Key-Establishment Protocols for Constrained Cyber-Physical Systems

Carlos Andres Lara-Nino, Arturo Diaz-Perez, and Miguel Morales-Sandoval

Abstract Cryptographic keys are critical components when deploying efficient and strengthened security solutions for confidentiality, integrity, and authentication in different computer application domains. In this Chapter, we present three key-establishment protocols that are well-suited for constrained cyber-physical systems (CPSs), using wireless sensor networks (WSNs) as the particular application scope. The focus was on two-party and balanced protocols suitable for the heterogeneity and nondeterministic characteristics of WSNs. The protocols under study offer different security features that might be attractive for different applications depending on the information sensitivity and computing capabilities of the underlying devices. We studied two lightweight key-establishment protocols based on elliptic-curve cryptography (ECC), enhanced by the use of other cryptographic constructions, such as ciphers, hash functions, key derivation, and physically unclonable functions (PUFs). We also present a novel protocol for key establishment constructed on isogeny-based key-encapsulation mechanism SIKE, well-suited for operating in CPSs in the context of a post-quantum computing scenario.

1 Introduction

Cyber-physical systems (CPSs) offer multiple opportunities for deploying applications that have a direct relation with the physical world. As detailed in [1], the core idea of CPSs is the monitoring and controlling of physical objects through interconnected software systems. The idea of CPSs is tightly related to the concepts of

C. A. Lara-Nino · M. Morales-Sandoval
CINVESTAV Campus Tamaulipas, Ciudad Victoria 87130, Mexico
e-mail: carlos.lara@cinvestav.mx
e-mail: miguel.morales@cinvestav.mx

A. Diaz-Perez (✉)
CINVESTAV Campus Guadalajara, Zapopan 45017, Mexico
e-mail: adiaz@cinvestav.mx

A. I. Awad et al. (eds.), *Security in Cyber-Physical Systems*, Studies in Systems, Decision and Control 339, https://doi.org/10.1007/978-3-030-67361-1_2

ubiquitous computing, sensor networks, and the Internet of Things (IoT). The main difference is that CPSs focus on the interaction of the objects with their environment [1].

CPSs are of great relevance for applications such as the navigation and control of autonomous vehicles, the management of water resources, power systems, and smart grids, the supervision and control of oil and gas distribution systems, and remote healthcare monitoring.

Some of these applications demand the miniaturization of the devices in order to either reduce manufacturing costs, such as in management and monitoring, or to improve the user's perception of the technology, which is desirable in healthcare applications. Downsizing devices and reducing their manufacturing costs tend to impose stricter restrictions on the platform. The use of processors with lower specifications, smaller memories, and low-cost power supply translates into constraints for applications running on the device.

By *constrained devices*, we understand a network participant that must adhere to physical restrictions given by the application or environment where it is used. Such constraints might come in the form of performance, storage, bandwidth, or energy limitations. Consequently, a *constrained environment* is defined as a computational system of multiple elements that can be homogeneous or heterogeneous, and which contains devices of limited capabilities: constrained devices.

Examples of constrained devices are wireless-sensor-network (WSN) motes and radio-frequency-identification (RFID) tags. As a consequence, WSNs, RFID, and similar applications are considered constrained environments. Our work focused on WSNs, as illustrated in Fig. 1, a key enabler for CPSs [1].

A constrained CPS tasked with managing sensitive data requires at least the same security services as those of a conventional network, although these devices have less processing power [2]. In some CPS cases, constrained devices are deployed under hostile environments. This implies that an attacker can have physical access to the network. Additional security measures should be considered to patch these vulnerabilities. In big-data scenarios, the high data volume from sensors, even if it is not inherently sensitive, can be exploited for inferring knowledge about the monitored systems. Due to these reasons, all messages transmitted through sensor nodes must be provided with information security.

Resilience against the intentions of malicious actors can be obtained by providing the data with security services. *Confidentiality* can thwart eavesdropping; *integrity* and *authentication* are used to corroborate the veracity of a message; *availability* ensures that the data can be accessed on-demand. These precepts are enforced through the use of cryptographic algorithms. However, most of these cryptosystems require that the participants in the communication exchange share a common data-denominated key. As stated in [3], key management is one of the fundamental issues in CPS security.

CPS characteristics must be considered in the design of security systems: heterogeneity, real-time operation, extended threat models, interoperability, and survivability. These particularities make the design of efficient security solutions quite challenging.

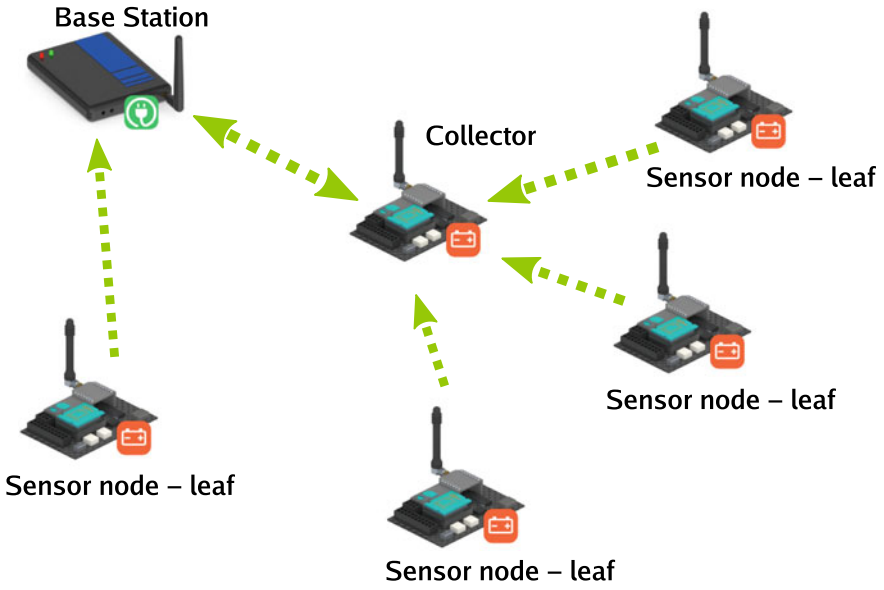

Fig. 1 WSN comprises a base station and multiple sensor nodes. Nodes that are more geographically separated from the base station must employ multi-hop links to transfer their messages to it. The network topology is nondeterministic, and sensor nodes are powered with batteries

According to [1], the great potential and envisioned benefits of CPSs stand in stark contrast to the different security threats that limit the widespread adoption of the technology by reducing the user's trust in these systems. The authors identified the divergence with the client-server model of the Internet stack as the main challenge. Thus, solutions developed for the Internet cannot be directly applied to CPSs. This poses challenges and opportunities in seeking new security solutions for these applications. Furthermore, the evolving nature of these technologies, the increment of their features, and the emergence of new ways of interaction depend on a constantly expanding threat model. The authors of [4] noted that understanding and addressing these threats is a critical challenge in order to improve a user's acceptance of the technology, which would in turn further the development of these systems.

In the past decade, the study of security solutions for constrained devices has gained popularity. Cryptographic algorithms have played an important role in providing constrained systems with the required security services for data confidentiality, integrity checks, and authenticity by means of encryption, authentication codes, and digital signatures. These cryptographic and lightweight solutions for networked environments are constructed from symmetric or asymmetric (public-key) cryptography primitives. In these scenarios, key security is critical for system safety.

Some of the challenges that must be solved by lightweight key-establishment solutions for constrained CPSs are reducing the complexity of underlying operations,

decreasing storage costs, mitigating lengthy processing delays, and adapting to the relentless advance of attack threats and vulnerabilities.

The main contributions of this work are twofold:

1. We study the suitability of different solutions for providing key establishment to constrained cyber-physical systems.
2. We provide three two-party balanced key-establishment protocols that are well-suited for constrained cyber-physical systems (CPSs).

