IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 24, 2018, accepted November 6, 2018, date of publication November 15, 2018,
date of current version December 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881444

Elliptic Curve Lightweight Cryptography: A Survey

CARLOS ANDRES LARA-NINO[1], ARTURO DIAZ-PEREZ[2], AND MIGUEL MORALES-SANDOVAL[1]
[1]CINVESTAV Tamaulipas, Ciudad Victoria 87130, Mexico
[2]CINVESTAV Guadalajara, Zapopan 45019, Mexico

Corresponding author: Carlos Andres Lara-Nino (clara@tamps.cinvestav.mx)

This work was supported in part by CONACyT under Grant 336750, in part by CINVESTAV, in part by Fondo Sectorial de Investigación para la Educación, CONACyT Mexico, under Project 281565.

ABSTRACT Since it was invented in 1986, elliptic curve cryptography (ECC) has been studied widely in industry and academy from different perspectives. Some of these aspects include mathematical foundations, protocol design, curve generation, security proofs, point representation, algorithms for inherent arithmetic in the underlying algebraic structures, implementation strategies in both software and hardware, and attack models, among others. The main advantage of ECC is that shorter keys (less-memory requirements and faster field arithmetic operations) can be used if compared with other cryptosystems, which has made it the ideal choice for implementing public key cryptography in resource constrained devices, as the ones found in the envisioned applications of the Internet of Things, e.g., wireless sensors. In this application domain, lightweight cryptography has emerged as the required one because of the scarce computing resources and limited energy in devices. In this paper, we present a survey of ECC in the context of lightweight cryptography. The aim of this paper is to identify the criteria that make an ECC-based system lightweight and a viable solution for using in practical constrained applications. Representative works are systematically revised to determine the key aspects considered in ECC designs for lightweight realizations. As a result, this paper defines, for the first time, the concept and requirements for elliptic curve lightweight cryptography.

INDEX TERMS Cryptography, elliptic curve, lightweight, survey.

I. INTRODUCTION

In recent years, the trend in manufacturing of electronic devices has been marked by the reduction of their physical size, the push to reduce production costs, and the increment of the connectivity of said appliances. *Smart* objects, which are capable of performing small computations and data collection, become more ubiquitous each day. All of the information which is collected from these objects can provide insights on the behavior of its user or its environment. Hence the need to protect these data.

A constrained environment is considered a computational system of multiple heterogeneous elements, where the underlying computational devices are of limited capabilities. These limitations are related to the processing power, the communications bandwidth, the storage memory, the size of the device, or the energy availability of the devices. Examples of constrained devices are the *Wireless Sensor Network* (WSN) motes, *Radio-Frequency Identification* (RFID) tags, and *Internet of Things* (IoT) nodes, thus WSN, RFID, and IoT applications are considered constrained environments.

Providing security services for these new generation networks has proven to be a difficult challenge. Strict constraints on resources such as processor time, bandwidth, hardware, and in some cases energy supply, restricts the security algorithms that can be utilized. These applications require at least the same security services of a conventional network, even having less processing power. Moreover, like in the case of WSN, the constrained devices might be deployed in hostile environments and an attacker can have physical access to the network. Additional security measures, such as side channel countermeasures, should be considered to patch these vulnerabilities.

A. CRYPTOGRAPHY FOR CONSTRAINED DEVICES

Some of the most important security services required in IoT include privacy (confidentiality), trust (integrity, authentication), nonrepudiation (signature, access control), and availability. For some instances, protection against a node capture, impersonation, duplication of the data, and forensic attacks is also required. Cryptography can provide the means for most of the demanded security services in this domain.

However, as pointed out by NIST "[...] cryptographic standards were designed to perform well on general-purpose computers" [1]. But modern technologies have capabilities far more limited than general-purpose computers. As it is also mentioned in the NIST report, it is often the case that conventional cryptographic algorithms can be implemented

to fit the resource requirements of constrained applications. But this comes at the cost of reduced performance and lower efficiency. It can be inferred that conventional cryptography can be used to provide security services under constrained environments, but it might not be the best solution.

The push for newer algorithms which satisfy the security needs of IoT and other such systems has created a new branch in cryptography denominated Lightweight Cryptography. Its aims are to provide privacy, integrity and trust among other services, by using symmetric and asymmetric mechanisms, but taking into consideration reductions in the implementation sizes, the processing latencies, and the energy costs of the solution. This field is an evolving study area, not only the security paradigms change, new attack models are revealed, and security levels are phased out.

Lightweight cryptography is related to the problem of providing security to constrained environments by means of low-cost cryptographic algorithms. It is the set of tools designed to offer security services with reduced costs. The optimizations followed in this approach generally involve tradeoffs between implementation size, performance, and security.

It results difficult to determine a threshold value for a cryptographic primitive to be denominated *lightweight*, with reference to one or more metrics of interest: physical size, latency, energy. Take as example the literature for cryptographic hardware solutions where lightweight symmetric algorithms have shown to require an order of magnitude less area units than their generic counterparts at the cost of reduced performance. Current understanding of ''lightweightness'' also includes performance and energy as critical design goals. Designing security solutions that can be denominated ''lightweight'', while achieving small implementation size, low energy consumption, and adequate performance is a challenging task.

In the past decade, the study of *lightweight* cryptographic primitives has gained popularity. First initiated with focus on block ciphers and later on hash functions, this field of research has propelled the development of multiple algorithms, some of which have been standardized [2]. Most recently, the focus of lightweight cryptography has translated from symmetric to asymmetric constructions. However, whereas the progress for the former has been steady and fruitful, the latter has found moderate success.

Some of the challenges in lightweight asymmetric algorithms are the complexity of the operations, the size of the operands, the lengthy delays in processing, and the relentless advance of attack models which threaten any hasty proposal.

B. LIGHTWEIGHT PUBLIC KEY CRYPTOGRAPHY

Keyed cryptographic algorithms are those requiring some secret material (key) to protect the data. They can be divided in two main groups: symmetric and asymmetric. The first group includes all the systems which use a single secret key in their operation. From the key nature it follows that this area is known as *Symmetric Cryptography*. The algorithms that form the second group use a key pair instead, where one of the keys is *private* and the other *public*. The key pair is generally created from a mathematical function that establishes a relation between the private and the public key, but with special properties to avoid deriving the private key from the public one. This second group of algorithms constitutes what is known as *Asymmetric Cryptography* or *Public Key Cryptography* (PKC).

PKC is critical for networked environments. It has been used in encryption, signatures, digital envelopes and key establishment to provide confidentiality, integrity, authentication, nonrepudiation, availability and access control services. PKC is a costly security mechanism especially for constrained devices. Encryption and digital signatures in PKC demand complex group operations. The operands used in these procedures can have lengths of thousands of bits in some cases. From the different PKC alternatives reported in the literature, those that rely on elliptic curves are the most favorable for implementation in restricted devices.

Elliptic Curve Cryptography (ECC) utilizes an elliptic curve defined over a finite field $\mathbb{F}_q$, which is denoted by $E(\mathbb{F}_q)$ and contains the affine points $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ that satisfy the Weierstrass equation (1).

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6, \quad \text{where } a_i \in \mathbb{F}_q \quad (1)$$

$E(\mathbb{F}_q)$ together with a special point named the point at infinity $\mathcal{O}$ form an abelian group. $\mathcal{O}$ serves as the neutral element in the group operation. The security of ECC rests on the difficulty of solving the Discrete Logarithm Problem over such a group, abbreviated as ECDLP, for which only algorithms with exponential computational complexities are known.

It is not an easy task to find a group $E(\mathbb{F}_q)$ with the required properties that make the ECDLP difficult to solve. Hence, the elliptic curves known to be secure are reported in the literature and included in standards. Conventional PKC based on ECC generally uses these standardized structures, which define secure realizations of ECC but that were not thought to be utilized in constrained environments. In recent years, the definitions for new elliptic curves not only seek to achieve high security levels, but also to reduce operational costs and to reduce the hardware resources required to perform computations efficiently. These new elliptic curves are left out of the scope of standards but are an attractive alternative for WSN, RFID, *e*-health, and other emerging technologies in the IoT domain.

In this work we address ECC solutions that are suitable for constrained applications and therefore denominated lightweight. In the literature is has not been demonstrated if lightweight realizations of ECC are due to 1) the underlying mathematical model, or 2) the design and implementation decisions. The review presented in this work provides insights on which elliptic curves are the most used or considered for a lightweight ECC implementation. We ultimately identify the characteristics of the primitives and requirements that shape the *Elliptic Curve Lightweight Cryptography* (ECLC) concept and provide guidelines for the future development of such systems. Figure 1 shows a wordcloud of keywords associated with ECLC.

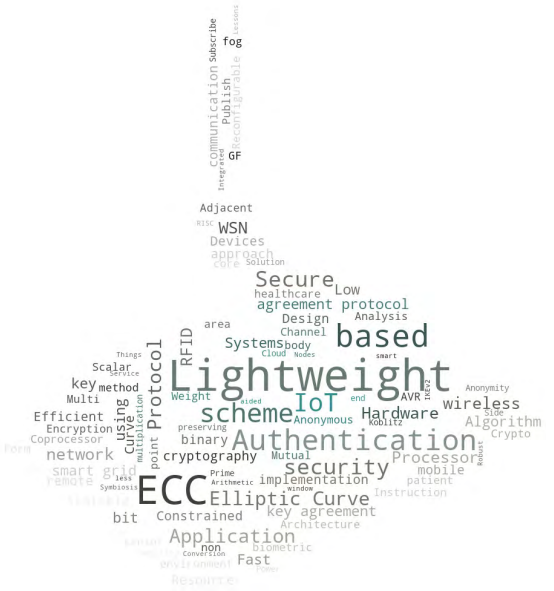

FIGURE 1. Wordcloud for Elliptic Curve Lightweight Cryptography (ECLC).

C. ECLC SYSTEM AND SECURITY MODEL

ECLC can be useful for emerging technologies in the IoT paradigm such as in RFID, WSN, *e*-Health, Smart Grid, and others. It differentiates from ECC mainly in the fact that ECLC realizations must exhibit awareness of the application constraints and accommodate such requirements accordingly. Such constraints can be grouped broadly as performance, size, energy, and security. ECLC is similar to ECC in that the former ought to preserve the same security features as the latter, and thus it can be used for implementing comparable security services.

The cryptographic strength of both, ECC and ECLC, relies on the hardness of the discrete logarithm problem. This notion, paired with security protocols can be used for providing services of key establishment, encryption, authentication, and signatures. These security solutions can then be implemented in constrained environments as the aforementioned for applications in healthcare, military, rescue, security, among others.

