $\mathcal{R}^t \leftarrow F_{PUF}(C^t)$
b. $e = \mathcal{H}_1(Cert_N, \mathcal{R}^t)$
c. $\boldsymbol{k_N} = ek_n + r\ (mod\ q)$
d. $\boldsymbol{K_N} = eR + \boldsymbol{K_S}$
4
$\boldsymbol{K_N} = \mathcal{H}_1(Cert_N, \mathcal{R}^t)\, R + \boldsymbol{K_S}$
Level 1 – SKEM Symmetric Session Key
5
a. $t \in [1, \ldots, q-1]$
b. $C = tG$
c. $h = \mathcal{H}_1(C, ID_N || ID_S)$
d. $\boldsymbol{K} = \mathcal{H}_2(t + \boldsymbol{k_N} h)\, \boldsymbol{K_S}$
$D = Sensor\ Data$
$Enc_{(\boldsymbol{K}, D)} = Ciphertext$
$(ID_N, C, Ciphertext)$
6
a. $h = \mathcal{H}_1(C, ID_N || ID_S)$
b. $\boldsymbol{K} = \mathcal{H}_2(\boldsymbol{k_S}(C + h\boldsymbol{K_N})$
$Dec_{(\boldsymbol{K}, Ciphertext)} = Sensor\ Data$

Figure 2. 2-level security key establishment protocol. The computations are grouped into six steps, each indicating sequential operations at the sender and receiver.

3.3.1. Level 2: ECQV Asymmetric Key

The establishment of a secure communication channel starts with the ECQV handshake modified with the PUF function. An implicit certificate is issued by a node device as follows:

- Step 1—Certificate request: A node generates a random number $k_n \in \{0, q\}$ and calculates the corresponding elliptic curve point using $R_n = k_n G$. The random number

k_n must be securely stored at the node side; then, the point R_n, the node's identity ID_N, and a randomly chosen challenge $\mathcal{C}^t$ are sent to the server as a certificate request.

- Step 2—Certificate generation: After receiving the request, the server generates another random number and its corresponding point (k_s, R_s). Then, the implicit certificate is generated through hashing an elliptic curve point $R = R_n + R_s$. Upon storing the challenge–response pair, the server computes the implicit signature r with the corresponding response $\mathcal{R}^t$ of the received challenge $\mathcal{C}^t$.

$$r = \mathcal{H}_1(Cert_N, \mathcal{R}^t)k_s + k_S(mod\, q) \tag{1}$$

- Step 3—Key extraction: After receiving the implicit certificate, the node generates the response $\mathcal{R}^t$ from the previously selected challenge $\mathcal{C}^t$. Afterward, it extracts its designated key pair (k_N, K_N) using $\mathcal{R}^t$.

$$e = \mathcal{H}_1(Cert_N, \mathcal{R}^t) \tag{2}$$

$$k_N = ek_n + r(mod\, q) \tag{3}$$

- Step 4—The server can compute the public key of the node with Equations (2) and (3) by itself, thus reducing the cost of a message from the node to share its public key.

As mentioned earlier, depending on the security requirement of IoT applications, the ECQV handshake will be executed to provide the level 2 security channel. Otherwise, the node only executes the SKEM handshake to exchange data with the lowest cost.

3.3.2. Level 1: One-Pass SKEM Symmetric Session Key

In order to establish a level 1 secure channel, the node and the server execute the SKEM scheme in two main steps:

- Step 5—Key encapsulation: The node device generates a random point C on the elliptic curve. Then, the encapsulation of the random point C and the IDs of the node and the server are computed:

$$h = \mathcal{H}_1(C, (ID_N||ID_S)) \tag{4}$$

The session key is created at the node side using the encapsulation, the node's private key, and the server's public key. The session key is computed as follows:

$$K = \mathcal{H}_2((t + k_N h)K_S) \tag{5}$$

With the session key, the node can encrypt its data and send it to the server. Thanks to the one-way design of SKEM, the node can combine the ciphertext and the materials to generate the same session key into a compound message. Then, it transmits the message to the server.