This Chapter is structured as follows. In Sect. 2, we discuss the different characteristics observed on key establishment protocols and how these make them more or less suitable for the application scope of WSNs and thus CPSs. Section 3 elaborates about notions on security services and cryptographic principles that are used in this chapter. Section 4 presents an analysis of relevant works from the literature. In Sect. 5, we describe and analyze two key establishment protocols based on elliptic curve cryptography; we explore their characteristics, assess their communications and processing overheads, and study their security properties. Section 6 presents a novel key establishment protocol created to operate with quantum-safe encapsulation mechanisms on two-party scenarios; this protocol is also evaluated and compared against the solutions described in Sect. 5. Lastly, Sect. 7 summarizes our findings and concludes this chapter.

2 The Problem of Key Establishment

Standards such as IEEE 802.15.4 [5] specify mechanisms for obtaining confidentiality and authentication on low-rate wireless-personal-area networks (LR-WPANs) by using standardized cryptographic algorithms. However, these cryptosystems require that link participants have a shared key. This can be challenging for constrained CPSs like WSNs or related technologies.

Given that the topology of a WSN is nondeterministic, it can be expected that each of its nodes is capable of creating a secure link with any other node in its proximity. A straightforward approach for key establishment consists of storing a master key in each device in the network; however, if an attacker manages to retrieve this information, the security of the whole system would crumble. On the other hand, if each node must store a session key for linking up with every possible device in the network, then the device's memory requirements would exponentially grow with the number of network participants.

The key-establishment problem (KEP) lies in the difficulty of enabling a group of two or more network participants to establish a shared piece of information in a secure fashion. As mentioned before, this key is fundamental for securing the communication channel and providing network messages with security services. Key-establishment protocols are algorithms created for solving this problem.

As a goal-driven process, key establishment can be broadly divided into key transport and key agreement. According to [6], these are defined as:

- A key-transport protocol or mechanism is a key-establishment technique where one party creates or otherwise obtains a secret value, and securely transfers it to the other(s).
- A key-agreement protocol or mechanism is a key-establishment technique in which a shared secret is derived by two (or more) parties as a function of information contributed by or associated with each of these, ideally so that no party can predetermine the resulting value.

Key-predistribution schemes are a particular class of key agreement, where shared keys are completely determined a priori by using some primordial keying materials. In this case, the key is fixed or static, and in some instances, such as for WSNs, it cannot be modified post-deployment. In contrast, dynamic key-establishment schemes are those where the key can be established by the participants on subsequent executions. Given that the key can be constructed by employing secret materials from all participants or generated by some coordinator, either key-agreement or -transport solutions can be classified as dynamic.

Another useful classification for key-establishment protocols was proposed in [7]. In this case, the discriminant characteristic was the method employed for establishing the keys. In that work, four main classes were identified:

1. Key predeployment of:
   - global key: a single key that is preloaded to all sensor nodes in the network;
   - full pairwise key: in a network of n nodes, each node has to store a key for each of the other $n - 1$ nodes, thus having to store $\frac{n(n-1)}{2}$ keys; and
   - random key set: each node is loaded with a set of keys chosen randomly from a key pool.
2. Key derivation from pre-deployment information:
   - using a transitory master key that expires after some event; and
   - using a keying root that serves as provisional trust.
3. Key-management schemes based on hard mathematical problems:
   - solutions based on symmetric cryptography;
   - solutions based on asymmetric cryptography; and
   - hybrid approaches.
4. Over-the-air key-establishment protocols that:
   - extract secret keys from received signal strength; and
   - leverage channel anonymity for generating pairwise secret keys.

In most cases, dynamic key-exchange mechanisms represent the most viable solution for KEPs by enabling a node to establish shared secrets with nearby devices after deployment.

In IEEE 802.15.4, the mechanism for network participants to establish shared keys is not specified; this is also the case for other norms. Even though key establishment is an old problem, and that there are multiple standardized solutions available, most of these solutions were designed for general applications and do not consider the

multiple limitations of constrained devices. Envisioned solutions for cryptographic-key establishment on constrained CPSs must be carefully designed for incurring low overheads in terms of processing, storage, and transmission costs.

Of the described KEP solution approaches, those that employ cryptographic algorithms are generally preferred. Symmetric cryptography approaches tend to be more efficient, but have shortcomings for networked environments, which can be addressed with the use of asymmetric cryptography [8]. From this class of algorithms, solutions based on elliptic curves have the main advantage of needing lower memory and processing overheads for the underlying system [9].

3 Security Notions

Providing information security greatly depends on assumptions made about attacker capabilities and system vulnerabilities. Security services ensure that certain data characteristics are protected. As introduced in previous sections, the most basic of such services are confidentiality, integrity, and authentication.

Data confidentiality implies that only authorized parties have access to the information. When an attacker gains access to the data, its confidentiality is broken, as the privacy of the information cannot be guaranteed. If the attacker's goal is to modify a message, this represents an attack on the information's integrity. Authentication is a particular case of integrity where data origin is also verified.

Most key-establishment protocols rely on these basic security services for constructing or distributing secrets in a safe manner. The strength of a protocol relies on the resilience of its building blocks against cryptographic attacks.

A data cipher is a cryptosystem formed by an encryption function $\mathcal{E}$ and a decryption function $\mathcal{D}$. The main purpose of these algorithms is to ensure privacy by means of the confidentiality service. During its operation, $\mathcal{E}$ employs a key K_E from key space $\mathcal{K}$ to map plaintext P from message space $\mathcal{M}$ into a ciphertext C in $\mathcal{C}$, the ciphertext space. Decryption function $\mathcal{D}$ and a decryption key K_D, also in $\mathcal{K}$, are necessary to retrieve P from C.

If K_E is the same as K_D, it is said that the cipher is symmetrical. On the other hand, if K_E differs from K_D, the cipher is considered asymmetric; in this case, K_E would be of the public domain, while K_D would have to be kept private. The public key of a network participant is used by third parties to encrypt messages that only the private key holder can retrieve. The public key K_E is obtained from K_D by using one-way functions that rely on hard mathematical problems so that K_D cannot be retrieved from K_E. These asymmetrical key systems conform to public-key cryptography (PKC).

Ensuring the integrity and authenticity of a message or its sender requires that the exchange participants gain an information advantage over a potential attacker. These data are either pre-distributed over a trusted channel or derived from some session data that are unknown to the adversary. Message authentication codes (MACs) are tags appended to a message so that the receiver could verify that tag and corroborate

the relevant security properties. These codes can be obtained through the use of MAC functions. These cryptosystems incorporate a generation engine $\mathcal{T}$ and a verification function $\mathcal{V}$. To generate T, generator $\mathcal{T}$ employs an authentication key K_T from key space $\mathcal{K}$, and the input message, preferably a ciphertext C, so that $T = \mathcal{T}(C, K_T)$. The verification function $\mathcal{V}$ employs received tag T, received message C', and its verification key K_V for computing $T' = \mathcal{T}(C', K_V)$; if $T \equiv T'$, the verification is valid, and C can be decrypted; else, a nonalphabetic symbol is produced. When K_T is the same as K_V, the MAC function is symmetric, for example, a cipher with an authentication mode or an HMAC. If these keys are different, then the MAC function is called a signature, where K_V is derived from K_T with a one-way function. In this case, K_V is public, and K_T is private. The signer uses its private key for creating a MAC that anybody can verify by using K_V.

There are multiple ways in which the protocol itself can lead to unseen vulnerabilities, even if the underlying ciphers and MACs are secure. In [10], the authors reflected that "it is quite easy to propose protocols in which subtle security problems later emerge". Some of these problems arise from common issues:

- It is unwise to derive a shared secret from the result of a key-establishment mechanism by truncation. Even if retrieving the whole shared key could be a computationally intractable problem, an attacker might still be able to retrieve a reduced portion of it. The indirect use of the shared key also shields it from information leaking; revealing partial information about the key can lead to faulty protocols.
- In practice, an attacker can not only listen to the channel but also inject data into the line. This is the difference between passive and active attacks. The latter is closer to real-world scenarios.
- It should be assumed that a device is capable of maintaining multiple link instances with different network participants. Even if one of these keys is leaked, this should not compromise the other instances.
- The fact that a protocol is logically correct does not imply that it is secure.
- It is necessary to specify what exactly the problem being solved is. Providing a model of adversarial capabilities and a definition of security is critical for determining if a protocol is secure.