Figure 2 illustrates the use of ECLC for providing key agreement using WSN as case study. We would like to remark that this is just an instance of the many technologies that can benefit from its advantages. In a classical WSN model, sensor nodes are distributed in environments of difficult reach; the information harvested by these nodes is collected by a base station with internet connection and the end users obtain information from the base station through the internet. A characteristic of sensor nodes is that they suffer from critical constraints in regards to performance, size, and energy. In order to achieve secure communications in a wide-area, the nodes must be able to establish multi-hop links. ECLC can enable the nodes to link up with each other with low processing and storage costs. In this example from Figure 2, a basic ECDH-like protocol is described; in an

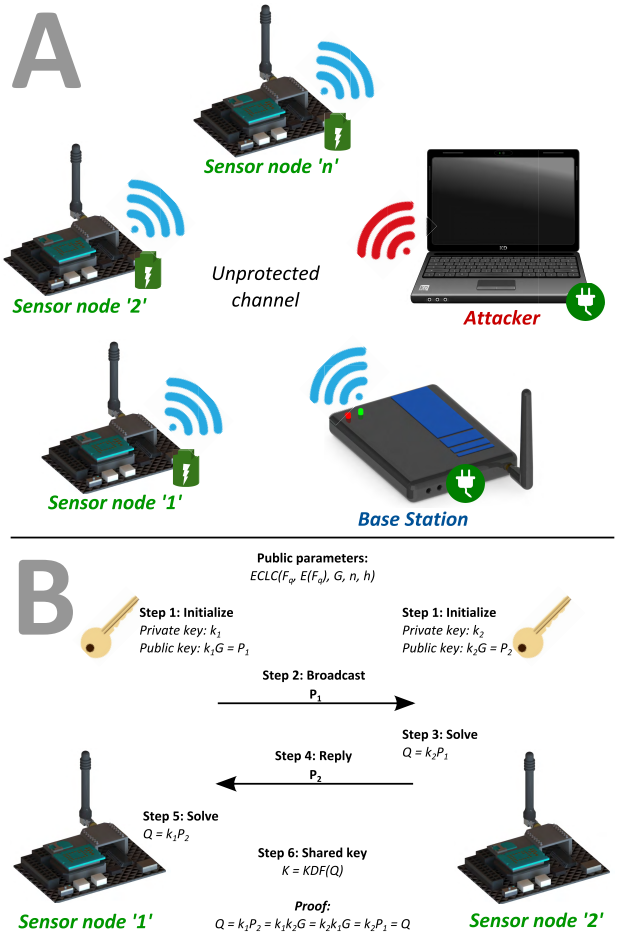

FIGURE 2. Use of ECLC for providing key establishment in the context of WSN: (A) System model. (B) Security model.

ECLC solution, the protocols, algorithms, and realizations must be aware of the nodes constraints.

Any ECLC solutions ought to be as secure as its ECC counterparts. The hardness of ECLC still relies on the difficulty of finding discrete logarithms over elliptic curve groups. The order of the group will be directly related with the security of the solution, as well as with the complexity of evaluating the group operations. Improving the performance, the size, or the energy consumption of an ECLC realization should not compromise the security of the system. Therefore the underlying field size should be defined according to recommended ECC security levels [3].

Under certain scenarios, however, the designer might determine that the information being protected does not require long term security. One particular example are WSNs which, by definition, are only meant to last from a few weeks to months. In this instance, using an elliptic curve group which guarantees security for thousands of years might not be required. By using smaller fields performing group operations becomes cheaper, as mentioned. Such idea has been explored in [4]. Nonetheless, the impact of disclosing data which on first sight might appear innocuous is difficult to assess: undoubtedly, nowadays information has a high value.

ECC is a set of tools to provide security. To extend this definition to lightweight cryptography, it follows that the security is to be provided for constrained environments.

Then, we can understand lightweight cryptography as the set of tools designed to provide security while observing the application constraints. If these tools are based on elliptic curve constructions, then

Definition 1: ECLC is the set of elliptic curve protocols, domain parameters, algorithms, and implementation techniques, tailored for providing security services under constrained environments.

ECLC is a novel concept. The attribute of lightweightness has been vested upon ECC-based constructions in multiple instances, as the survey evidences. Nonetheless, nobody has sought to answer two fundamental questions: what is lightweight in the context of ECC, and how to assess that an ECC-based system is lightweight. For some, these questions might appear trivial; however, reaching a common understanding and reference frame is a key point of science. Our innovation lies on exploring ECC in the context of lightweight cryptography for providing answers for these questions.

D. PROBLEM STATEMENT

In this survey, for the first time, we survey works which use elliptic curve cryptography are denominated lightweight. In a sense, this is a survey on ECC in the context of lightweight cryptography. This is one of the main differences between our work and previous ECC surveys.

Other significant differentiations which can be made between our work and previous ECC surveys is the depth and extension of our work. In this document we provide preliminaries, a state of the art review, novel concepts, a summary of strategies, and in general an extensive analysis of the surveyed data, qualitatively and quantitatively.

The quick review of previously published ECC-related surveys provided in Table 1 can highlight some of their general limitations. It is possible to find many more unpublished works but most of them suffer from lack of scope, scientific character, and extension. A particular exception is the work in [5] where hardware realizations of ECC are covered with sufficient detail.

Our research aims at providing answers, in the broad sense, for two main questions: What is lightweight in the context of ECC? and how to determine if an ECC-based security solution is lightweight?

From the discussion provided in this section we also identify additional research questions which are, as far as we know, yet open:

- If there are elliptic curves designed to fit the needs of constrained devices, is it possible to call such curves lightweight?
- Is the "lightweight" adjective exclusive for the implementation of these systems?
- If the underlying ECC constructions are not standardized, will there be any traction on implementing them?
- What are the strategies used in the design and implementation of lightweight security solutions based on ECC?

The knowledge derived by answering the above questions can serve to build novel lightweight security solutions for IoT and related applications.

TABLE 1. Brief summary of some published ECC-related surveys.

Year	Work	Ref.	Scope	Pages	References
2006	Dormale and Quisquater	[6]	High speed/ Hardware	13	80
2011	Kalra and Sood	[7]	Introductory/ Security	5	25
2014	Sakharkar et al.	[8]	Routing/ Security	5	17
2017	Harkanson and Kim	[9]	Introductory/ Applications	7	30

E. CONTRIBUTIONS OF THE SURVEY

This work has five main contributions:

1) Surveys, for the first time, ECC in the context of lightweight cryptography.
2) Quantifies the State of the Art bounds for ECC-based systems which are denominated lightweight.
3) Determines the criteria for ECC-based systems to be denominated lightweight.
4) Proposes a design methodology that guides the realization of ECLC solutions.
5) All the tables and graphs data are available in a public repository[1] as well as in IEEE DataPort.[2]

F. HOW TO READ THIS SURVEY

Our survey first provides introductory notions so that researchers which are first approaching the topic of ECC can finds some pointers on mathematical fundamentals, suggested literature, and in general concepts and applications of ECC; these are included in Section II. Readers well versed on ECC from a general point of view can skip these preliminaries.

The data collection techniques, our categories for classifying the papers, and the data extracted from the different works are reported in Section III. In this part we go in great detail to describe the surveyed data since some of these concepts require being clearly delimited.

The main body of the survey is presented in Section IV. In this part the reader can find tables which summarize characteristics and quantifiable results from the different works. The section is divided in three categories which include in broad terms: protocols, algorithms, and implementations (software/hardware). For each category we provide a discussion and identify strategies used in the literature. The qualitative analysis of the papers is expanded in Section V where we provide our views on each of the different papers as a whole, aiming at assessing if they do in fact belong to ECLC.

Qualitative analysis of the surveyed papers can be found in Section VI. This section is rich with resources for studying the surveyed papers using different analysis approaches.

[1] https://www.tamps.cinvestav.mx/~datasets/
[2] http://dx.doi.org/10.21227/bqfj-6c39

This part of the document requires that the reader has followed the previous three sections since the labeling of the data might be confusing otherwise.

Sections VII, VIII, and IX, provide insights on the characteristics of the topic surveyed, in regards to applications, strategies, trends, and open problems. Lastly, Section X concludes the work with a summary of our findings. If the reader is interested in our findings, beyond the detailed survey and analysis of the papers, or in the answers we provide for the research questions enumerated they can go directly to these last sections.

II. ELLIPTIC CURVES AND CRYPTOGRAPHY

The security of PKC systems relies on mathematical principles denominated *one-way functions*. These relations have two main characteristics: first that knowing all the input variables it is *easy* to solve for the result, second that if the result and only some input variables are known it is *difficult* to solve for the missing inputs. Only a handful of these problems are known and used in cryptography, the most popular ones being the *integer factorization problem* and the *discrete logarithm problem*. The latter is of particular interest to us since the security of most ECC systems depends on it.

A. BASIS

All modern PKC constructions rely on group's theory in order to guarantee the security of the systems. In the following we provide some definitions useful to frame the concepts around ECC that are used in the literature review.

Definition 2: A *group* is a set of elements G equipped with a binary operation $f : G \times G \rightarrow G$ such that f is associative with an identity element e and every element in G has an inverse. G is said to be *abelian* if and only if f is commutative.

Let "f" be called the *group operation*. In practical realizations, groups can be additive (if f represents an addition of elements) or multiplicative (if f represents a multiplication). Independently of the group realization, usually the notation is multiplicative. That is, the group operation is denoted by '*', the identity element in G is '1', and the inverse for any a in G is a^{-1}. The consecutive application of the group operation n times over a group element p is usually represented as p^n.

It is important to determine the cardinality of any group used in cryptography since the difficulty of any mathematical problem defined over such groups is associated with the number of elements in the group.

Definition 3: The group is *finite* if and only if G is a finite set, in which case the number of elements in G, its cardinality, is called the *order* of G.

However, by using the totality of the elements in a group can reduce the complexity, at the same time, if we know that all the elements are valid.

A subgroup H of G is a subset of G, which contains e, if and only if $f : H \times H \rightarrow H$ holds for every element in H and the inverse of such elements is also in H: for any $p \in G$, the production $q = p^n$ for any $n \in \mathbb{Z}$ is a subgroup of G generated by p, denoted as $\langle p \rangle$.

By knowing the group *generator*, a wide set of elements can be specified with a single root element, instead of enumerating all the contents in the set.

Definition 4: The generator, root, or primitive element of G is any $r \in G$ such that $\langle r \rangle = G$.

Cyclic groups are a particular instance of groups useful for cryptographic applications since the results of any group operation is also in the group, hence it is not needed to verify the result and the calculations ought not need to be repeated.

Definition 5: If exists an $r \in G$ such that $\langle r \rangle = G$, then G is said to be cyclic. The subgroups found in a cyclic group are also cyclic.

If a generator is known, and it is used to represent a subgroup of G, the cardinality of this subgroup is also an interesting property to know. The number of elements in a subgroup created by a generator is known as the *generator order*.