- Step 6—Key decapsulation: After receiving the information, the server retrieves each segment and starts computing the session key:

$$K = \mathcal{H}_2(k_S(C + hK_N)) \tag{6}$$

After the server decrypts the ciphertext, successful communication is guaranteed. A meaningful message indicates that the communication is successful; otherwise, an adversary may have compromised the node. If this is the case, the server can force an ECQV key pair update or wait until the node's key pair expires.

4. Security Analysis

This section presents a security analysis of the proposed protocol. The analysis was conducted by applying the informal (non-mathematical) methodology to provide a direct interpretation of the supported and non-supported security properties of the protocol. According to the criteria of the secure authentication and key agreement scheme, it satisfies several security properties and can defend against common attacks.

(a) *Mutual authentication*: In the level 1 secure channel, signcryption assures that only the intended participants with valid identities can extract the correct session key K. The modified ECQV scheme guarantees the server that only the sensor node having ID_N and a valid CPR can derive the correct key pair (k_N, K_N) from the implicit certificate. On the other hand, knowing the server is the only one having stored its CRPs, the sensor node can be certain that only the server (with its valid private key k_S) can derive the correct session key K. Therefore, the proposed protocol can achieve mutual authentication even for level 1 security communication channels.

(b) *Integrity*: Both parties using ECQV and SKEM schemes verify their knowledge of a key pair and a symmetric key after successfully using the keys. Thus, the content of the exchange messages can be verified after meaningful communication is conducted. Any modification of the messages will result in invalid keys and meaningless decrypted messages.

(c) *Impersonation attack*:

- *Node impersonation attack*: We assumed that an adversary tries to impersonate a node with ID_N to communicate with the server. At level 1 security, the adversary cannot compute the session key due to lacking a valid key pair (k_N, K_N) that is bound to the ID_N and the PUF's response $\mathcal{R}^t$. While at level 2 security, the adversary cannot replicate a PUF primitive on the node; hence, it cannot derive the valid key pair (k_N, K_N) without knowledge of the response $\mathcal{R}^t$.
- *Server impersonation attack*: With one-pass SKEM, the server does not respond to the sensor node; thus, we assumed that an adversary tries to impersonate the server to issue the implicit certificate. This attack is only possible if the adversary can break into the server and retrieve the CRP database. Otherwise, the certificate issued by the adversary is meaningless and shall result in an invalid key pair (k_N, K_N).

As a result, the malicious adversary cannot impersonate either a legitimate server or an IoT device.

(d) *Known session key attack*: A protocol with this property can prevent an adversary from accessing future communication to the server, even if they obtain one or more previous session keys. In the proposed protocol, the generation of a sensor node's key pair and session keys requires ephemeral random materials. Therefore, if the session keys are revealed, the future communication channels will remain secure.

(e) *Node capture attack*: We assumed that an adversary has captured a node with ID_N. Through physical memory disclosure attacks, the adversary can retrieve stored data on the node device, such as the valid key pair. As a result, the communication with the level 1 channel would be compromised. To eliminate the risk, a level 2 secure channel computes a fresh key pair (requiring new random numbers and shuffling with new CRP) for every communication session. Another approach to mitigate the risk is to periodically update the key pair by triggering the ECQV scheme after a certain period. Consequently, the proposed protocol offers two methods: (a) complete immunity to the node capture attack and (b) a method that minimizes data leaking.

(f) *Perfect/partial forward secrecy*: Perfect forward secrecy is a security property that ensures that compromising the key pair of one or both entities does not lead to the compromise of past session key establishment. Partial forward secrecy refers to a situation where some, but not all, past communications remain confidential, even if the key pairs are compromised. The level 1 secure channel can offer partial forward secrecy in the case of updating a node's key pair after a certain time. In contrast, triggering the ECQV scheme at level 2 security creates ephemeral key pairs, meaning that one session key corresponds to one key pair. A three-pass handshake can ensure perfect forward secrecy.