These points are relevant in the design of secure key-establishment protocols. In the particular case of constrained CPSs, such as WSNs and the IoT, additional factors must be considered:

- These networks can be deployed in nonstructured environments. In such scenarios, the availability of network infrastructure such as stable links and trusted third parties cannot be guaranteed.
- The rapid deployment of the networks and mobility make it impossible to have a defined topology. Each participant should be able to establish a secure channel with another participant at any given time.
- The evolving nature of the networks and the diversity of tasks performed implies that their composition is heterogeneous. The computational load of the protocols

needs to be even so that even the most constrained participants can consistently establish secure links.
- The wide areas where these networks are deployed and their proximity to the physical world grant attackers physical access to the devices. This is a unique characteristic of some CPS networks.
- Most network participants are under some type of processing, bandwidth, or energy constraints. Hence, protocol complexity should be kept to a minimum.

In order to outline the scope of this work, the following notions from [10] were employed:

1. The goal of key distributions considered is for the parties to simultaneously authenticate one another and come into possession of a secure, shared session key.
2. An active adversary attacks the network. The adversary controls all the communication among the players: it can deliver messages out of order and to unintended recipients, concoct messages entirely of its own choosing, and start-up entirely new instances of players. Furthermore, it can mount various attacks on a session key [...]

However, given the particular conditions of CPS networks, adversarial capabilities need to be enhanced:

2a. The adversary has physical access to the network participants.

This critical condition implies that the protocol must account for the potential capture, displacement, impersonation, and cloning of devices. These challenges are not trivial when the physical restrictions of the network participants are considered.

4 State of the Art

Solving the problem of key establishment between participants of a CPS network is regarded as the main security concern in this area [11]. As reviewed in Sect. 2, these mechanisms are classified according to four main strategies:

1. keys are preloaded;
2. challenges are employed on the basis of prior available information;
3. cryptographic algorithms are required for deriving or transmitting shared secrets; and
4. information from the channel is used to generate a key.

These solutions can also be classified depending on the general structure of the protocol. If any pair of devices can establish a shared secret, we denominate such proposals as distributed (D). Conversely, if a device requires the intervention of a central entity for joining the network, such protocol is said to be centralized (C).

Other characteristics that are relevant in the study of key-establishment protocols are:

- security fundamentals: underlying principle for assuming solution security;
- application: a particular environment for which a solution is conceived, where application constraints can guide the design process of this solution; and
- network assumptions: suppositions regarding the network composition or infrastructure that are fundamental for a particular solution and tend to restrict the solution scope.

In the following, we examined different works that proposed key-establishment solutions for CPSs.

4.1 Literature Review

In [12], the authors introduced a modified-matrix-based pairwise key-establishment scheme for wireless mesh networks. In their approach, each node was preloaded with a key seed that, together with a public matrix, was used to generate a column of a secret matrix. This matrix was created and broadcast by a network router, so any adjacent pair of nodes could obtain a key pair by selecting the respective matrix column. The main assumption of this work was that mesh routers are more powerful than the nodes; hence, offloading some matrix computations reduces storage and communication at the nodes. The computation cost for the nodes was equivalent to performing a polynomial evaluation, while the communication costs of employing a large matrix as public key were not addressed.

The authors of [13] proposed a hybrid key-distribution scheme by employing chaotic maps for key generation, and a zero-knowledge-proof protocol for authentication. The proposal claimed to provide authentication, integrity, and confidentiality. According to the authors, the protocol was less complex and required fewer message exchanges than previous schemes did, while improving security.

A key-establishment approach based on ambient wireless signals and symmetric cryptography was proposed in [14]. The authors stated that the heterogeneity of CPS manufacturers makes key-predistribution models impractical. They proposed to use a key-derivation method by [15] in order to generate a trusted root with the central authority (CA) of the network. Following this authentication step at the physical layer, the node obtained a set of credentials from the CA that were used in higher layers. One of the main concerns with this approach is the assumption from [15] that an attacker cannot obtain a trusted root unless it is in close proximity to authentic nodes. However, CPSs are often deployed in unstructured environments, so an attacker could gain physical access to the network.

Some CPSs, such as those used in automotive applications, have particularities that demand the design of ad hoc security solutions. In [16, 17], an authenticated key-establishment protocol for automotive CPSs was proposed. The described approach employed high-security asymmetric and symmetric cryptography algorithms such as ECDSA, AES, and SHA-3 for providing key establishment, confidentiality, integrity, and authentication to vehicular networks. It was assumed that only intravehicle elec-

tronic control units (ECUs) were valid network participants. These nodes were multicore processors with multithreading capabilities in charge of different systems of the vehicle. One of the main concerns of the proposal was fault tolerance, which was solved by performing redundant computations. The authenticity of the ECUs was resolved with the use of public-key certificates. However, since the network was assumed to be intravehicular, dynamic key-establishment mechanisms would not be required. The number of participants was also limited; hence, key-predistribution approaches were also be used. The authors justified their use of public-key cryptography in potential key-recovery attacks and vulnerabilities on the generation of master keys, but these could be addressed with less costly approaches, such as the use of physically unclonable functions (PUFs). While the network participants could shoulder this security overhead, more computations also convey a greater risk of operational faults.

For WSNs, the authors of [18] proposed a key-predistribution scheme based on polynomial pool-based key predistribution and random key predistribution. The approach required to preload each sensor with a set of random polynomial shares and a set of random keys. This generated better chances for nodes to establish a viable network while reducing the impact of node capture by an attacker. Nonetheless, the security and viability of the scheme still depended on node memory availability. Then, that solution was as effective as random key-predistribution models since preloaded keys were not removed from the devices after deployment. Furthermore, the proposed approach considered three mechanisms that could be used by each node upon device discovery. This not only increased the possibility of introducing unseen vulnerabilities but as the authors acknowledged, "path-key establishment is a complicated procedure. It requires more communication and computational overhead for the establishment of path keys between neighboring nodes." This contrasts with the also acknowledged "constrained memory, energy, and computational capabilities of sensor nodes".

In [19] the authors proposed a solution for authentication and key-establishment in cloud-assisted CPSs within the context of a smart grid. The protocol was designed to provide mutual authentication between user and cloud service, and between smart meters and the cloud. When the parties in any of these cases were mutually authenticated, a trusted authority was tasked with enabling these actors to establish session keys. The security of this scheme relied heavily on ECC, enhanced with biometrics on the user's end of the protocol. To prevent replay attacks, the authors considered that all the participants were synchronized with a clock. This protocol was further studied in [20], where the authors claimed to have found deadlocking errors and vulnerabilities against reply and denial-of-service attacks. The scheme was corrected in that work at the cost of increasing computation and communication costs.

For a conventional smart-grid model, the authors in [21] proposed an authenticated key-establishment protocol. This solution relied on the availability of a trusted actor for validating the authenticity of the parties. The protocol employed ECC and symmetric-cryptography algorithms for providing basic security services to the network.

Table 1 Characteristics of surveyed key-establishment proposals for cyber-physical systems (CPSs)

Year	References	Strat.	Struc.	Fundamentals	Application	Network assumptions
2013	[12]	2	D	Matrix arithmetic	Wireless mesh networks	The routers are more powerful than the clients
2017	[13]	3	D	Chaotic Chebyshev polynomials, Zero Knowledge Proof	Environmental monitoring	Availability of Machine to Machine communication
2017	[14]	4	C	Ambient wireless signals, symmetric cryptography	Generic Cyber-Physical Systems	System authority available. Attackers have restricted physical access
2018	[16, 17]	3	D	Asymmetric and symmetric cryptography	Automotive CPS	The networks are intra-vehicle. The nodes are multi-core processors with multithreading capabilities
2018	[18]	1	D	Bivariate t-degree finite field polynomials	Wireless Sensor Networks	The devices have sufficient memory resources to implement a functional configuration of the solution
2020	[19]	3	C	Biometric authentication and asymmetric cryptography	Cloud-assisted Smart Grid	A trusted authority is available. Parties are synchronized with a clock
2020	[21]	3	C	Asymmetric and symmetric cryptography	Smart Grid	A certification agency is available

Table 1 provides a summary of the characteristics of the different works from the literature.