Definition 6: An element $p \in G$ has finite order if and only if $\langle p \rangle$ is finite. The order of p is the cardinality of $\langle p \rangle$ and is given by the smallest $t \in \mathbb{Z}$ such that $p^t = e$. This is also called the generator order for $\langle p \rangle$.

Groups are defined for a single group operation. The algebraic structure which uses both an additive and multiplicative composition laws is called a *ring*.

Definition 7: A set of elements R with two binary operations $\{f, g\}$ is a ring if and only if R is a commutative group with f, and g is associative and distributive over f. In this case f and g represent additive and multiplicative group operations with 0 and 1 as identity element, respectively.

Definition 8: If a ring R is commutative and all of its nonzero elements are invertible it is said that R is a *field*.

In order for all the elements in a field to have an inverse, it must be defined by either 0 or a prime.

Definition 9: If the order of a field K is finite, then it is said to be a *finite field*.

In Elliptic Curve Cryptography the elliptic curves are always defined over finite fields. The order of the finite field K is given by p^m, where p is a prime and m is the finite dimension of the vector space in K. Finite fields are also known as Galois fields, denoted by GF(p^m).

Definition 10: For any prime p and any positive integer m there exists a finite field with $q = p^m$ elements denoted by $\mathbb{F}_q$.

A finite field of order q ($\mathbb{F}_q$) exists if and only if its order is a prime power $q = p^m$. The most common constructions include the cases where $q = p$ (denoted prime field $\mathbb{F}_p$) and where $q = 2^m$ (denoted binary field $\mathbb{F}_{2^m}$). These are the basis for defining the most popular elliptic curve systems.

For further details on groups and fields theory the reader might consult [10, Ch. 2].

B. ELLIPTIC CURVES

Following the definition in (1), Fig. 3 shows two elliptic curve groups E($\mathbb{F}_q$) where $q = 19$ and $q = 97$.

All these tuples denoted as $E(\mathbb{F}_q)$ are called *points* with x and y referred as coordinates [11]. The set $E(\mathbb{F}_q)$ together

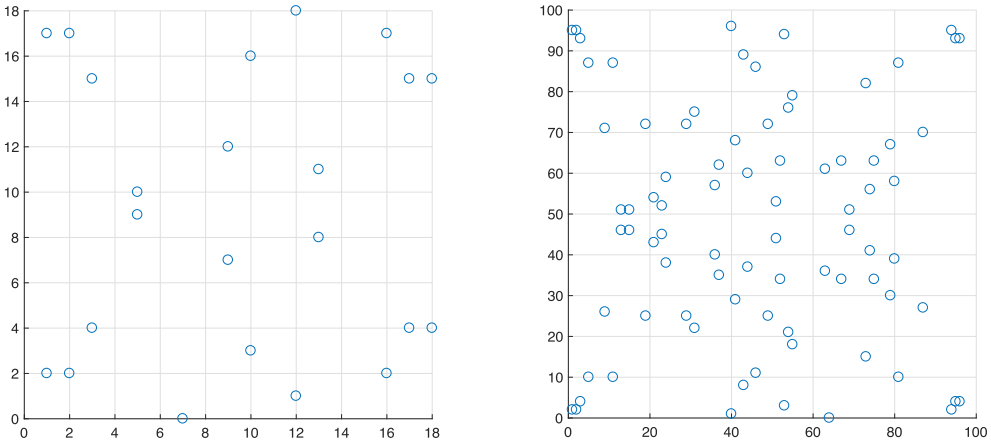

FIGURE 3. Points in the curve $y^2 = x^3 - 7x + 10 \in \mathbb{F}_q$ with $q = 19$ (left) and $q = 97$ (right). Note that, for every x, there are at most two points.

with a point at infinity $\mathcal{O}$ form a group. The group operation is the *addition law*, which can be performed through arithmetic operations in $\mathbb{F}_q$ according to well defined formulae [11]. With this addition rule, the set of points $E(\mathbb{F}_q)$ forms an abelian group with $\mathcal{O}$ serving as the identity element. Cyclic subgroups of such elliptic curve groups can be used to implement cryptographic systems.

1) PROPERTIES

The *order* of an elliptic curve group $\{E(\mathbb{F}_q) \cup \mathcal{O}\}$, denoted by o, is the number of points in E. The values of o and q are related by the *Frobenius trace* $t = q + 1 - o$. *Hasse's* theorem implies that $|t| \leq 2\sqrt{q}$. Given a point $P \in E(\mathbb{F}_q)$ its order is the smallest positive integer n such that $nP = \mathcal{O}$. By *Lagrange's* theorem, the order of a point $P \in E(\mathbb{F}_q)$ divides the order o of the group $E(\mathbb{F}_q)$. Thus $oP = \mathcal{O}$ for any $P \in E(\mathbb{F}_q)$ and, consequently, the order of a point is always less than or equal to the order of the elliptic curve.

2) ARITHMETIC

Different types of computations are involved in ECC. These are of diverse nature, depending if they are defined for the elliptic curve, for the elliptic curve group, or for the finite field.

a: CURVE ARITHMETIC

In the case of curve operations the inputs are points in the elliptic curve, which are used to find other points in the elliptic curve. A single calculation belongs to this set, the scalar multiplication. This is the embodiment of the group law: the application of successive group operations (addition) to a point $P \in E(\mathbb{F}_q)$ will generate as a result other points also in the curve. If the number of additions applied to the point equals the order of the elliptic curve group, the result is the original point.

Scalar multiplication in the curve $\mathrm{E}(\mathbb{F}_q)$ means calculating k additions of a point P and is represented by kP. In this process, a scalar $k \in \mathbb{N}$ and a curve point $P \in E(\mathbb{F}_q)$ are used to find a different point $Q \in E(\mathbb{F}_q)$ such that $Q = kP$ through a defined set of rules. Multiple methods for performing scalar multiplication have been proposed, among them: double and add, NAF, Montgomery ladder, to mention a few [12]. All these methods are built on group operations, which are defined for a specific elliptic curve.

b: GROUP ARITHMETIC

This type of arithmetic comprises the realization of the elliptic curve point addition. It is independent of kP algorithms and varies depending on the curve type, finite field and coordinates type being used. The addition of two points in the elliptic curve

$$P + Q \quad \forall\, P, Q \in E(\mathbb{F}_q) \tag{2}$$

and point doubling

$$P + P \quad \forall\, P \in E(\mathbb{F}_q) \tag{3}$$

are the staples of group operations. A scalar multiplication generally uses both these calculations and thus finding efficient formulae[3] for group operations is critical.

c: FIELD ARITHMETIC

The field operations are those defined for $\mathbb{F}_q$. The point addition and doubling are performed as a sequence of operations over different coordinates from the input points, these coordinates are field elements. The field arithmetic is linked and not dependent with the curve arithmetic or the group operations. Different configurations of field operations perform group calculations. The most common of these procedures include field multiplication, polynomial reduction, field addition and subtraction, field squaring and field inversion, to mention a few.

Fig. 4 summarizes the different levels of operations in ECC.

[3]Efficiency is often measured in quantity, diversity, and quality of the underlying field arithmetic.

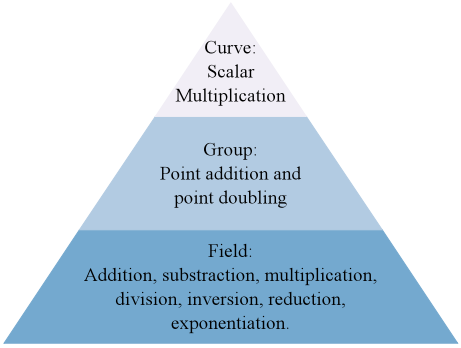

FIGURE 4. Operations in ECC divided by levels.

3) POINT REPRESENTATION

A point $P \in E(\mathbb{F}_q)$ can have different representations which satisfy equivalent models of the Weierstrass equation. Mathematical transformations are applied to each point in order to generate a projection of the curve. These transformation functions must be bijective. The main reason to perform a coordinate transformation over a point is to achieve simplifications for performing curve operations.

The basic system of point representation is *affine*. Under this system each point $P \in E(\mathbb{F}_q)$ is represented by a coordinates pair, generally (x, y). Only two values are required to represent a point and, in some cases, a single coordinate is sufficient. The main drawback is that in the computation of the group operation, multiple field inversions are required to retrieve a result in affine coordinates. Inversions are resource intensive operations in $\mathbb{F}_q$ so they should be avoided in constrained environments.

Projective coordinates are used to solve this problem. A projective point $P \in E(\mathbb{F}_q)$ is denoted by a coordinate's tuple $(X : Y : Z)$. The transformation from affine to projective coordinates generally follows $x = X/Z$ and $y = Y/Z$. In this case the result can be obtained without any inversion operation. An additional value is required to represent a curve point (for the Z coordinate).

Mixed coordinates is a term used to refer to those systems which perform group operations with mixed point representations. In most of the cases, point addition takes an input in affine coordinates but delivers the output in projective coordinates to avoid using field inversions. Point doubling, on the other hand, can use projective inputs to produce projective outputs with low cost. Such systems require a single coordinate transformation at the end of each scalar multiplication thus improving the efficiency of the calculations.

A special point representation denominated w-coordinates was proposed in [13]. In this system a curve point is represented by a single coordinate (w). The equivalence of these coordinates with the affine system is usually $w = x + y$. However, as it is the case with affine coordinates, performing curve operations in w systems involve multiple field inversions. The projective version of these coordinates corresponds with the relation $w = W/Z$ and works in the same way as projective coordinates to reduce the need for field inversions. These are called *projective-w* coordinates. The main reasons for using these special coordinates are to reduce storage requirements and to improve operations efficiency. Depending on the nature of $w = f(x, y)$, the point addition can be tweaked in order to reduce the number of field operations required. Mixed systems of w coordinates can also be constructed. The use of such coordinates reduces the storage space and the number of field operations required, also eliminating the need for field inversions. The main drawback for the w coordinates is that converting the points back to the affine domain involves the *half trace* function [13], which is costly.

4) ELLIPTIC CURVE FAMILIES

The generalized form of an elliptic curve can be reduced or simplified to identify particular sets of curves known as *families*. Fig. 5 presents the taxonomy of the different elliptic curve families. The most relevant ones are described below.

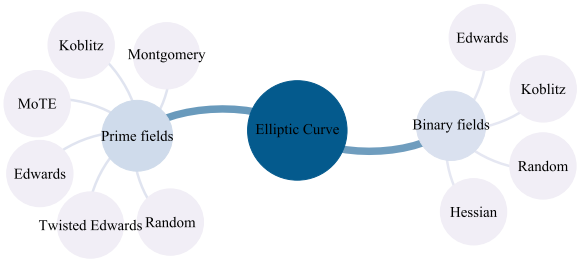

FIGURE 5. Families of elliptic curves defined over prime and binary fields. Each family has a corresponding curve model which represents its curves.