(g) *Replay attack*: A replay attack is a form of network attack where an attacker intercepts and maliciously retransmits data that was previously recorded. A one-pass SKEM is susceptible to this kind of attack. Gorantla addressed some solutions to this problem by assigning a session ID for every communication session or applying a time stamp [25]. Updating the key pair of a node is a method to mitigate the issue. Therefore, a secure channel at level 2 can completely prevent this kind of attack.

(h) *Message modification attack*: As we discussed regarding the integrity property, if an attacker manages to capture and alter any part of a message being exchanged over a secure channel, it can result in different keys being extracted by both the sensor node and the server. This can also cause both parties to fail to agree upon the same session key. Consequently, the communication is meaningless and will not leak any critical information; therefore, the protocol is immune to modification attacks.

The summary of the security features of our proposal is shown in Table 2, which also includes comparisons with other protocols.

Table 2. Comparison of functionality features of the proposed protocol with related protocols.

Features	Proposed Level 2	Proposed Level 1	[14]	[28]	[29]	[30]	[31]
Mutual authentication	●	●	●	●	●	●	●
Integrity	●	●	●	●	●	●	●
Impersonation attack	●	●	●	●	●	●	○
Session key attack	●	●	●	●	○	●	●
Node capture attack	●	○	○	●	●	●	○
Perfect forward secrecy	●	●	●	●	●	●	●
Replay attack	●	○	●	●	○	●	○
Message modification attack	●	●	●	●	●	●	●
Anonymity	○	○	●	●	○	●	○

● Secure against a particular attack or supports a particular feature. ○ Insecure against a particular attack or does not support a particular feature.

5. Performance Evaluation

This section describes the experimental setup to analyze the performance of the proposed protocol and the existing protocols. The comparison includes the communication and computational costs. The communication costs were evaluated in terms of the number of exchange message and the total size of these messages. The computational costs were evaluated on an embedded system that can be implemented on actual IoT devices. The protocol was tested with a secp256k1 256-bit EC, with a 256-bit size for the SHA2 hash function. The identity of a node ID and a challenge C^t for a PUF primitive were assumed to be 8 bytes.

5.1. Communication Overhead

In a wireless sensor network with battery-powered or self-charging node devices, to seamlessly monitor the environment, the nodes must be consistently active and periodically transmit data [32]. The proposed protocol aimed to assist the monitoring operation by lowering the complexity of establishing secure communication channels. At the minimum cost with level 1 security, the encapsulated material for session key generation was an identity of a node and a random point of the elliptic curve. This message (ID_N, C) cost $8 + 32 = 40$ bytes. The communication overhead counted only the cost to establish the keys; thus, this message size excluded the size of the ciphertext. The computation overhead of a level 2 secure channel was higher compared with level 1, as it required two more messages. The request from a sensor node (ID_N, C^t, R_n) was $8 + 8 + 32 = 48$ bytes. The corresponding implicit certificate $(r, Cert_N)$ was $32 + 40 = 72$ bytes. In total, the handshake at level 2 security cost three exchange messages and 160 bytes. Note that the overall communication cost varied depending on the frequency of reissuing the key pairs.

We compared our protocol with other existing key establishment schemes that also provide authentication properties in Table 3. In [14], the key establishment required two parties to update information for consecutive communication. Excluding the registration phase, the protocol needed four exchange messages, with a total of 168 bytes. The research in [29] proposed two protocols, in which the handshake of the first protocol cost four exchange messages and 420 bytes, while the second one cost two messages and 142 bytes. In [28,30], the size of messages to create an authenticated channel nearly doubled the cost of a proposed protocol at level 2 security with quantity.

Table 3. Communication cost comparison.