The key-establishment protocols proposed in this work employ a hybrid approach by combining symmetric and asymmetric algorithms. Their aim is to enable any pair of devices to establish a shared secret in a secure way without extended network assumptions. The target application is WSNs; therefore, the proposed adversarial model was extended as defined in Sect. 3.

5 Lightweight Key-Establishment Protocols Based on Elliptic-Curve Cryptography

In 1976, Diffie and Hellman [22] proposed a key-establishment solution on the basis of the hardness of the discrete-logarithm problem (DLP) defined over multiplicative group Z_p^*:

Definition 1 Let a prime p and a generator $G \in Z_p^*$ be parameters of the public domain. Given $X = G^x$, compute x.

In the Diffie-Hellman (DH) protocol, let x, y, two random elements in Z_p^*, be the private keys of exchange parties A and B, respectively. The order of the group determines the complexity of computing the DLP, and consequently the security strength of key establishment.

To obtain a shared secret, A selects $x \in [1, p-1]$ at random and computes its public key $X = G^x$. This value is transferred to interlocutor B. Then, B selects $y \in [1, p-1]$ at random and computes its public key $Y = G^y$, which is transferred to A. Parties A and B then compute $K_A = Y^x$ and $K_B = X^y$, respectively. Note that

$$Y^x = (G^y)^x = G^{yx} = (G^x)^y = X^y; \tag{1}$$

thus, the exchange participants then share a common piece of information that can be used as a precursor for deriving cryptographic keys.

For a large group Z_p^*, the security of the DH protocol relies on the difficulty for an attacker of solving the Diffie-Hellman computational problem (DHCP):

Definition 2 Let a prime p and a generator $G \in Z_p^*$ be parameters of the public domain. Given G^x and G^y for x, y chosen at random from $[1, p-1]$, compute G^{xy}.

Or the Diffie-Hellman decisional problem (DHDP):

Definition 3 Let a prime p and a generator $G \in Z_p^*$ be parameters of the public domain. Given G^x, G^y, and G^z for x, y, z, chosen at random from $[1, p-1]$, decide whether $G^z = G^{xy}$.

As discussed in [10], although the DLP is considered a computationally hard problem, there is no hard proof that the DHCP can only be solved through computing discrete logarithms. Nonetheless, over the years, no such attack has been found; thus, the DHCP is also considered intractable for a computer. The corollary of this is:

> The Diffie-Hellman key exchange is secure in the sense that a computationally bounded adversary cannot compute the secret key shared by the participants.

Alternatively, gain some information advantage in distinguishing the shared key from a random string. The cost for the network participant is to perform modular exponentiation ($G^x \in Z_p^*$).

Over the years, a reduction in the computation bound for adversaries has required that the length of p be increased up to a few thousand bits. This has impacted the time complexity of the modular exponentiations that are fundamental in the exchange.

In order to improve the efficiency of this algorithm, a modification was proposed in 1986 for replacing the DH multiplicative group with abelian elliptic-curve groups [23]. This came to be known as the elliptic-curve Diffie-Hellman exchange (ECDH).

One of the main changes introduced with the use of elliptic-curve groups in the DH exchange is that the main operation, which had previously been modular exponentiation, was replaced by scalar multiplication. In the ECDH, this operation represents the consecutive addition of $k - 1$ instances of the group generator or base point, where this addition is defined over the elliptic-curve group. In the following, this operation is illustrated by using the $\cdot$ operator.

Let q a large prime defining finite field $\mathbb{F}_q$. Let E an elliptic curve over $\mathbb{F}_q$, whose set of points $E(\mathbb{F}_q)$—affine coordinate pairs $(x, y) \in \mathbb{F}_q^2$ solving for $E(x, y) \in E(\mathbb{F}_q)$—together with a point at infinity $\mathcal{O}$, form an abelian group of order n. Let G a generator for this group; the public key of ECDH is $P = k \cdot G$, where $k \in [1, n - 1]$ is the secret key.

Definition 4 Given adequate domain parameters (q, E, G, n), so that n is large, and the resulting value of $P = k \cdot G$, compute k.

This is known as the elliptic-curve discrete-logarithm problem (ECDLP), and it is considered intractable in polynomial time for a computationally bound adversary.

Let A and B two parties that agree on the common domain parameters: (q, E, G, n). Suppose A and B want to establish a shared key. Party A randomly chooses $a \in [1, n - 1]$ and computes $P_A = a \cdot G$, while B follows the same procedure and obtains $P_B = b \cdot G$. A and B publicly exchange these intermediate results. Upon receiving P_B, A computes

$$P_K = a \cdot P_B = (a \times b) \cdot G. \tag{2}$$

Now, B obtains the same result as

$$P_K = b \cdot P_A = (b \times a) \cdot G, \tag{3}$$

so, they are both in possession of a group element P_K that can be used for creating a shared key. The interaction diagram for the basic ECDH protocol is illustrated in Fig. 2.

Due to ECDLP, a or b cannot be computed given $\{P_A, G\}$ or $\{P_B, G\}$, respectively. Due to ECDH, P_K cannot be retrieved from P_A or P_B, employing the same computational and decisional notions of DH. As a protocol, the problems that an attacker must solve are the DHCP or the DHDP [9].

The computational advantage of ECDH over DH is that it allows for selecting $q < p$. In the elliptic-curve case, field length ought to be only some hundred bits long

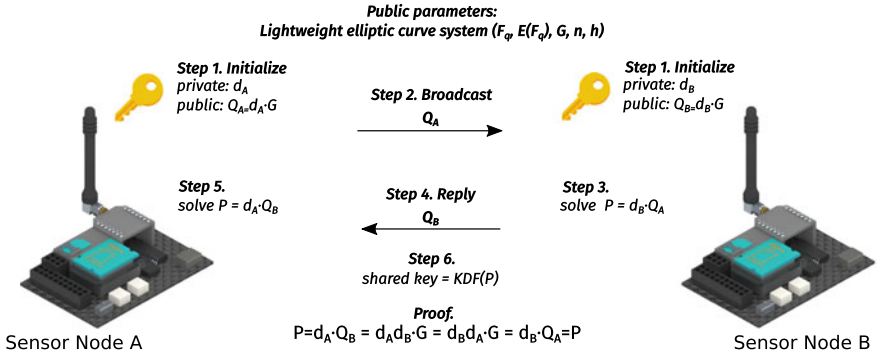

Fig. 2 Interaction diagram for the basic elliptic-curve Diffie-Hellman exchange (ECDH) protocol. In this scheme, parameters $\mathbb{F}_q$, $E(\mathbb{F}_q)$, and P are publicly known

for providing equivalent security to convectional DH instances, which would require a few thousand bits. This leads to performance improvement for scalar multiplication over modular exponentiations.

5.1 Problem of Authenticity

In the described key-establishment solutions, a critical concern is that users assume that the public keys they are receiving are legitimate. In the adversarial model employed in this Chapter, however, an attacker can take an active role in the channel. This can lead to man-in-the-middle-type attacks where one of the parties is impersonated. As stated in [10], "the real problem of key establishment is to exchange a key in an authenticated manner".

Network participants require some sort of information advantage to defeat active attackers. This is some data unknown to the adversary but shared by the exchange parties, a secret, or a way to verify the integrity of the message and the sender's authenticity–a tag. The first option brings us back to the main issue of key establishment in some kind of loop. The latter, as studied before, can be achieved with MAC functions, but these also employ a shared secret.

A popular approach is to offload the authentication problem to a third party that is trusted. This actor can either function as an auditor in the exchange or as a public registry of trusted parties.

In [10], the authors provided multiple examples of secure authenticated protocols. However, their scenarios supposed that a trusted actor was available for performing some of the computations or publishing an index of trusted parties. As mentioned before, our work did not make assumptions about the CPS network infrastructure.

The issue is the need to have a common piece of information agreed upon beforehand by the parties. Here, the main drawback of using a preshared secret is that,

if the secret is the same for each participant in the network, a single leak would compromise the security of the whole system. Conversely, a shared secret for each possible combination of participants would require massive storage capabilities in each device.