Definition 11: For elliptic curves defined over a field $\mathbb{F}_q$ with $q = p^m$ and $m = 1$ the *Weierstrass* equation is simplified as

$$E_p : y^2 = x^3 + ax + b \tag{4}$$

with $a, b \in \mathbb{F}_p$ and $4a^3 + 27b^2 \neq 0$.

These curves are generally denominated *prime curves* and the model that defines them is referred as *reduced Weierstrass equation*.

Definition 12: Nonsupersingular elliptic curves over a finite field $\mathbb{F}_q$ with $q = p^m$ and $p = 2$ are defined by

$$E_b : y^2 + xy = x^3 + ax^2 + b \tag{5}$$

with $a, b \in \mathbb{F}_{2^m}$.

This is the so called set of *binary elliptic curves*.

Other notable families of curves include *Montgomery curves*, *Koblitz curves*, *Edwards curves*, and the novel *MoTE curves* [14].

Definition 13: A *Montgomery curve* is a form of elliptic curve defined over $\mathbb{F}_q$, with characteristic different of 2, by

$$E_M : By^2 = x^3 + Ax^2 + x \tag{6}$$

with $A \in \mathbb{F}_q \setminus \{-2, 2\}$, $B \in \mathbb{F}_q \setminus \{0\}$ and $B(A^2 - 4) \neq 0$.

Introduced by Peter L. Montgomery in 1987, these curves are used in cryptographic applications. The major attraction of these curves is the possibility to perform point arithmetic with the x coordinate only [15].

Koblitz curves, also known as *anomalous binary curves*, were proposed by Neal Koblitz for cryptographic use in 1992 [16]. Compared to random binary curves, point multiplications methods which are significantly faster are available.

Definition 14: Koblitz curves satisfy an equation of the form

$$E_K : y^2 + xy = x^3 + ax^2 + 1 \tag{7}$$

with $a \in \mathbb{F}_2$.

The family of Edwards Elliptic Curves is defined as follows.

Definition 15: Let $\mathbb{F}_q$ be a field in which $2 \neq 0$ and let $d \in \mathbb{F}_q \setminus \{0, 1\}$; then

$$E_E : x^2 + y^2 = 1 + dx^2y^2 \tag{8}$$

defines an Edwards curve [16].

Computing scalar multiples in these curves takes fewer field operations than in other representations. Additionally, the formulas for addition on Edwards's curves can provide protection against simple side-channel attacks [16].

Definition 16: A variation of these curves called *twisted Edwards curves* satisfies that

$$E_T = a^2 + y^2 = 1 + dx^2y^2 \tag{9}$$

with $a, d \in \mathbb{F}_q \setminus \{0\}$ and $a \neq d$.

A *MoTE curve* can be described as an elliptic curve which has the Montgomery model as well as twisted Edwards model [15].

Definition 17: The Montgomery model of a MoTE curve is given by an equation as

$$E_M : -(A + 2)y^2 = x^3 + Ax^2 + x, \tag{10}$$

which means the parameter $B = -(A + 2)$. The birationally-equivalent twisted Edwards model of the above MoTE curve is given by

$$E_T : -x^2 + y^2 = 1 + \frac{2 - A}{2 + A}x^2y^2, \tag{11}$$

C. THE DISCRETE LOGARITHM PROBLEM

A discrete logarithm is an integer k solving $b^k = g$, where b and g are elements of a finite group. This construction was used to propose *one-way functions* utilized as the basis for *asymmetric cryptography* [17].

In order for a system based on discrete logarithms to be efficient, fast algorithms for computing the group operation must be available. For security, the discrete logarithm problem should be computationally intractable [12].

The most popular groups for implementing discrete logarithm systems are the cyclic subgroups of the multiplicative group of a finite field, and cyclic subgroups of elliptic curve groups.

Definition 18: Discrete Logarithm Problem over Elliptic Curves. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. Let P be a point in $E(\mathbb{F}_q)$, and suppose that P has a prime order n. Then, the cyclic subgroup of $E(\mathbb{F}_q)$ generated by P is

$$\langle P \rangle = \{\mathcal{O}, P, 2 \cdot P, 3 \cdot P, \ldots, (n - 1) \cdot P\}. \tag{12}$$

For these systems the group operation $\cdot$ is the consecutive addition of elliptic curve points or *scalar multiplication*. Given a point

$$Q = k \cdot P \in \langle P \rangle \tag{13}$$

and the root element or generator P, finding k is called the Discrete Logarithm Problem over Elliptic Curves (ECDLP) and is computationally intractable [12].

The DLP is of practical use in asymmetric cryptography since it is the basis for the key pair system. In the case of ECC, the prime q, the equation of the elliptic curve E, the point P, and its order n are public domain parameters. A *private key* is an integer k that is selected uniformly at random from the interval $[1, n - 1]$ and the corresponding *public key* is

$$Q = k \cdot P. \tag{14}$$

D. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptography (ECC) was discovered in 1985 by Neal Koblitz and Victor Miller. ECC schemes are public-key mechanisms used in cryptographic protocols in order to provide specific security services. The security the systems based on elliptic curves relies on the hardness of the ECDLP [12]. Currently the best algorithms known to solve this problem have fully exponential running time, in contrast to the subexponential-time algorithms known for the integer factorization problem. This difference is what creates the main advantage for ECC over other public key alternatives: smaller key sizes are sufficient to achieve an equivalent security level.

1) DOMAIN PARAMETERS

The domain parameters for an elliptic curve scheme are the necessary elements to describe an elliptic curve group. These include the elliptic curve E, a finite field $\mathbb{F}_q$, a base point $P \in E(\mathbb{F}_q)$, and its order n [12]. The parameters should be chosen according to application constraints. Typically, domain parameters are shared by a group of entities; however, in some applications they may be specific to each user.

Domain parameters $D = (\mathbb{F}_q, E, P, n)$ are comprised of:

1) The finite field $\mathbb{F}_q$.
2) The *coefficients* that define the equation of the elliptic curve E over $\mathbb{F}_q$.
3) Two field elements x_P and y_P in $\mathbb{F}_q$ that define a base point $P = (x_P, y_P) \in E(\mathbb{F}_q)$ in affine coordinates. P has prime order and is called the *base point*, *primitive root*, or $\langle P \rangle = E(\mathbb{F}_q)$.
4) The *order* n of P.

2) KEY ESTABLISHMENT

The purpose of a key establishment protocol is to provide two or more entities communicating over an open network with a shared secret key. The key may then be used in a symmetric-key protocol to achieve some cryptographic goal such as confidentiality or data integrity [12].

Elliptic Curve Diffie-Hellman (ECDH): Suppose two parties A and B want to establish a shared key. Let A and B agree on the common domain parameters D. A randomly chooses $a \in [1, n-1]$ and computes $P_A = a \cdot P$ while B follows the same procedure and obtains $P_B = b \cdot P$. A and B publicly exchange these intermediate results. If the ECDLP is hard in $E(\mathbb{F}_q)$, a or b cannot be computed given $\{P_A, P\}$ or $\{P_B, P\}$, respectively. Upon receiving P_B, A computes

$$P_K = a \cdot P_B = (a \times b) \cdot P. \tag{15}$$

Now B can obtain the same result as

$$P_K = b \cdot P_A = (b \times a) \cdot P, \tag{16}$$

thus they are both in possession of a group element P_K, becoming the shared key, which should not be computable from the public values P_A and P_B. The interaction diagram for the basic ECDH protocol is illustrated in Fig. 6.

The security of ECDH relies on the ECDLP. As a protocol, the problems an attacker must solve are the *Diffie-Hellman computational problem* or the *Diffie-Hellman decisional problem*.

- Computational Diffie-Hellman problem: Computing abP given aP and bP.
- Decisional Diffie-Hellman problem: Given aP, bP and cP to decide whether $cP = abP$.

3) DATA ENCRYPTION

In public-key encryption systems each entity A has a *public key* P_A and a corresponding *private key* a. In secure systems, the task of computing a given P_A is computationally intractable. The public key defines an *encryption transformation* E_{P_A}, whereas the private key defines the associated *decryption transformation* D_a. Any entity B wishing to send a message m to A obtains an authentic copy of A's public key P_A, uses the encryption transformation to obtain the ciphertext

$$c = E_{P_A}(m), \tag{17}$$

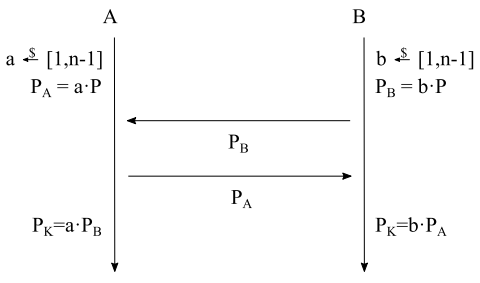

FIGURE 6. Interaction diagram for the basic ECDH protocol. In this scheme the parameters $\mathbb{F}_q$, $E(\mathbb{F}_q)$, and P are publicly known.

and sends c to A. To decrypt c, A applies the decryption transformation to obtain the original message

$$m = D_a(c), \tag{18}$$

as described in [18].

EC-ElGamal: In order to establish a secure communications channel, B creates a key pair $\{b, P_B\}$ as

$$P_B = b \cdot P \tag{19}$$

where $b \in [1, n-1]$ and publishes P_B while keeping b secret. To transmit a message m to B it must first be mapped to a point in the curve as

$$P_m \leftarrow m. \tag{20}$$

User A chooses a random integer a and sends the pair of points $\{P_A, P_E\}$ where

$$P_A = a \cdot P \tag{21}$$

and

$$P_E = P_m + a \cdot P_B. \tag{22}$$

To read the message, B multiplies the first point in the pair by his secret as

$$b \cdot P_A = (b \times a) \cdot P, \tag{23}$$

and then subtracts the results from the second point in the pair as

$$\begin{aligned} P_E - b \cdot P_A &= P_m + a \cdot P_B - b \cdot P_A \\ &= P_m + (a \times b) \cdot P - (b \times a) \cdot P = P_m \end{aligned} \tag{24}$$

as shown in [19]. Fig. 7 provides an interaction diagram for the EC-ElGamal cryptosystem.

The Elliptic Curve Integrated Encryption Scheme (ECIES): ECIES was proposed by Bellare and Rogaway, and is a variant of the ElGamal public-key encryption scheme [12]. In ECIES, a secret P_K obtained using ECDH is used to derive two symmetric keys k_1 and k_2 with a Key Derivation Function (KDF). The key k_1 is used to encrypt the message using a symmetric-key cipher, with encryption (E) and decryption (D) functions, whereas the key k_2 is used to authenticate the resulting ciphertext using a Message Authentication Code (MAC) function. The interaction diagram for ECIES is shown in Fig. 8.