Protocol	Number of Messages	Total Message Size (Bytes)
Ours—level 2	3	160
Ours—level 1	1	40
[14]	4	168
[28]	3	316
[29]—Protocol #1	4	420
[29]—Protocol #2	2	142
[30]	3	332

5.2. Computation Overhead

Different approaches can be used to evaluate the computation overhead of a communication protocol. Some studies, like [14,30], tested the cost of elliptic curve operations and hash functions on desktop processors and then deducted the overall timing cost from their proposed protocols. On the other hand, the studies in [28,29] calculated the overall timing cost by adopting the timing of those operations and functions from other works. These methods can only present a relative timing cost without considering the overhead of a system architecture, like bus interfaces and memory transactions. Therefore, we evaluated the proposed protocol by developing a hardware system. The primary purpose of this approach was to precisely assess the workload of executing the protocol on an embedded system suitable for sensor nodes. The open-source RISC-V processor was utilized to measure the computing performance by implementing two micro-architectural configurations. The RISC-V architecture offers a performance counter that can be accessed through the control and status registers (CSRs). Hence, the measurement metric extracted from the CSRs provides information about the actual clock cycle executed by the processor. This data can be used to derive the timing cost based on the operating clock frequency of the system. By analyzing this information, it is possible to gain insight into the costs associated with different architecture designs and identify areas for further architecture optimization.

The current options for processors in IoT devices include ARM, x86, and RISC-V. With the open-source instruction set architecture (ISA), RISC-V ISA provides a high flexibility in customization for a specific application compared with its counterparts. RISC-V enthusiasts have extended the processor with additional instructions and accelerators to enhance the computing efficiency and provide security and low-power design for IoT devices. Rocket is a variant of the RISC-V processor that allows for modifications at the system and micro-architecture levels. Its architecture is suitable for evaluating the performance of the proposed protocol. The performance data can be used to tailor the Rocket core further to realize the practicality of the protocol for resource-constrained IoT nodes.

The software version of the proposed protocol was implemented on a RISC-V system with a Rocket core, which is a variant of the RISC-V core that can be scaled down for low-power sensor nodes. The RISC-V system was synthesized on an Arty A7-100T Xilinx FPGA to evaluate the computation cost regarding a number of clock cycle counts. Instead of using pseudo- or software-based primitives, the RISC-V system integrated the physical hardware implementation of a true random number generator and a physical unclonable function, as shown in Figure 3.

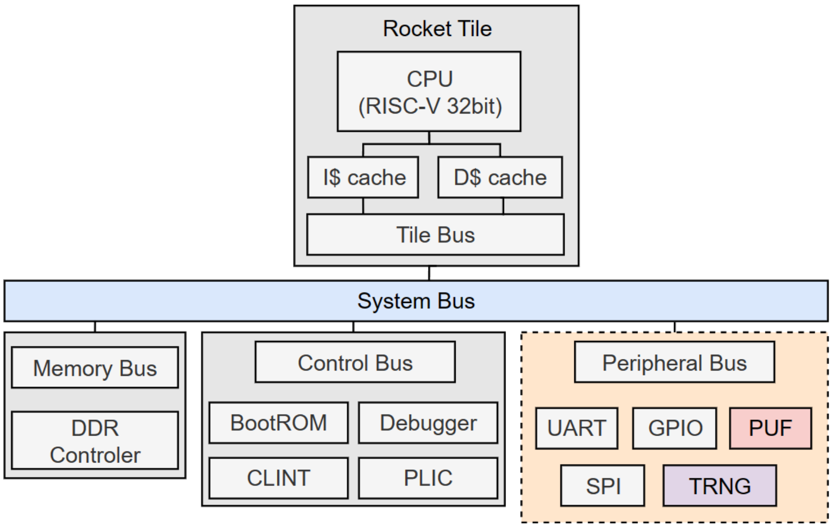

Figure 3. A 32-bit IMAC RISC-V system.

Table 4 shows the notation of computations applied in the proposed protocol and some related works in key establishment based on ECC with an authentication property. Table 5 compares the computation overheads of the protocol at the sensor nodes and the server separately. The proposed protocol aimed to minimize the operations of the sensor nodes as much as possible. The protocol allowed for establishing a level 1 secure channel with the least computational cost $(2T_{sm} + T_a + 2T_h)$. The protocol in [14] yielded the same cost in computing; however, the material to establish a share key was quadrupled. With level 2 security, the computation cost was less than [29,30]. However, the application of the protocol needed to take both the computation and communication of the handshake into consideration.

Table 4. Computation notation and timing of elliptic curve computations.