A solution approach proposed in the literature [24] is to employ a master session key for the establishment phase and discard it before time t has elapsed. This threshold is given by the expected time for an attacker to retrieve the master session key from a compromised device. This initial trust can then be used for building simple and efficient authenticated key-establishment protocols.

5.2 *Lightweight Authenticated Key Establishment*

In [24], the author described a lightweight key-establishment protocol for WSNs based on ECC. Their solution combined a conventional ECDH framework with the use of symmetric algorithms and a hash chain. The author claimed that the protocol is efficient, scalable, and elastic.

The protocol used an ephemeral master key as initial trust that facilitated the authentication of the parties. This key was combined with symmetric-cryptography algorithms for enhancing ECDH with mutual authentication. The employed hash chain was part of a node rejoin scheme that addressed the network variability of WSNs. Figure 3 illustrates the interaction diagram for this protocol.

In [24], the system model was that the network was single-hop, the nodes could communicate with each other, and the link was symmetrical. When both parties perform the same scale of computations, it can be said that the protocol is balanced.

The author proposed that there is a time threshold t, defined as the time required by an adversary to retrieve K_m from a captured node according to the current technology. That is, before t is elapsed, any node in possession of K_m is considered authentic.

The protocol has three steps:

1. Initialization. A shared key K_n is preloaded to each node. This is used as initial trust and represents the last element of a hash chain $K = \{K_1, K_2, \ldots, K_n\}$, where $K_{i+1} = H(K_i)$, and H is a hash function. A node that has K_n is considered secure within a timeframe t. Time t is derived from the required time for an attacker to retrieve keying materials from a captured node. During t, every node uses ECDH to link with other devices.
2. Key establishment. Two nodes use the initial key to perform pairwise key establishment. The work proposed to utilize two *modes* of operation, *new* and *old*. These serve as tags to indicate the type of security utilized in each message. When the key establishment is complete, all nodes should operate in *old* mode. In this phase, each node broadcasts a message that contains a security tag, sender ID, and an encrypted payload containing the sender ID, its public key, and the initial encryption key K_n. The advanced encryption standard (AES) was used to encrypt the message using the starting key K_n. Both, the ID on the header and the

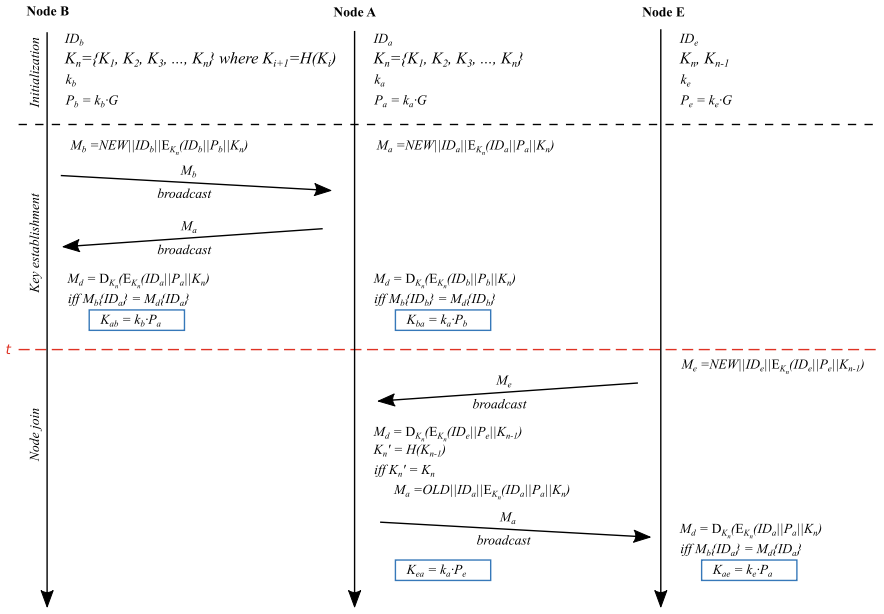

Fig. 3 Interaction diagram for the key-establishment protocol described in [24]

encrypted ID are used to authenticate the message. Once a pair of nodes exchange their public keys, they can establish a common secret using the ECDH.

3. Node join phase. When a new node tries to join the network, it broadcasts a message containing the security header, its ID, and an encrypted payload that contains its ID, its public key, and secret key K_{n-1}. The receiver verifies the new node by decrypting the message by using K_n, calculating $H(K_{n-1})$, and comparing this result with K_n. Once the identity of the joining node is verified, it is possible to establish a shared secret using ECDH.

5.3 Revisiting Ju's Protocol

The key issue with Ju's protocol lies in the provided authentication service. As stated in [10], the implicit authentication of encryption should not be used to replace message authentication codes. Moreover, in their work, only a small portion of the ciphertext was used for authenticating the sender. If the appropriate encryption mode is not employed, this can compromise the security of the system.

For key establishment by itself, encryption is not required when public-key algorithms are used. What is needed is a way to ensure that the received public key is authentic and that the integrity of the message is not compromised. A MAC function can be used for this end.

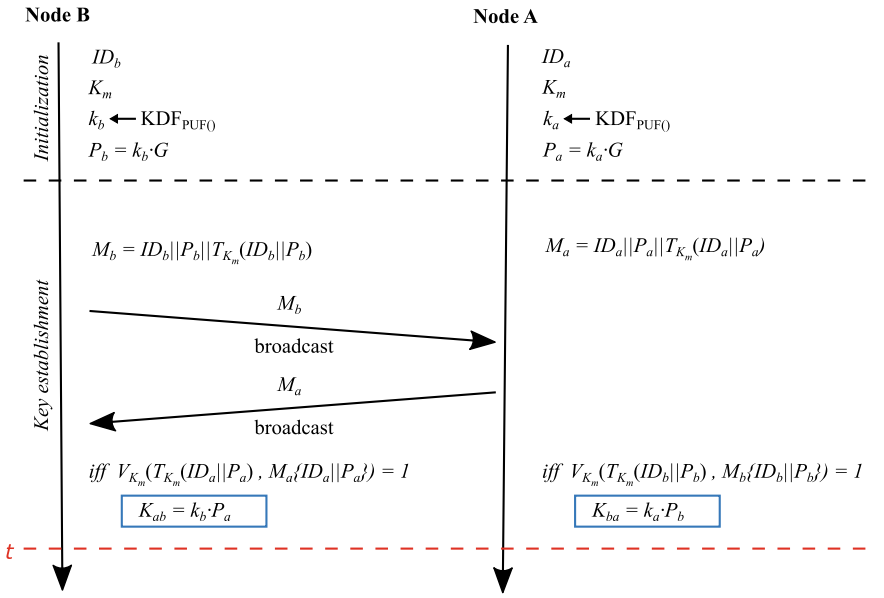

Fig. 4 Operation of the proposed key-establishment protocol based on Ju's work

Another improvement lies in enforcing the use of a key-derivation function (KDF) for obtaining the session key. In this case, a pseudorandom function (PRF) in the form of a PUF [25] was proposed. The use of a PUF prevents device cloning and impersonation attacks.

We found the broadcast system proposed in [24] is adequated for WSNs, as their topology is uncertain, and nodes should be able to collect multiple keys from any devices in their vicinity. The use of initial trust is an efficient way to ensure that MAC tags can be validated as long as t was not reached.

In our proposal, only the initialization and key-establishment steps were considered, whereas Ju's protocol has a node join phase. This phase was discarded since the proposed model does not consider that some new nodes could be introduced into the WSN.

The master key was not derived from a hash chain since it was not needed to recompute future master-key values. Once a time t had elapsed, the master key was discarded. Even if these data were eventually retrieved after t, this would not compromise the integrity of the network, as, at that point, the trust on this root would have expired.

The interaction diagram for the updated lightweight authenticated key-establishment protocol is provided in Fig. 4. This algorithm has an initialization phase when nodes compute their key pairs and a key-establishment phase when the network is formed.