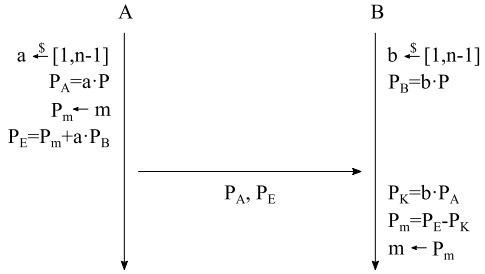

FIGURE 7. Interaction diagram for the EC-ElGamal cryptosystem. In this scheme the parameters $\mathbb{F}_q$, $E(\mathbb{F}_q)$, P and the mapping $m \rightarrow P_m$ are publicly known. The receiver of information must disclose its public key.

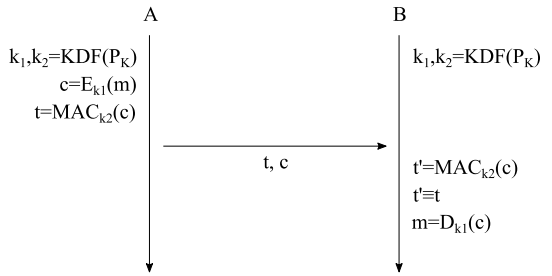

FIGURE 8. Interaction diagram for the ECIES cryptosystem. In this case P_K represents a shared secret generated using the ECDH algorithm, and is used to generate two secret keys with a key derivation function (KDF). The operations E and D are the encryption and decryption functions for a symmetric cipher. *m*, *c*, and *t* represent the plain message, the encrypted message and a MAC tag, respectively.

4) DIGITAL SIGNATURES

A *digital signature* of a message is a code dependent on the signers private key and the message being signed. Signatures must be verifiable; if a dispute arises as to whether a party signed a document, an unbiased third party should be able to resolve the matter using the public key of the signer [18].

ECDSA: With a private key a and a public key P_A, to generate a signature for the message m, A selects a random $k \in [1, n-1]$, computes

$$k \cdot P = (x_1, y_1). \tag{25}$$

Then A computes

$$r = x_1 \bmod n \tag{26}$$

and if the result is zero, a different k must be selected. Next, A calculates

$$e = \mathrm{H}(m) \tag{27}$$

and

$$s = k^{-1}(e + a \times r) \bmod n, \tag{28}$$

where H denotes a cryptographic hash function. If $s = 0$ the process must be restarted. The signature for the message m is the pair (r, s). Fig. 9 provides an interaction diagram for this signature scheme.

To verify the signature generated by A, user B must have an authenticated copy of P_A. First, B verifies that r and s are valid integers, if any verification fails the signature is rejected. After the verification, B computes

$$e = H(m) \quad \text{and } w = s^{-1} \bmod n. \tag{29}$$

Next, B calculates

$$u_1 = e \times w \bmod n \quad \text{and } u_2 = r \times w \bmod n. \tag{30}$$

Then B computes

$$X = u_1 \cdot P + u_2 \cdot P_A = (x_2, y_2). \tag{31}$$

If $X = \mathcal{O}$ the signature is rejected. Finally B computes

$$v = x_2 \bmod n. \tag{32}$$

The signature is accepted if and only if $v = r$.

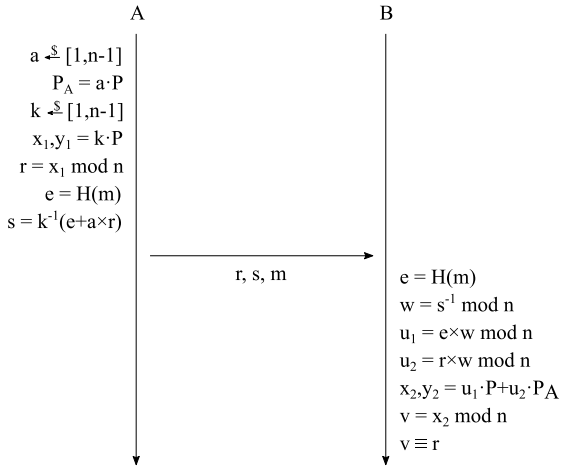

FIGURE 9. Interaction diagram for the ECDSA scheme. The message to be signed is represented by *m* and the signature itself is represented by (r, s).

Fig. 10 presents a classification of the reviewed cryptosystems. Some derived algorithms that were not mentioned in this review are also included in the classification for completeness, these can be looked up in [12].

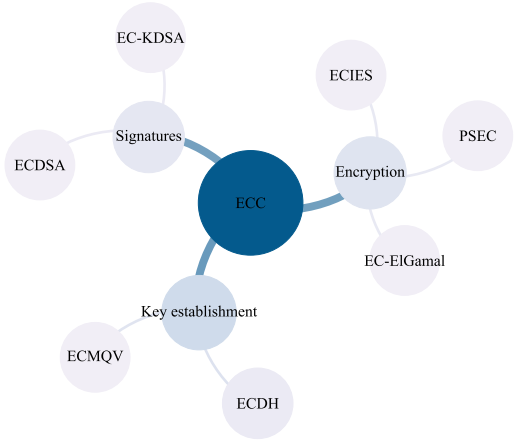

FIGURE 10. Classification of different Elliptic Curve based security schemes.

E. ECC AND OTHER PKC SOLUTIONS

As reviewed in the previous examples, the most important security services provided by ECC include key establishment, confidentiality, integrity, and authentication. However, ECC is not the only cryptographic solution for these tasks.

Before ECC gained widespread attention, systems which relied on integer factoring and discrete logarithms over multiplicative groups were used as PKC instances. Perhaps the most popular from these is RSA, proposed in 1977 for creating encryption and digital signature methods. Nowadays this scheme is still supported and used by many internet servers. Due to breakthroughs on algorithms for solving its underlying hard problem, the length of RSA's operands has increased to 3072-bits for a security level of 128-bits. In comparison, ECC only requires operands of 256-bits for the same security

level. This difference of an order of magnitude is a significant advantage for ECC as shorter operands imply shorter computing times and smaller storage requirements.

Garbled circuits [20] are a classical instance of a cryptographic approach for achieving secure two-party computation. These constructions were first described in 1986 and have been called "the first secure computation protocol." Their main application is enabling two mistrusting parties to achieve a secure function evaluation (SFE) while providing privacy, authentication, and obliviousness [21]. These protocols rely on a primitive called oblivious transfer which is difficult to achieve; classical proposals relied on RSA for this regard. The use of these constructions in constrained environments is limited by performance and storage issues that have not been properly addressed. Moreover, the proposals in the literature have limited the study of the security of garbled circuits to reduced adversarial models (*semi-honest*) and often rely on symmetric primitives which represent additional costs.

A notion which was considered almost from the beginning (one year after RSA), but believed to be inapplicable for many years is homomorphic encryption (HE) [22]. Under this scheme it is possible to perform computations on a ciphertext, without requiring decrypting the data. This implies that an entity applying some processing over confidential information does not require disclosure of such secrets. Although partial homomorphism was possible using many classical cryptosystems, among them RSA, fully homomorphic encryption (FHE) was created with the development of lattice based cryptography. Both learning with errors (LWE) and NTRU have been used to create homomorphic encryption schemes [23], [24]. The main application of FHE is to allow an untrusted party to carry computations over a protected secret without granting it access to the data, this represents an extension of the notion of privacy.

Beyond the scope of HE, lattice-related problems (SIS, LWE, RLWE, MLWE, LWR, MLWR) have been employed for creating PKC solutions [25]. The main advantage of lattice-based cryptosystems include strong provable security guarantees, flexibility, and high asymptotic efficiency [26]. Although lattice-based systems can perform encryption and decryption operations with relative efficiency, their key sizes are much larger than those of RSA or ECC, which makes their use impractical for constrained devices.

When the underlying one-way function used in a cryptographic construction uses an error correcting code, such a solution is denominated a code-based cryptosystem [27]. The McEliece public key encryption scheme, proposed in 1978, is acknowledged to be the first of these systems. The main feature of code-based systems is their security—the McEliece cryptosystem remains secure nowadays with just minor tweaks. Their main downside is the size of their key pairs, larger than even those used in lattice instances. Both lattice and code solutions are believed to resists quantum attack models and so, although not as efficient as ECC, are of interest for the research community. The study of the impact of quantum computers on PKC, as well as the discussion on the advances in the development of quantum computers are outside the scope of this work.

Systems based on elliptic curve groups achieve greater efficiency and flexibility than the aforementioned alternatives. They have been adopted in a wide range of applications, and in some cases under critical constraints. ECC is the most popular of such systems, but in the literature it is possible to find different alternatives which extend its security features.

Bilinear pairings over elliptic curve groups have been used to construct cryptographic schemes which make use the user's identity [28]. Identity based encryption (IBE) [29] is perhaps the most notorious of such proposals. Under this model, the public key of a user is a random string which can provide some meaningful information associated with the identity of the user: name, address, email. The secret keys are distributed to each user by a trusted third party in a secure way. The main advantage of these constructions is the ease in the management and distribution of public keys. Although this field has great potential in consumer appliances, as described in [28] "[i]t makes the cryptographic aspects of the communication almost transparent to the user", for constrained devices the increased complexity obfuscates its advantages.

Attribute based encryption (ABE) [30] is a modification of the identity-based systems on that the "identity" is "a set of descriptive attributes." The security of ABE also relies on bilinear pairings defined over elliptic curve groups. This scheme can be used in structured multi-party environments where a single policy can provide distinct access levels based on each party's attributes. Efficiency concerns also restrict the use of ABE solutions in constrained environments.

Although the research on pairing based cryptography is still novel, in the literature we can find some instances of such systems in the context of lightweight cryptography [31]–[34]. Such works are not included in this survey since the study of lightweight pairings is not our main focus. Nonetheless, since elliptic curves are an essential part in pairing based cryptography (IBE, ABE,. . .), ECLC would also impact the realization of such systems and possibly will enable their practical use in IoT domains.

F. SUMMARY

Up to this point we have reviewed concepts which are important in understanding the survey. Fig. 11 provides a diagram which summarizes the relations between some of the different topics reviewed.

III. METHODS

This section describes the methodology for the literature review, the classification categories, and the surveyed data. The labels introduced in this section are used throughout the rest of the document for identifying the surveyed data.