Notation	Description	RV32-IMAC #Cycle	RV32-I #Cycle
T_h	Hash function	13,943 ± 10	14,178 ± 7
T_{sm}	ECC scalar multiplication	129,821,421 ± 26,894	136,189,184 ± 17,860
T_a	ECC point addition	266,589 ± 32	274,658 ± 13
T_r*	Random number generation	80,649 ± 20	82,349 ± 9
T_{puf}*	PUF function	135,810 ± 14	137,442 ± 28
T_{mod}	Modulo	23,711 ± 4	25,160 ± 18
T_{ba}	Big number addition	698 ± 2	838 ± 1
T_{bm}	Big number multiplication	5009 ± 5	5075 ± 6

* Timing varied depending on the design and system implementation of the primitives.

Table 5. Computation cost comparison.

Protocol	Sensor Node	Sever
Ours—level 2	$4T_{sm} + 2T_a + T_h$	$3T_{sm} + T_a + 3T_h$
Ours—level 1	$2T_{sm} + T_a + 2T_h$	$2T_{sm} + 2T_h$
[14]	$2T_{sm} + 3T_h$	$2T_{sm} + 3T_h$
[28]	$4T_{sm} + 3T_h$	$5T_{sm} + 5T_h$
[29]—Protocol 1	$11T_{sm} + 3T_a + 10T_h$	$4T_{sm} + T_a + 8T_h$
[29]—Protocol 2	$5T_{sm} + 2T_a + 7T_h$	$3T_{sm} + T_a + 7T_h$
[30]	$5T_{sm} + T_a + 16T_h + T_f^*$	$4T_{sm} + T_a + 8T_h$

T_f^*: fuzzy extractor.

Table 4 also presents the measurement results of the EC operations, the hash function, and basic operations on big numbers used in the proposed protocol. The measurement was conducted on two different Rocket cores with two configurations: RV32I and RV32IMAC.

The notations I and IMAC describe the extension of the instruction set architecture used to design the Rocket core. The extension I is the base integer instruction set that can support various operating environments. The extension M contains instructions that support multiplication and division. By default, the RV32I can emulate any extension; however, this experiment aimed to test the significance of other extensions on computing the proposed protocol. The measurement exposed the core operations of each computation to reduce the increase in the cycle count of branching and other operations of the program. It was expected that the ECC scalar multiplication required the most computing effort. The measurement results show that the RV32IMAC with more instruction extensions and hardware cost could achieve a lower cycle count when processing the computations, however the performance improvement was insignificant. In specific, the ECC scalar multiplication could achieve only a 5% increase in performance.

We conducted another timing measurement to assess the execution of the proposed protocol on the Rocket RV32IMAC. The operation of the RISC-V system was expected to be the same as if it was operating at a sensor node. The computations of the protocol were broken down into smaller steps. The timing measurement was conducted to wrap around a function that computed all of the operations in a single step. Table 6 shows the cycle count and execution time of the system running at a 50 MHz clock supply.

Table 6. Timing measurement of individual step on RISC-V 32-bit IMAC.

Proposed Protocol	Computation		#Cycle	Time (ms)
Level 2 ECQV Asymmetric key pair	1.a	$T_r + T_{mod}$	105,810 ± 24	2.11
	1.b	T_{sm}	130,443,888 ± 56,314	2608.88
	1.c	T_r	80,649 ± 20	1.78
	3.a	T_{puf}	135,810 ± 14	2.72
	3.b	$T_h + T_{mod}$	39,103 ± 28	0.78
	3.c	$T_{bm} + T_{ba} + T_{mod}$	30,867 ± 20	0.62
	3.d	T_{sm} + T_a	130,789,946 ± 56,350	2615.80
		Subtotal	261,626,073	5232.69
Level 1 SKEM Symmetric session key	5.a	$T_r + T_{mod}$	105,810 ± 20	2.11
	5.b	T_{sm}	130,443,888 ± 56,314	2608.88
	5.c	$T_h + T_{mod}$	39,103 ± 28	0.78
	5.d	$T_{bm} + T_{ba} + T_h + T_{sm}$	130,463,538 ± 56,326	2609.27
		Subtotal	261,052,339	5221.04
		Total	522,678,412	10,453.73