5.3.1 Considerations

Each deployed node i has an identifier ID_i, a master key K_m, and a private key k_i, derived from a PRF. Each node computes a public key $P_i \in \mathrm{E}(\mathbb{F}_q)$ as $k_i \cdot G$ upon deployment. All participants possess the same information. Elliptic-curve domain parameters $\{\mathrm{E}(\mathbb{F}_q), G, n\}$ and description of MAC function $\{\mathcal{T}, \mathcal{V}\}$ are of public knowledge.

5.3.2 Steps

Based on the two first steps from Ju's protocol, the protocol consists of two steps: initialization and key establishment.

Initialization

Every sensor node i is loaded with an ID_i and an initial trust K_m. Each node derives a private key k_i from a PUF with associated public key $P_i = k_i \cdot G$.

Key Establishment

During this phase, the participants perform two key tasks. First, they construct a message containing their ID, their public key, and the MAC for these two values. There is no need to provide confidentiality for the payload since none of these data is secret. These messages are then broadcast to any device on their neighborhood. The second task consists of listening to the channel for incoming broadcasts. The receiver must verify the authenticity and integrity of these messages by means of the accompanying MAC tag. The security of this scheme relies on the secrecy of K_m up to t.

When authentication is successful, the receiver device generates a session key for the device with ID_i, and indexes this session key and their public key in an IDs directory. Logically, if the incoming-broadcast authentication was successful, then the sender device should have followed the same steps and indexed the receiver. This can be corroborated with an acknowledgement message per common network operation (ACK).

5.4 Security Analysis of Proposed Elliptic-Curve Protocol

The use of a symmetric component enhances a conventional ECDH and results in an efficient and simplified design. Here, K_m acts as a source for authentication, preventing man-in-the-middle and denial-of-service (DoS) attacks during network formation. These are two critical ECDH problems.

Key-establishment protocols that completely rely on symmetric components are vulnerable to node capture. This is addressed by using ECDH and discarding K_m after t has elapsed. Any captured node, by definition after t, does not compromise

the network, as the only retrievable information by an attacker is at most a small index of session keys.

The use of a PUF as a precursor for the private key of the node provides additional security protections against physical attacks like cloning. Enhanced security can be obtained by using the PUF value and a KDF for creating the private keys.

Even though the use of a master key with a time-bound t provides some advantages for creating a lightweight protocol, this time t also prevents further exchanges to update the session key. To obtain forward security, the devices should adopt a refreshment system in order to update the session key.

6 Key Establishment in the Post-quantum World

Up to this point, the reviewed key-establishment solutions are considered secure on the difficulty of computing discrete logarithms with appropriate restrictions. This is a challenging problem for classical computers. However, that is not the case if quantum processors are involved.

In 1997, Peter Shor published quantum algorithms for computing prime factorization and discrete logarithms in polynomial time [26]. The main implication of that work is that a significant part of modern cryptography will become obsolete if a large enough quantum computer is built [27]. Banking, government, healthcare, commerce, and virtually any application deployed over the Internet would be affected. As stated in [28], cryptography has entered a race against time to adapt to this new threat. Adapting cryptography for resisting quantum attacks while maintaining low-enough overheads for constrained CPSs is a particularly difficult task.

The extent of the power of quantum computing is an open discussion. Theoretical understanding of quantum algorithms and their application to classical problems have only started receiving attention in the past decade. As a result, the reach of applications for quantum computers is still unclear. One of the few points of agreement is that quantum computing is believed to be unable to solve classical NP-complete problems. Nonetheless, quantum computers can solve problems that were believed to be unsolvable in polynomial time, such as DLP and ECDLP.

This has prompted the question of whether authenticated key-exchange protocols exist that are tailored for constrained environments that are not vulnerable to potential quantum adversaries. So far, the answer has been no. This issue is addressed in the following.

6.1 Proposed Approach

In classical cryptography, the key agreement is achieved thanks to the Diffie-Hellman key exchange (DH) and its variations. The main trait of this algorithm is to allow for two parties to establish a shared key with equal contributions in a way in which

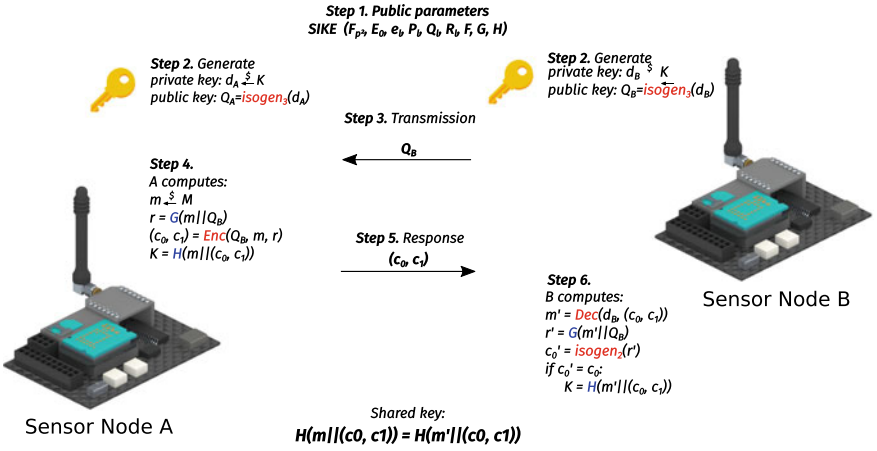

Fig. 5 Interaction diagram for the supersingular isogeny key-encapsulation (SIKE) algorithm

neither party could individually predict the resulting shared secret. However, very few quantum-resistant algorithms have the required commutability for creating DH-like constructions. Only the ding key exchange [29] and systems reliant on isogenies between supersingular elliptic curves [30, 31] offer this advantage. However, the security understanding of these constructions is still limited.

Key-encapsulation mechanisms (KEMs) are systems proposed for achieving key establishment with the use of public-key-encryption (PKE) algorithms. The Fujisaki-Okamoto (FO) [32, 33] and the Hofheinz-Hövelmanns-Kilts (HHK) [34] transforms are two constructions that were conceived for this end. The main characteristic of these systems is that they allow for converting a CPA-secure PKE into a CCA-secure KEM with tight security—see [35] for the description of these security notions. The limitation of these solutions is that, compared with DH-like exchanges, only one of the parties is responsible for creating the session key. This secret is then encapsulated and transmitted to the second party.

The supersingular isogeny key-encapsulation (SIKE) suite proposes a CPA-secure PKE system and then uses a variation of the HHK transform for obtaining a CCA-secure KEM [36]. Their modification of the HHK construction allows for reducing the complexity of the final validation step in the KEM. Figure 5 illustrates the key-establishment procedure of a SIKE KEM.

In the protocol from Fig. 5, Node A was entrusted to generate a secure session key m. Additionally, Node A assumed that the public key received from Node B was authentic, as no additional checks were performed. General applications can employ standardized authentication techniques or rely on trusted parties for corroborating the authenticity of the public key and its sender. However, constrained devices cannot afford to implement such solutions. This is a problem that has so far not been addressed in the literature.

The CCA security of SIKE KEM allows for reusing a public key in multiple exchanges without additional vulnerabilities due to reaction attacks. This can be advantageous for constrained devices since the public key can be calculated offline and then stored in the device. Furthermore, the public key does not need to be protected and can be stored in external memory.

6.2 Protocol Design

A modification of SIKE for obtaining an authenticated key exchange with mutual key derivation is proposed. This protocol is illustrated in Fig. 6.

The protocol is composed of two main steps:

1. Initialization. This process can be carried out offline, each device is assigned an ID, a master key computes a secret key from a physically unclonable function, and uses this secret key to obtain a public key with the public generators and base curve of SIKE.