A. SYSTEMATIC REVIEW

This review was conducted over papers retrieved from four electronic collections: IEEE Xplore, Springer Link, ACM

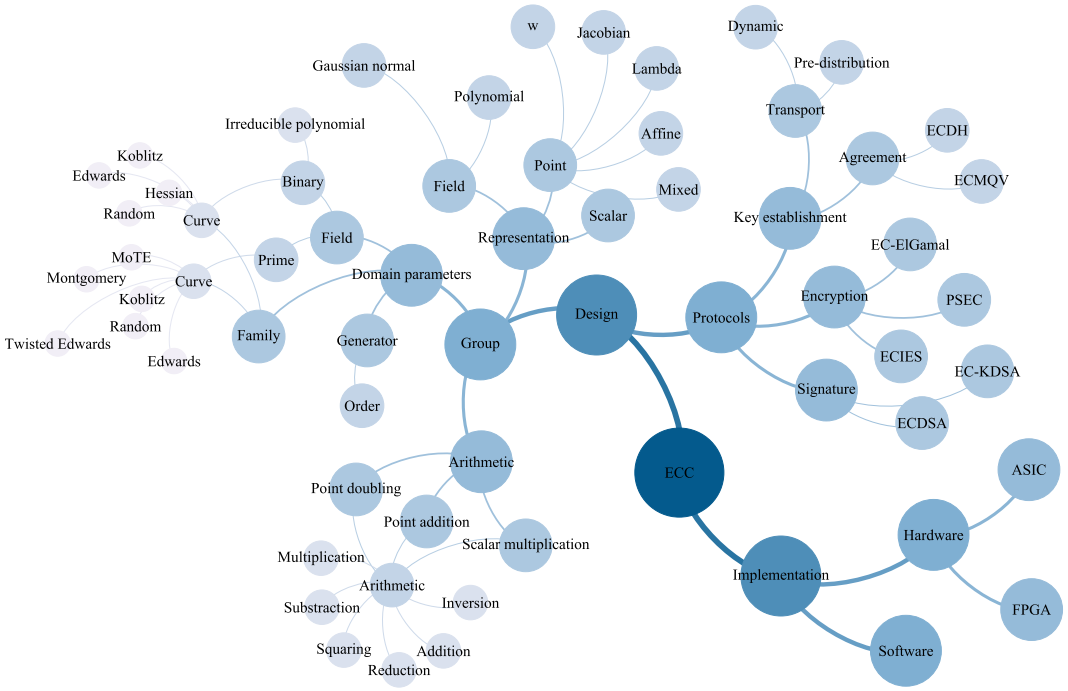

FIGURE 11. Summary of different concepts reviewed and their relations.

Digital Library, and Science Direct. The terms ''lightweight ECC'', ''lightweight elliptic curves'', ''low-area ECC'', and ''low-power ECC'' were used to search for the related works.

Given the scope of this work, the analysis was focused on those papers where the authors use the adjective ''lightweight'' to describe their proposal. The additional terms included in the querying process allowed us to gather a wider set of documents, which were then screened. The review was updated up to September 2018.

In total 91 papers were obtained from the digital libraries consulted. Using the criteria of being called ''lightweight'' and using Elliptic Curve Cryptography this set was reduced to 62 papers. All of these are considered in our qualitative and quantitative analysis.

B. SURVEY CATEGORIES

The qualitative assessment of the surveyed papers is divided in two parts. First we provide summaries of the surveyed data in which the papers are grouped according to the categories described in the following. In the second part we offer our qualitative assessment of all the works as a set. The latter represents an analysis on ECLC as a whole.

We use three categories for structuring the presentation of the surveyed papers:

CAT1 Papers which propose elliptic-curve based lightweight protocols.

CAT2 Papers which propose elliptic-curve based lightweight algorithms for solving group operations.

CAT3 Papers which propose elliptic-curve based lightweight realizations, divided in software, FPGA, and ASIC.

These categories allow us to focus the analysis and discussion on reduced contexts. In this way we can provide general strategies, metrics, and results for each case.

C. SURVEYED DATA

In the following we review the information that we sought to extract from every paper and introduce labels for identifying each field.

A) WHY IS IT CONSIDERED LIGHTWEIGHT?

In our review we found out that a variety of motivations followed by authors to describe their works as ''lightweight'' or use this keyword. However, an specific reasoning is rarely made explicit. Six main motivations are identified:

[A1] New protocol: A protocol is presented to establish a shared secret, to authenticate a node, or to sign a message. In this case the objective is to reduce the number of operations and thus to reduce the quantity of messages being sent.

[A2] Optimized implementation: With aims to implement at the lowest possible cost the arithmetic in ECC that include: scalar multiplication, point addition, point doubling, field operations. This category also covers software or hardware implementations which make use of optimization techniques such as highly optimized assembly and low level digital design.

[A3] Modification to ECC: Contributions in this category present an alternative to generic ECC constructions aiming at reducing resource consumption or improving its implementations performance in constrained devices. Examples of

this include modifications to the form of the underlying finite field, the elliptic curve family, the group algorithms, the point representation, or the scalar representation.

[A4] ECC-based: Some works define their proposals as lightweight just by using ECC instead of classical alternatives like RSA. Whereas there is some truth to such affirmation, what we propose as lightweight involves more design and implementation considerations which tailor the solution to the application.

[A5] Tailored system: By replacing instances of asymmetric primitives or RSA and its relatives on legacy systems with ECC, some proposals claim to achieve lightweightness. This can be considered similar to A4, however in this case more design aspects are involved.

[A6] More lightweight than the SoTA: Some validation is provided when a solution claims to be lightweight by requiring less resources than some other work in the literature. This does not say anything about the work used as reference, however. If an ECC based system is implemented, and then said to be lightweight from the results obtained but the design aspect is missing, we assign it this classification.

B) ABSTRACTION LEVELS OF THE WORK

Design *abstraction levels* can be observed in the development of solutions for constrained environments. This idea was originally proposed Fan *et al.* [35] illustrate a security pyramid which delimits four abstraction levels to achieve a low-power implementation of ECC: System, Protocol and Algorithm, Architecture, and Circuit. This concept can be extended to lightweight solutions which consider not only power reduction, but area, performance, and security as well. In this survey we propose five abstraction levels which can be identified in the design of a lightweight security solution. We have opted to separate the protocol and algorithm levels since they differ substantially on the underlying problems and the evaluation metrics used.

[B1] System: A contribution which considers technology-specific optimizations to improve the quality of their proposal (for example, optimizations which exploit the characteristics of WSN) is considered to observe the *System* abstraction level. By nature, all protocols are considered to be ''system aware'', as they are generally designed with the application requirements in mind.

[B2] Protocol: A lightweight solution which seeks to provide security services in the scope of public key cryptography (i.e., authentication, signature, key establishment) for constrained environments (the *System*) ought to consider the protocol design in aims to reduce the computational time and the amount of information being transmitted over the wireless channel. Works which propose optimizations concerning the protocols being utilized are considered to meet the *Protocol* abstraction level.

[B3] Algorithm: The security algorithms are closely associated with the underlying elliptic curve group. If a proposal aims to be adapted by constrained devices, it is expected that it should make use of algorithms adapted for the application (the *System* and the implementation technology). The *Algorithm* abstraction level is met if the works fulfill this requirement. Software implementations are also considered in this category, since the optimizations focus on tailoring the algorithms to a given processor architecture.

[B4] Architecture: Architectures which implement the proposed algorithms efficiently in order to meet the application constrains. The design of a digital circuit to perform computations (regardless of the implementation technology) entitles a work with the *Architecture* abstraction level. Different technologies exist which allow for implementing architectures of digital circuits, the most common ones being FPGA and ASIC. In the former a description of the architecture is created and then mapped to an array of logic elements present in the FPGA. Most of this process is done automatically by vendor specific design tools and so implementations targeting FPGA are said to meet only the *Architecture* abstraction level.

[B5] Circuit: ASIC design provides greater control over the final product and enables optimizations at circuit level which can help improve the efficiency of a solution. Proposals which use this implementation technology are said to meet the *Circuit* abstraction level. Low-level FPGA design also enables for circuit level optimizations, however these processes are uncommon as FPGAs are generally used for rapid development and testing. If a publication deals with low-level FPGA design it can also be said to meet the *Circuit* abstraction level characteristic.

C) MAIN GOALS OF THE WORK

We identified the main contributions of a paper based on its design goals. Some proposals seek throughput improvement, runtime reduction, or latency reduction. Other aim at reducing physical area or memory usage. Some try to improve security or reduce the amount of data transmitted. By identifying a ''main'' goal, we do not say that a work overlooks the other goals, but that they simply take extra steps to improve a specific characteristic. The main aim(s) of a work can be:

[C1] Performance: To reduce the cycle count or the latency of the proposal. This usually encompasses faster algorithms, reduced number of operations, and reduced number of steps, among others. This trait is commonly associated with metrics such as throughput, latency, frequency, and runtime. The implementation platform determines important characteristics such as the frequency, which affects the throughput and the runtime. A more technology-independent measurement is the latency, which is the required number of cycles to perform a task, usually clock cycles or processor cycles. The latency, however, depends on the implementation strategy for the algorithm.

[C2] Hardware resources: To reduce memory footprint of a software implementation or elements count in the case of hardware. The use of smaller fields generally falls in this category, along with reducing the variety of the operations and

optimizing the circuit design for hardware implementations. For software oriented solutions the resource usage usually translates into memory usage, which can be program memory (code lines) or application memory (bits). In regards to hardware implementations, most commonly this is associated with the target platform. In the case of FPGAs the resource usage can be given as a count of FPGA elements utilized (slices, Look-up Tables, Flip-Flops, block RAMs, multipliers (DSPs)). For ASIC the resource usage is equivalent to the implementation size, which is commonly given in gate equivalents (GE).

[C3] Security: To select a high security setting and then part from it to build the lightweight solution. Mitigating a wide range of attacks and improving the security features of the system also falls in this category. Some of the reviewed works measure the security of their proposals based on the number of attacks resisted. Other approaches may consider relating this with the key size utilized.

[C4] Bandwidth: To reduce the quantities of data transmitted, which frees up the communications channel and saves energy. In the design of protocols it is usually reported the number of bits that are required to be transmitted. This can also be related to the number of messages required to be exchanged. Both influence in the bandwidth of the system and the energy consumption of the platform.

[C5] Energy: Systems which have been designed in order to reduce power dissipation or energy consumption. The power dissipation is a key feature for passive elements such as RFID tags, it can be divided in quiescent and dynamic power. It has a direct relation with the physical size of the device, the operational power and frequency, and the temperature. Energy consumption is derived as a function of power dissipation and execution time, it is important for battery-powered devices, where a higher consumption impacts the lifetime of the platform.

D) SYSTEM

Lightweight solutions are required for different constrained environments. Depending on the specific application, the interests or goals of a paper might be shifted. For each work we have assigned a ''main'' system that they target. The IoT tends to the used as a cover-all alternative, but given its heterogeneity and novelty little is known about its requirements. On contrast, RFID and WSN have clearly defined scopes and a handful implementation devices. Designing for WSN or RFID, therefore, enables some works to take more application-related trade-offs. Emerging technologies are also represented with *e*-Health and Smart Grid, these are well defined but more recent. Embedded and mobile systems are as well mentioned as target systems by some works. These have been around for some time and include devices with less energy/processing/cost restrictions than IoT and similar.

[D1] IoT: The Internet of Things, an interconnected network of heterogeneous applications which are expected to change the world in the next 10 years.