Based on the experimental results, the RISC-V Rocket core with extra extensions and hardware cost could not deliver a noticeable increase in computing efficiency compared with the default ISA. Therefore, the base configuration Rocket RV32I is suitable as the central processing unit for resource-constrained sensor nodes. To further improve the performance of computing the EC operations, an accelerator designed specifically for EC multiplication attached to the Rocket core as a RoCC, an accelerator connected directly to the Rocket core, is a direct and cost-effective solution. The trade-off between the level 2 and level 1 security of the proposed protocol was that the level 2 secure channel offered more security properties at the cost of doubling the computation cost. In contrast, with a level 1 secure channel, a sensor node required less effort to establish a communication channel, thus reducing the active time of the processor and the transceiver. However, the modified ECQV scheme had to be triggered periodically to mitigate the risk of some attacks.

6. Discussion

The proposed protocol offers a hybrid cryptography approach, integrating both asymmetric and symmetric cryptography to leverage the advantages of each. Unlike other solutions that impose high costs in communication and computation in rekeying every session, the protocol provides a lightweight mechanism to reuse the asymmetric key pair while updating fresh session keys. With these functions, the protocol can still offer essential security properties and prevent common attacks in IoT networks. However, it lacks support for anonymity, which is a critical requirement in highly sensitive data transmission scenarios, such as healthcare, e-payment, and e-voting. Anonymity ensures that the identities of communicating parties remain concealed, thereby protecting privacy and preventing unauthorized tracking. Furthermore, to maximize the benefit of the level 1 security channel, deciding the lifetime of a node's key pair is critical. An optimal update period can maximize the benefit of the fast rekeying mechanism and mitigate the risk of particular attacks like replay attacks. Additionally, despite the intended limitation in computation at the node side, the time required to compute elliptic curve cryptography remains relatively high. Consequently, this limitation restricts the frequency of sensor data transmission periods, particularly in applications requiring the transmission of sensor data every minute. Therefore, we aim to improve the protocol by addressing such limitations in future work at the system's architecture level. Furthermore, a systematic approach to designing a secure system for sensor nodes that encompasses both network and hardware device security is needed. Trusted execution environment (TEE) mechanisms can be integrated to improve a node's security against physical attacks, like side-channel attacks. Safeguarding random sources and protecting the memory storage of keys are mandatory.

7. Conclusions

This paper presents an ECC-based authenticated key establishment protocol for wireless sensor networks. The protocol combines implicit certificate ECQV, one-pass signcryption key encapsulation mechanism schemes, and a physical unclonable function primitive to construct a two-level security protocol. Each level offers a method to establish a secure channel with a trade-off in computation cost, communication cost, and security properties. The security analysis demonstrated that the proposed protocol is able to defend against common attacks on IoT networks. The protocol enables a fast rekeying method with only one message exchange at the minimum cost, which is beneficial for wireless sensor networks. In addition, we evaluated the overhead of the proposed protocol to other existing ECC-based authenticated key establishment protocols. The result shows that our protocol performed better in terms of the computation cost, number, and size of the exchanged messages. The comparison results indicate that the proposed protocol is a competitive approach for achieving highly secure and low-cost communication in wireless sensor networks. Moreover, a timing measurement of the protocol on a 32-bit RISC-V system, which was integrated with two essential hardware primitives (TRNG and PUF), was provided to evaluate the practicality of the protocol on an embedded processor. Based on the measurement results, further improvement on the RISC-V hardware can result in an overall low-cost RISC-V system that supports the protocol for sensing or monitoring applications.