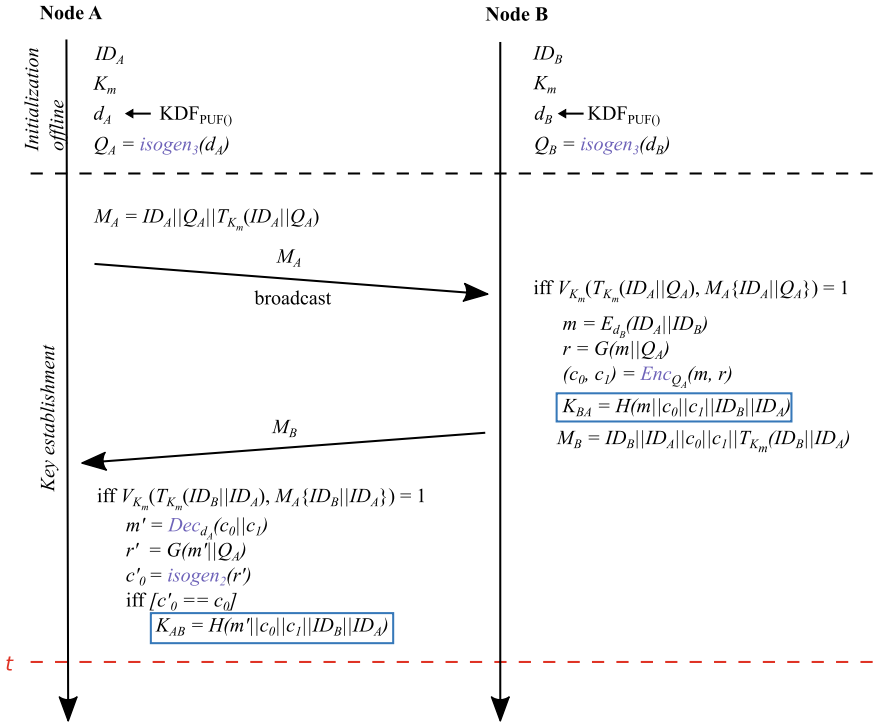

Fig. 6 Key establishment achieved with the proposed protocol. In this scheme, functions in purple are those specified in SIKE. The shared key is derived from a hash computation

2. Key establishment.
   - A device broadcasts a message containing its ID, its public key, and a MAC tag generated using the master key. This serves to authenticate the device with nearby devices. Unlike DH-like variants, SIKE follows a challenge/response approach; thus, the protocol ought to be performed once for each pair of participants.
   - The node that receives the broadcast authenticates the message with the MAC. If verification is successful, the receiver performs the encapsulation of the shared secret; this shared secret or session key is derived as the ciphertext of both devices' IDs employing the private key of the receiver. The issuer generates a new message with the ciphertext resulting from SIKE encapsulation and the respective MAC tag. The session key for the participant is generated as a side product of the encapsulation.
   - The broadcast issuer, upon receiving a reply, verifies its integrity with the corresponding MAC. It then decapsulates the secret and verifies its authenticity through partial re-encryption. If the SIKE ciphertext is valid, the device computes the session key. The new node is then authenticated and starts issuing a broadcast to allow for more nodes to join the network.

This protocol provides mutual authentication and key agreement for any pair of devices in the network. The authentication of the system relies on the difficulty of forging a MAC tag or a forensics attack for recovering the master key. Considering that the fastest of these procedures require a time t, it follows that the scheme is secure up to t. During this time, the network should be consolidated.

The proposed protocol requires encapsulation and decapsulation functions from the SIKE specification. These functions use the underlying public-key encryption scheme specified in [36]. In these procedures, the core functions perform the computation of public keys ($isogen_\ell$) and shared keys ($isoex_\ell$). These are the most expensive operations, and two of each are performed in the envisioned protocol.

6.3 Security Analysis of Proposed Post-quantum Protocol

First, systems based on supersingular isogenies can offer commutability. So, in principle, it is possible to create a Diffie-Hellman-like key exchange. Such an algorithm exists and is described in [30]. This SIDH algorithm allows for two parties to obtain shared secrets that are derived by using information from both participants. For this reason, it can be classified as a dynamic key-agreement protocol. However, the security of SIDH is limited to the CPA scenario. The main implication for a device using this algorithm is that the public key must be renewed for each new session. This involves additional storage and processing costs that are detrimental to constrained CPSs.

By employing a transformation derived from Cramer-Shoup due to [33, 34], SIDH can be transformed into the CCA-secure KEM known as SIKE. In this process, the

cryptosystem acquires adaptive security under the random oracle model at the cost of becoming a key-encapsulation system. This implies that SIKE is a dynamic key-transport protocol, which might be vulnerable to key-generation faults.

The first aim of the proposed enhanced SIKE is to restore the *key-establishment* characteristic of the protocol, that is, the session key is derived with contributions from both parties. For this, taking Fig. 6 as a reference, party B derives the session key m as the result of encrypting the identities of both parties under its secret key; here, assume that B acts in good faith. In the SIKE specification, m had a length of 128 and 256 bits, which had a good relationship with the block length of most standardized ciphers.

The identity value is recommended to be at least equal to cipher block size c, so that at least two cipher blocks are processed; by doing so, the protocol is resilient against birthday attacks. The security of m relies on the strength of the selected cipher with an appropriate confidentiality mode behaving as a PRF. The length of the proposed m is then $2c$, which poses a challenge for SIKEp434, where $m = 128$ if $c = 128$. This does not affect the calculation of shared key K, since it is the result of a hash but must be considered on deriving ciphertext c_1; truncating m is not advised, so the implementer has the choice to employ an additional hash for reducing m to the appropriate length, or compute $c_1 = h \oplus m_h \oplus m_l$, where $m = m_h||m_l$. The identity values are also included in deriving the session key K by hashing; this part enforces that both parts act in good faith.

The second enhancement confers SIKE with the mutual authentication feature. In the general Internet scenario, authentication servers and trusted parties are readily available to validate the authenticity of a public key and the integrity of a message. However, in the envisioned application scope, relevant to constrained CPSs such as WSNs, assumptions regarding network infrastructure cannot be made. Hence, parties should be able to authenticate each other by themselves.

This is achieved by employing the ephemeral-master-key strategy from [24]. Every exchanged message during the key-establishment stage of the protocol carries a generated MAC using the ephemeral master key. This MAC function can be implemented by using the main block cipher of the device under an appropriate authentication mode to improve the efficiency of the system. This MAC must exhibit unforgeability and collision resistance under the Chosen Message Attack model so that the exchanged public keys and identities are trusted. In the broadcast reply, the MAC tag does not cover the SIKE ciphertext. This is done for efficiency reasons since ciphertexts alone are already authenticated by the partial re-encryption of SIKE.

Since the ephemeral master key is not used for providing confidentiality, the issue of forwarding secrecy does not need to be addressed. However, network elasticity is restricted, since no more nodes are allowed to join after t; this also implies that drastic changes in the network topology might compromise WSN operation capabilities.

Although the initial topology of a WSN is not given, it does not usually change. Other types of networks better represent problems associated with mobile targets, for example, vehicular ad hoc networks (VANETs). The main source of topology

disruption can be attributed to reallocation attacks, but it can be argued that, if the attacker could access a large enough number of nodes, these would be subtracted rather than relocated. The use of PUFs can deter any attempts of sequestering or cloning the nodes.

6.4 Application Scope

The post-quantum protocol proposed in this section is aimed at filling a niche where constrained CPSs require long-term security. Even with the most optimistic forecasts for the development of real large-scale quantum computers, we are looking at a good decade-long window where modern PKCs would remain secure. However, as mentioned before, the concern lies in those applications whose data need to remain safe for longer periods.

Some of these applications include healthcare monitoring, which protects personal data, vehicular networks where exchanged messages within the network contain proprietary information critical to the product, and mobile military networks where the transmitted information by devices can be classified to protect national security interests. In these scenarios, we are looking at a good 20–50-year window where information must remain secure.

Arguably, devices used in these applications exist on the high-end profile for CPSs, but they are still bound by performance and energy constraints. The availability of solutions that can work standalone without a given topology and offer long-term security is critical for protecting sensitive data with due care and diligence. Herein lies the relevance of our work.

7 Conclusions, Final Remarks, and Future Work

The main goal of CPSs lies in connecting the cybernetic and physical worlds. These technologies offer significant advantages for applications of the management and control of public infrastructure, distribution systems, supervision of remote tasks, and healthcare. Therefore, they are intricately connected with the human world. Any data being collected, processed, and transmitted by interconnected devices must be safeguarded. This is a difficult task for constrained CPSs such as WSNs. One of the most effective approaches for ensuring information security is cryptography, which commonly relies on the use of cryptographic keys. Thus, key establishment is the main component when securing current and future CPS applications by employing cryptographic algorithms.