[D2] RFID: Extremely constrained systems which operate passively. Their cost is low and so have widespread use in commercial application such as inventory monitoring.

[D3] WSN: Represent the link between the physical and the cybernetic world. Information systems rely on information, and this information can only be retrieved effectively by sensors. The deployment costs and security concerns have restricted their practical use to research and military applications. Since these devices are actively consuming energy and their retrieval after deployment is impractical, open problems regarding extending battery duration, energy harvesting, and preventing contamination of the environment must be resolved.

[D4] *e*-Health: With a population that grows old many countries must seek ways to automate healthcare systems. We have grouped technologies such as WBANs and remote monitoring of patients into this category. On this group the applications are deeply connected to the user, and so ensuring confidentiality, privacy, and comfort are key aspects for any such solution.

[D5] Smart Grid: Novel technology which proposes to improve the electrical grid in order to improve its efficiency and QoS.

[D6] Embedded systems: Denomination used to refer to ''everything on a board'' systems. The definition says little about its requirements. Embedded systems have been built for decades and are a core technology for industrial automation processes.

[D7] Mobile systems: The physical embodiment of ubiquity. Human society took the phone and turned it into a network which enables sharing multimedia information in real time, all the time, everywhere that there is coverage. State of the art mobiles have higher specifications than many desktop computers. However, the security requirements of the former are enhanced given their wireless nature. A mobile terminal must be protected against remote and direct access, in the similar way of a WSN.

E) IMPLEMENTATION TECHNOLOGY

Among the surveyed works a differentiation can be made according to the implementation platform used. Whereas some works have been developed with aims of achieving high efficiency as software solutions for constrained processors (usually with word sizes of 8-bit), others try to achieve high efficiency through hardware circuitry reduction (as dedicated coprocessors or stand-alone computing units). By choosing the implementation target in an early stage of development, some considerations can be made to further optimize the system at the cost of a loss in its generality. By maintaining generality, the cost is a loss in the improvement potential. The set of papers can be divided in two broad groups: software and hardware. The latter can then be divided between FPGA and ASIC.

[E1] Processor: A defined general-purpose architecture is used to run a sequence of instructions or code. This code is created in order to implement a specific processing task with

the help of memory spaces. The number of instructions in the sequence and the amount of memory required are called temporal and spatial complexity, respectively. Reducing these complexities fits within the algorithm abstraction level.

[E2] FPGA: A reconfigurable array of gates which can be activated to create a specific datapath. This reconfiguration map is generally created by automated design tools based on an RTL design, synthesized from an architectural description. The architectural description used to be created with a Hardware Description Language (HDL), however, nowadays High Level Synthesis tools are available which allow using common languages like C in order to describe the desired behavior. As the designer goes deeper in the design tree, the optimization possibilities widen. In all the reviewed FPGA-related works the proposals seem to use a HDL to describe the architecture.

[E3] ASIC: The same RTL design used for FPGAs can be compiled for ASIC libraries. An Application-Specific Integrated Circuit (ASIC) is a chip manufactured with the minimum number of transistors required to implement an architecture. The area or physical size is reported in area units (mm^2, μm^2, nm^2), this can be used to obtain an estimation of the number of gates required for the circuit: gate equivalents (GE). This metric is the most common ASIC measurement given. It is commonly accepted that a GE equals a 2-NAND in size.

F) SECURITY LEVEL

We have retrieved details of the fields, groups and curves utilized when available. However, the complete information is generally not available and thus cannot be used to classify the works with a high level of detail. It is understandable that the information is often not available since generality and optimization go hand in hand. A work with high generality can function with a wider range of fields and curves. This generality can be exchanged to achieve higher efficiency. Most of the works do describe the field length or the security level, however. We have divided the different security levels according to NIST recommendations for key usage:

[F1] Less than 112-bit: this is no longer recommended.

[F2] Equal to 112-bit: this is recommended until 2030.

[F3] Higher than 112-bit: which can provide security beyond 2030.

IV. SURVEYED WORKS

This section presents data gathered from the set of publications retrieved, which have been denominated lightweight in the publication or related media. We have divided the papers according to the results provided: protocol designs, algorithm specifications, software design, FPGA implementation, ASIC implementation.

A. CAT1: LIGHTWEIGHT PROTOCOLS

Protocols denominated lightweight aim to provide security services for constrained applications. These security services involve most commonly authentication, signature, and key establishment, but data encryption is also considered. The works which involve protocol design usually do not carry out implementation of their solutions, and so the efficiency of these schemes is evaluated using estimated metrics for the number of operations required and the amount of bits transmitted.

Table 2 presents a summary of the most important characteristics of the reviewed protocols. The security services and application scope for each protocol are provided. The different cryptographic primitives required and the participants considered are also reported.

CAT1 STRATEGIES AND REMARKS

All the protocols reviewed specify an application scope. Most predominantly, IoT and its underlying technologies are chosen as proposed applications. In order of predominance, the proposed protocols perform tasks of authentication, key establishment, encryption, and digital signatures.

Although most of the works specify a preference for a prime or a binary finite field, the field length is not always available. A specific elliptic curve is not clearly proposed in most of the cases and neither is the group generator or its order. Similarly, the group operations or the point representation required to perform the scalar multiplication are rarely described.

From the reviewed works which have been classified under the protocol abstraction level, all of them rely on the scalar multiplication, with high predominance of pseudo-random number generators (PRNG) and symmetric primitives such hash functions and block ciphers. Some of them require additional field operations which have been included as modular arithmetic in Table 2.

As for evaluation metrics, the number of operations, the runtime, and the bits transmitted are the most popular alternatives. If the protocol was implemented then it is possible to obtain a runtime, if not, the operations cost is provided. The amount of information transmitted is also important since it determines the bandwidth requirements of a proposal.

Designing a security protocol is a challenging task. The base knowledge required ranges from cryptographic basis, passing by networking theory, to attack models.

In the case of the first, it is necessary to define a strong cryptographic basis for the security of the protocol to rely on. Elliptic curves are interesting in this regard as they require smaller operand sizes than other asymmetric techniques and so contribute to reduce the transmission costs.

Assumptions about the network model and topology determine the number of participants in the protocol. Even though the basic scenario involves a two party system, it is also possible to consider third party service providers (e.g. Certifying Agencies), back-end servers, databases, or group schemes. Having less participants simplifies the network specifications required but increases the complexity in the design and so the overhead of the protocol in the underlying systems.

The application scope can also help to determine the attacks which are more important to offer protections

TABLE 2. Summary of the main characteristics for the surveyed protocols denominated lightweight.

Year	Work	Ref.	Basis	Services	Application	Primitives required	Participants	Evaluation
2007	Kim *et al.*	[36]	ECC over $\mathbb{F}_{2^m}$ and $m = 193$	– Sign.	RFID	kP, PRNG.	– RFID tag – RFID reader – Back-end server with a DB	Operations
2010	Kim *et al.*	[37]	ECC over $\mathbb{F}_p$	– Enc.	RFID	kP, PRNG.	– Mobile terminal – Server	Runtime
2012	Ju *et al.*	[38]	ECC over $\mathbb{F}_p$	– Key est.	WSN	kP, PRNG, Hash function, Symmetric cipher.	– Two sensor nodes	Runtime and operations
2013	Bakhache *et al.*	[39]	ECC over $\mathbb{F}_q$	– Key est. – Auth.	Low-power networks	kP, PRNG, Hash function.	– Two users	Operations
2014	Druml *et al.*	[40]	ECC over $\mathbb{F}_p$	– Auth.	RFID	kP, PRNG.	– Smart card – Mobile reader device – Back-end server	Runtime
2014	Yao *et al.*	[41]	ECC over $\mathbb{F}_p$	– Enc. – Sign.	Mobile systems	kP, PRNG, Hash function, MAC function, Symmetric cipher.	– Two mobile devices	Operations
2014	He *et al.*	[42]	ECC over $\mathbb{F}_p$	– Auth	RFID	kP, PRNG.	– RFID tag – Back-end server with a DB	Bits
2016	Chaudhry *et al.*	[43]	ECC over $\mathbb{F}_p$ and $\lvert p\rvert = 160$	– Key est. – Auth.	IoT	kP, PRNG, 4 Hash functions.	– Server – Users	Operations
2016	Reddy *et al.*	[44]	ECC over $\mathbb{F}_q$	– Auth. – Key est.	Mobile systems	kP, PRNG, Hash function, MAC function .	– Smart card – Foreign agent – Home agent	Operations
2016	He *et al.*	[45]	ECC over $\mathbb{F}_p$	– Key est. – Auth	Smart grid	kP, PRNG, 3 Hash function.	– Smart meter – Server provider – Trusted anchor	Runtime
2016	Lavanya and Natarajan	[46]	ECC over $\mathbb{F}_p$	– Key est. – Auth.	IoT	kP, PRNG, Hash function, Symmetric cipher.	– Initiator – Responder – Security association	-
2016	Kaur *et al.*	[47]	ECC over $\mathbb{F}_{2^m}$ and $m = 163$	– Auth	RFID	kP, PRNG.	– RFID tag – Server	Bits and runtime
2017	Zhang *et al.*	[48]	ECC over $\mathbb{F}_q$	– Auth. – Key est.	IoT WSN	kP, PRNG, 4 Hash functions, MAC function.	– Client – Server	Operations
2017	Win *et al.*	[49]	ECC over $\mathbb{F}_p$ and $\lvert p\rvert = 160$	– Auth.	IoT	kP, PRNG, 4 Hash functions, Symmetric cipher.	– Trusted 3rd party – Users	Operations and runtime
2017	Mathur *et al.*	[50]	ECC over $\mathbb{F}_p$	– Auth. – Enc. – Key est.	IoT	kP, PRNG, Hash function, Symmetric cipher.	– Sensors – Router – Base station	Runtime, cycles, and energy
2017	Mahmood *et al.*	[51]	ECC over $\mathbb{F}_p$ and $\lvert p\rvert = 160$	– Auth.	Smart grid	kP, PRNG, 3 Hash functions	– User – Trusted 3rd party	Operations and bits
2017	Diro *et al.*	[52], [53]	ECC over $\mathbb{F}_p$ and $\lvert p\rvert = 160$	– Auth.	IoT	kP, PRNG.	– Nodes – Broker	Operations and bits
2017	Badra and Zeadally	[54]	ECC over $\mathbb{F}_p$ and $\lvert p\rvert = 160$	– Auth. – Key est. – Enc.	Smart grid	kP, PRNG, Symmetric homomorphic encryption, MAC function.	– Energy supplier – Smart meter	Operations and bits
2017	Lavanya and Natarajan	[55]	ECC over $\mathbb{F}_p$	– Key est.	IoT	kP, PRNG, Hash function, Symmetric cipher, Signature.	– Initiator – Responder	Network metrics
2017	Meddah *et al.*	[56]	ECC over $\mathbb{F}_p$	– Auth. – Enc.	PHR over the Cloud	kP, PRNG, Hash function, MAC function, Symmetric cipher.	– Certificate authority – Cloud Service Provider – Users	Bits
2017	Mohammedi *et al.*	[57]	ECC over $\mathbb{F}_q$	– Auth. – Key est.	Mobile healthcare	kP, PRNG, Hash function.	– Remote server – Mobile device	Bits and runtime.
2017	Hasan *et al.*	[58]	ECC over $\mathbb{F}_p$ and $\lvert p\rvert = 224$	– Auth. – Enc.	IoT	kP, PRNG, Hash function, Symmetric encryption.	– Integrated agent – Smart object	Runtime and operations
2017	Sojka and Langendoerfer	[59]	ECC over $\mathbb{F}_p$	– Enc. – Sign.	WSN	kP, PRNG, Symmetric cipher.	– Signer – Verifier	Bits and cycles
2018	Shen *et al.*	[60]	ECC over $\mathbb{F}_p$	– Auth. – Key est.	WBAN	kP, PRNG, 2 Hash functions, MAC function, Symmetric encryption.	– Personal digital assistance – Sensor nodes – Application provider	Operations
2018	Tewari *et al.*	[61]	ECC over $\mathbb{F}_{2^m}$ and $m = 193$	– Auth.	RFID	kP, PRNG, Hash function.	– Server – Tag	Bits
2018	Diro *et al.*	[62]	ECC over $\mathbb{F}_p$ and $\lvert p\rvert \in \{160, 256, 512\}$	– Enc.	Fog of Things	kP, PRNG.	– Fog node – IoT device	Runtime and bits
2018	Vaniprabha and Poongodi	[63]	ECC over $\mathbb{F}_q$	– Auth.	WBAN	kP, PRNG, 2 Hash functions.	– Verifier – Bio sensor – Data controller – Data feeder	Runtime and network metrics
2018	Shen *et al.*	[64]	ECC over $\mathbb{F}_q$	– Auth.	WBAN	kP, PRNG, 3 Hash functions, MAC function.	– Network manager – WBAN user – Cloud server	Runtime and operations
2018	Mood and Nikooghadam	[65]	ECC over $\mathbb{F}_p$ p521, Curve25519	– Auth.	Smart grid	kP, PRNG, Hash function.	– Trust anchor – Two users	Runtime, bits, and operations.
2018	Almulhim and Zaman	[66]	ECC over $\mathbb{F}_q$	– Auth.	IoT-based *e*-Health	-	– Healthcare provider – Server – Base station – Head node – Sensor nodes	-
2018	Mohammedi *et al.*	[67]	ECC over $\mathbb{F}_p$ and $\lvert p\rvert \in \{160, 192, 224, 256\}$	– Auth.	Home healthcare systems	kP, PRNG, Hash function.	– Remote server – Patient	Runtime and bits.