Author Contributions: Conceptualization, C.-K.P.; Methodology, T.-K.D.; Software, T.-K.D.; Investigation, T.-K.D.; Resources, K.-D.N. and B.-K.-D.-N.; Writing—original draft, T.-K.D.; Writing—review & editing, K.-D.N., B.-K.-D.-N., T.-T.H. and C.-K.P.; Supervision, T.-T.H. and C.-K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Aras, E.; Ramachandran, G.S.; Lawrence, P.; Hughes, D. Exploring the Security Vulnerabilities of LoRa. In Proceedings of the 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), Exeter, UK, 21–23 June 2017; pp. 1–6. [CrossRef]
2. Sundaram, J.P.S.; Du, W.; Zhao, Z. A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues. *IEEE Comm. Surv. Tutor.* **2019**, *22*, 371–388. [CrossRef]

3. Loukil, S.; Fourati, L.C.; Nayyar, A.; In, C.S. Investigation on Security Risk of LoRaWAN: Compatibility Scenarios. *IEEE Access* **2022**, *10*, 101825–101843. [CrossRef]
4. Han, J.; Wang, J. An Enhanced Key Management Scheme for LoRaWAN. *Cryptography* **2018**, *2*, 34. [CrossRef]
5. Naoui, S.; Elhdhili, M.E.; Saidane, L.A. Enhancing the Security of the IoT LoraWAN Architecture. In Proceedings of the International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), Paris, France, 22–25 November 2016; pp. 1–7.
6. Pu, C.; Zerkle, H.; Wall, A.; Lim, S.; Choo, K.-K.R.; Ahmed, I. A Lightweight and Anonymous Authentication and Key Agreement Protocol for Wireless Body Area Networks. *IEEE Internet Things J.* **2022**, *9*, 21136–21146. [CrossRef]
7. Zheng, Y.; Chang, C.-H. Secure Mutual Authentication and Key-Exchange Protocol between PUF-Embedded IoT Endpoints. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Republic of Korea, 22–28 May 2021; pp. 1–5. [CrossRef]
8. Torres, N.; Pinto, P.; Lopes, S.I. Security Vulnerabilities in LPWANs—An Attack Vector Analysis for the IoT Ecosystem. *Appl. Sci.* **2021**, *11*, 3176. [CrossRef]
9. Keoh, S.L.; Kumar, S.S.; Tschofenig, H. Securing the Internet of Things: A Standardization Perspective. *IEEE Internet Things J.* **2014**, *1*, 265–275. [CrossRef]
10. Sciancalepore, S.; Piro, G.; Boggia, G.; Bianchi, G. Public Key Authentication and Key Agreement in IoT Devices with Minimal Airtime Consumption. *IEEE Embed. Syst. Lett.* **2017**, *9*, 1–4. [CrossRef]
11. Rehman, A.; Haseeb, K.; Fati, S.M.; Lloret, J.; Peñalver, L. Reliable Bidirectional Data Transfer Approach for the Internet of Secured Medical Things Using ZigBee Wireless Network. *Appl. Sci.* **2021**, *11*, 9947. [CrossRef]
12. Gaabouri, I.E.; Senhadji, M.; Belkasmi, M.; Bhiri, B.E. A Systematic Literature Review on Authentication and Threat Challenges on RFID Based NFC Applications. *Future Internet* **2023**, *15*, 354. [CrossRef]
13. Mao, G.; Liu, Y.; Dai, W.; Li, G.; Zhang, Z.; Lam, A.H.F.; Cheung, R.C.C. REALISE-IoT: RISC-V-Based Efficient and Lightweight Public-Key System for IoT Applications. *IEEE Internet Things J.* **2024**, *11*, 3044–3055. [CrossRef]
14. Li, B.; Zhang, G.; Lei, S.; Fu, H.; Wang, J. A Lightweight Authentication and Key Agreement Protocol for IoT Based on ECC. In Proceedings of the 2021 International Conference on Advanced Computing and Endogenous Security, Nanjing, China, 21–22 April 2022; pp. 1–5. [CrossRef]
15. Zhang, W.; Lin, D.; Zhang, H.; Chen, C.; Zhou, X. A Lightweight Anonymous Mutual Authentication with Key Agreement Protocol on ECC. In Proceedings of the IEEE Trustcom/BigDataSE/ICESS, Sydney, Australia, 1–4 August 2017; pp. 170–176. [CrossRef]