In this chapter, three alternatives of two-party, balanced key-establishment protocols for constrained CPSs were described. The solutions under study were analyzed under fair assumptions, and they rely on proven cryptographic principles. Two elliptic-curve-based solutions that are simple and efficient for solving the problem at

hand were first reviewed. We revised the appropriateness of using these systems in the envisioned application scope of WSNs and proposed improvements for enhancing the security of an initial solution of interest in order to derive the second algorithm. We then addressed the possibility of a threat model involving quantum adversaries by proposing a novel key-establishment protocol that inherits the efficiency enhancements of ECC-based solutions but employs quantum-safe cryptographic algorithms.

The work presented here is a first in the area of security in constrained environments for modern computing scenarios and needs further study to corroborate the pertinence of the assumptions and security claims required for constrained CPSs.

Multiple challenges and opportunities can be addressed in future work. First, while it was shown that the proposed algorithms are correct, and informal security assumptions were claimed, it is necessary to demonstrate that the proposed protocols are secure through formal analysis. Second, it is necessary to quantify the operational costs for these solutions to delimit the prospective application domains where they can be used. Lastly, efficient realizations of these algorithms need to be obtained so they can be implemented in actual CPS applications.

Acknowledgements This work was supported by CONACyT under grant number 336750 and CINVESTAV. This work was also funded by "Fondo Sectorial de Investigación para la Educación", CONACyT México, through the project number 281565.

References

1. Henze, M., Hiller, J., Hummen, R., Matzutt, R., Wehrle, K., Ziegeldorf, J.H.: Network Security and Privacy for Cyber-Physical Systems, pp. 25–56. Wiley (2017)
2. Frahim, J., Pignataro, C., Apcar, J., Morrow, M.: Securing the Internet of Things: A Proposed Framework. Technical report, Cisco Security (2012). https://tools.cisco.com/security
3. Wang, Y., Nikolai, J.: Key Management in CPSs, pp. 117–136. Wiley (2017)
4. Ziegeldorf, J.H., Morchon, O.G., Wehrle, K.: Privacy in the Internet of Things: threats and challenges. Secur. Commun. Netw. **7**(12), 2728–2742 (2014)
5. IEEE: IEEE Standard for Low-Rate Wireless Networks. IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), pp. 1–709 (2016)
6. Menezes, A.J., Vanstone, S.A., Oorschot, P.C.V.: Handbook of Applied Cryptography, 1st edn. CRC Press, Inc., Boca Raton, FL, USA (1996)
7. Jilna, P., Deepthi, P.P.: Light Weight Key Establishment Scheme for Wireless Sensor Networks, pp. 124–137. Springer International Publishing, Cham (2016)
8. Yang, Y., Lu, J., Choo, K.-K.R., Liu, J.K.: On Lightweight Security Enforcement in Cyber-Physical Systems. In: Güneysu, T., Leander, G., Moradi, A. (eds.) Lightweight Cryptography for Security and Privacy, pp. 97–112, Springer International Publishing, Cham (2016)
9. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, New York Inc., Secaucus, NJ, USA (2003)
10. Goldwasser, S., Bellare, M.: Lecture Notes on Cryptography (July 2008). https://cseweb.ucsd.edu/~mihir/papers/gb.pdf
11. Shafi, Q.: Cyber-Physical systems security: a brief survey. In: 2012 12th International Conference on Computational Science and Its Applications, pp. 146–150 (2012)
12. Zhang, Y., Xu, L., Xiang, Y., Huang, X.: A matrix-based pairwise key establishment scheme for wireless mesh networks using pre deployment knowledge. IEEE Trans. Emerg. Top. Comput. **1**(2), 331–340 (2013)

13. Boubakri, W., Abdallah, W., Boudriga, N.: Chaotic ZKP based authentication and key distribution scheme in environmental monitoring CPS. In: Sabir, E., García Armada, A., Ghogho, M., Debbah, M. (eds.) Ubiquitous Networking, pp. 472–483, Springer International Publishing, Cham (2017)
14. Zhang, Y., Xiang, Y., Huang, X.: A cross-layer key establishment model for wireless devices in Cyber-Physical systems. In: Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security, CPSS'17, pp. 43–53. Association for Computing Machinery, New York, NY, USA (2017)
15. Mathur, S., Miller, R., Varshavsky, A., Trappe, W., Mandayam, N.: ProxiMate: proximity-based secure pairing using ambient wireless signals. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, MobiSys'11, pp. 211–224. Association for Computing Machinery, New York, NY, USA (2011)
16. Giri, N.K.: A Dependable and Secure Approach for Secret Key Establishment and Operation in Automotive CPS. Master's thesis, Kansas State University, Manhattan, Kansas (2018)
17. Giri, N.K., Munir, A., Kong, J.: An integrated safe and secure approach for authentication and secret key establishment in automotive Cyber-Physical systems. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) Intelligent Computing, pp. 545–559. Springer International Publishing, Cham (2020)
18. Zhang, J., Li, H., Li, J.: Key establishment scheme for Wireless Sensor Networks based on polynomial and random key predistribution scheme. Ad Hoc Netw. **71**, 68–77 (2018)
19. Challa, S., Das, A.K., Gope, P., Kumar, N., Wu, F., Vasilakos, A.V.: Design and analysis of authenticated key agreement scheme in cloud-assisted Cyber-Physical Systems. Futur. Gener. Comput. Syst. **108**, 1267–1286 (2020)
20. Chaudhry, S.A., Shon, T., Al-Turjman, F., Alsharif, M.H.: Correcting design flaws: an improved and cloud-assisted key agreement scheme in Cyber-Physical Systems. Comput. Commun. **153**, 527–537 (2020)
21. Farhdi Moghadam, M., Mohajerzdeh, A., Karimipour, H., Chitsaz, H., Karimi, R., Molavi, B.: A privacy protection key agreement protocol based on ECC for smart grid, pp. 63–76. Springer International Publishing, Cham (2020)
22. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**, 644–654 (1976)
23. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) Advances in Cryptology-CRYPTO'85 Proceedings, pp. 417–426. Springer, Berlin, Heidelberg (1986)
24. Ju, S.: A lightweight key establishment in Wireless Sensor Network based on Elliptic Curve Cryptography. In: 2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment, pp. 138–141 (July 2012)
25. Maes, R.: Physically Unclonable Functions: Constructions, Properties and Applications. Springer Publishing Company, Incorporated (2013)
26. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**, 303–332 (1999)
27. Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on post-quantum cryptography. Technical report, National Institute of Standards and Technology (2016). http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf
28. Bernstein, D.J., Lange, T.: Post-Quantum cryptography. Nature **549**, 188–194 (2017)
29. Ding, J., Takagi, T., Gao, X., Wang, Y.: Ding key exchange. Technical report, National Institute of Standards and Technology. NIST Post-Quantum Cryptography-Round 1 Submissions (2017)
30. Jao, D., De Feo, L.: Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies, pp. 19–34. Springer, Berlin, Heidelberg (2011)
31. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. Cryptology ePrint Archive, Report 2018/383 (2018)
32. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) Advances in cryptology-CRYPTO'99, pp. 537–554. Springer, Berlin, Heidelberg (1999)

33. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptol. **26**, 80–101 (2013)
34. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) Theory of cryptography, pp. 341–371. Springer International Publishing, Cham (2017)
35. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for Public-Key Encryption Schemes. In: Krawczyk, H. (ed.) Advances in cryptology-CRYPTO'98, pp. 26–45. Springer, Berlin, Heidelberg (1998)
36. Jao, D., Azarderakhsh, R., Campagna, M., Costello, C., Feo, L.D., Hess, B., Jalali, A., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Renes, J., Soukharev, V., Urbanik, D.: Supersingular Isogeny Key Encapsulation. Technical report, National Institute of Standards and Technology. NIST Post-Quantum Cryptography-Round 1 Submissions (2017)