16. Certicom Research. *SEC 4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)*; Certicom Research: Mississauga, ON, Canada, 2013.
17. Cramer, R.; Shoup, V. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. *SIAM J. Comp.* **2003**, *33*, 167–226. [CrossRef]
18. Bjørstad, T.E.; Dent, A.W. Building Better Signcryption Schemes with Tag-KEMs. In Proceedings of the International Conference on Public Key Cryptography (PKC), New York, NY, USA, 24–26 April 2006; pp. 491–507. [CrossRef]
19. Bjørstad, T.E.; Dent, A.W.; Smart, N.P. Efficient KEMs with Partial Message Recovery. In Proceedings of the International Conference on Cryptography and Coding, London, UK, 12–14 December 2023; pp. 233–256. [CrossRef]
20. Barnes, R.; Bhargavan, K.; Lipp, B.; Wood, C. RFC9180-Hybrid Public Key Encryption. February 2022. Available online: https://www.rfc-editor.org/rfc/rfc9180.html (accessed on 2 February 2024).
21. Zheng, Y. Digital Signcryption or How to Achieve Cost(Signature & Encryption) ≪ Cost(Signature) + Cost(Encryption). In Proceedings of the Annual International Cryptology Conference CRYPTO ’97, Santa Barbara, CA, USA, 17–21 August 1997; Volume 1294, pp. 165–179.
22. Dent, A.W. Hybrid Cryptography. IACR Cryptology ePrint Archive, 2004. p. 210. Available online: https://eprint.iacr.org/2004/210 (accessed on 30 April 2024).
23. Dent, A.W. Hybrid Signcryption Schemes with Outsider Security. In Proceedings of the Information Security (ISC 2005), Singapore, 20–23 September 2005; pp. 203–217.
24. Dent, A.W. Hybrid Signcryption Schemes with Insider Security. In Proceedings of the Information Security and Privacy (ACISP), Brisbane, Australia, 4–6 July 2005; pp. 253–266.
25. Gorantla, M.C.; Boyd, C.; Nieto, G.; Manuel, J. On the Connection between Signcryption and One-Pass Key Establishment. In Proceedings of the Cryptography and Coding, Cirencester, UK, 18–20 December 2007; pp. 277–301.
26. Krawczyk, H. HMQV: A High-performance Secure Diffie-Hellman Protocol. In Proceedings of the Annual International Conference on Advances in Cryptology (CRYPTO), Santa Barbara, CA, USA, 14–18 August 2005; pp. 546–566.
27. Okamoto, T.; Tso, R.; Okamoto, E. One-Way and Two-Party Authenticated ID-Based Key Agreement Protocols Using Pairing. In Proceedings of the International Conference on Modeling Decisions for Artificial Intelligence (MDAI), Tsukuba, Japan, 25–27 July 2005; pp. 122–133. [CrossRef]
28. Challa, S.; Wazid, M.; Das, A.K.; Kumar, N.; Reddy, A.G.; Yoon, E.-J.; Yoo, K.-Y. Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications. *IEEE Access* **2017**, *5*, 3028–3043. [CrossRef]
29. Porambage, P.; Braeken, A.; Schmitt, C.; Gurtov, A.; Ylianttila, M.; Stiller, B. Group Key Establishment for Enabling Secure Multicast Communication in Wireless Sensor Networks Deployed for IoT Applications. *IEEE Access* **2015**, *3*, 1503–1511. [CrossRef]

30. Srinivas, J.; Das, A.K.; Wazid, M.; Vasilakos, A.V. Designing Secure User Authentication Protocol for Big Data Collection in IoT-Based Intelligent Transportation System. *IEEE Internet Things J.* **2021**, *8*, 7727–7744. [CrossRef]
31. Porambage, P.; Schmitt, C.; Kumar, P.; Gurtov, A.; Ylianttila, M. Two-phase Authentication Protocol for Wireless Sensor Networks in Distributed IoT Applications. In Proceedings of the 2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, Turkey, 6–9 April 2014; pp. 2728–2733. [CrossRef]
32. Pu, C.; Lim, S. A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation. *IEEE Syst. J.* **2018**, *12*, 834–842. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.