against. Creating a protocol which resists more attack models improves the security of the network, but this results on requiring more underlying operations, more diverse cryptographic operations, and often more complex network specification requirements.

From the survey, it can be noted how a common approach is to limit the computations that are performed in the constrained device and to define properly which attacks should the system be protected against. These strategies coupled with the use of efficient classes of elliptic curves can help to

achieve a protocol specification which can be denominated lightweight.

B. CAT2: LIGHTWEIGHT ALGORITHMS

In our survey we have identified cryptographic algorithms which are designed to be more efficient than generic alternatives. Table 3 provides a summary of the different algorithms surveyed. In this table we report the problem which is solved by the algorithm, the cryptographic basis of the proposal, the approach followed by the authors to design the algorithm, and the goal of such design.

CAT2 STRATEGIES AND REMARKS

In most of the surveyed works with emphasis at the algorithmic level the main goal is to improve the running time of the scalar multiplication. By exploring novel families of elliptic curves, special primes to define finite fields, or special point representations, each work aims to reduce the number of field operations required in the group arithmetic. The end goals for such optimizations are usually improving performance by requiring fewer operations or by requiring fewer storage registers.

Even though it results convenient to divide the algorithms found in the literature in two classes for benchmarking purposes, the strategies involved in the creation of a lightweight algorithm are similar.

Using special families of elliptic curves or special point representations allows reducing the number of field operations required to perform group operations (point multiplication, point addition, point doubling). This helps both the algorithms and their implementation.

The use of mathematic resources such as the common Z strategy (Co-Z) [77] is an alternative which can also lead to reduction in the number of field operations or registers required in an algorithm. In this case the point representation is modified during processing, at the cost of point conversions before or after processing.

Factoring common expressions is a technique which helps to reduce the number of computations required at the cost of extra memory to hold the intermediate results. This method is interesting for systems that can afford to use some temporary registers in order to reduce the total number of operations. In the same tone, performing the calculations as constants pre-deployment can free up processor cycles at the cost of the energy required to hold the value. Both strategies are found in contributions of novel formulae with reduced number of field operations.

The field representation can also be exploited in order to improve the efficiency of an ECC-based realization. The Optimal Prime Fields and the pseudo-Mersenne prime fields are a couple of such cases. The influence of the prime selection varies from reducing storage space, achieving completeness in the group operations, obtaining processor-friendly field arithmetic, among others.

C. CAT3: LIGHTWEIGHT ECC REALIZATIONS

1) SOFTWARE IMPLEMENTATIONS

Table 4 shows a summary of the papers reporting algorithms for lightweight cryptography implemented in software. This table reports the cryptographic basis of the different implementations but also provides technical details about the underlying platform, the performance benchmark, and the implementation goal.

2) HARDWARE ARCHITECTURES

All the hardware implementations surveyed propose an architectural design. Generally speaking, a hardware architecture is the realization of an algorithm using digital circuits. In this sense, a group of hardware components is interconnected in a way that through the use of control signals it is possible to perform computations over the input data and achieve a desired result. These can be classified as processors, co-processors or standalone architectures. The main difference between these approaches lies on the generality of the solution. Whilst a processor should be able to perform a wide range of related operations, compromising the generality allows for implementation optimizations to achieve more efficient solutions.

Table 5 summarizes the main characteristics and benchmark for the different FPGA implementations found in the

TABLE 3. Summary of the different abstract algorithms denominated lightweight which were surveyed.

Year	Work	Ref.	To solve	Curve	Field	Approach
2010	Sojka *et al.*	[59], [68]	Enc. and sign.	SEC	$\mathbb{F}_p$ where $32 \leq \lvert p \rvert \leq 64$	Applying several changes in the standard ECC parameter sets and algorithms the authors propose a public/secret key hybrid with a reasonable security level and shorter key sizes.
2014	Azarderakhsh *et al.*	[69]	kP	Koblitz	$\mathbb{F}_{2^m}$ where $m = 163$	The work introduces a new technique for point addition in affine coordinate which requires fewer registers.
2014	Liu *et al.*	[15], [70]	kP	GLV, Montgomery	$\mathbb{F}_p$ where $p = u \cdot 2^k + v$	The authors study a special call of finite fields denominated Optimal Prime Fields (OPF) which reduce storage requirements and simplifies the modular reduction operation. Using these OPFs the authors propose a special variant of Montgomery multiplication.
2014	Liu *et al.*	[14], [71]	kP	MoTE	$\mathbb{F}_p$ where $\lvert p \rvert \in \{160, 192, 224, 256\}$	A special form of elliptic curves are presented to exploit the advantages of OPF. These are used to develop optimized field algorithms and achieve efficient implementations.
2015	Roy *et al.*	[72]	kP	Koblitz	$\mathbb{F}_{2^m}$ where $m = 283$	The work introduces a lightweight variant of a scalar conversion algorithm that enables enhanced performance in the implementation of ECC with binary Koblitz curves.
2015	Koziel *et al.*	[73]	kP	Binary Edwards	$\mathbb{F}_{2^m}$ where $m \in \{163, 233, 283\}$	The authors propose formulae for point addition and doubling in Binary Edwards curves which reduce the number of field operations required for a scalar multiplication and the number of registers.
2017	Khleborodov	[74]	kP	Prime	$\mathbb{F}_p$	Proposes scalar multiplication algorithms based on binary and binary-NAF scalar representation methods using mixed coordinate representations.
2018	Järvinen *et al.*	[75]	Sign.	Koblitz	$\mathbb{F}_{2^m}$ where $m \in \{163, 233, 283\}$	"In this paper, we show how the computationally weaker party of a cryptosystem can delegate conversions to the more powerful party by computing all operations directly in the τ-adic domain with a small datapath extension for τ-adic arithmetic."
2018	Khleborodov	[76]	kP	Prime	$\mathbb{F}_p$	Proposes the window NAF form for scalar representation, which is used to develop a scalar multiplication algorithm with mixed coordinates.

TABLE 4. Summary of the reviewed ECC implementations in software denominated lightweight.

Year	Work	Ref.	Impl.	Curve	Field	Sec.	Processor	Runtime (ms)	Freq. (MHz)	Cycles	Power (μW)	Energy (μJ)
2011	Sojka *et al.*	[78]	kP	-	$\mathbb{F}_p$	16	MSP430F5438	1816.28	8.00	-	-	-
				-	$\mathbb{F}_p$	64	MSP430F5438	81457.76	8.00	-	-	-
				-	$\mathbb{F}_p$	80	MSP430F5438	157432.61	8.00	-	-	-
				-	$\mathbb{F}_p$	96	MSP430F5438	259788.89	8.00	-	-	-
2012	Wenger [79]	kP		Montgomery	$\mathbb{F}_p$	80	ATmega128	-	-	5545000	-	-
				GLV	$\mathbb{F}_p$	80	ATmega128	-	-	3930000	-	-
2013	Wenger	[80]	kP	secp160r1	$\mathbb{F}_p$	80	MSP430	-	1.00	5721420		
							openMSP430	-	1.00	5445010	55.90	304.30
				secp192r1	$\mathbb{F}_p$	96	MSP430	-	1.00	9100128		
							openMSP430	-	1.00	8650455	53.90	466.70
				sect163r2	$\mathbb{F}_{2^m}$	80	MSP430	-	1.00	7446677		
							openMSP430	-	1.00	7216905	49.10	354.30
				c2tnb191v1	$\mathbb{F}_{2^m}$	96	MSP430	-	1.00	8610906		
							openMSP430	-	1.00	8376138	55.40	463.80
2013	Namal *et al.*	[81]	Auth.	-	-	-	i5	-	2670.00	-	-	-
2014	Druml *et al.*	[40]	Auth.	secp192r1	$\mathbb{F}_p$	96	Infineon	26.20	30.00	-	-	-
2014	Höller *et al.*	[82]	kP	B163	$\mathbb{F}_{2^m}$	80	-	-	13.56	9700000	-	-
2015	Piñol-Piñol *et al.*	[83]	Sign	RFC6090	$\mathbb{F}_p$	80	STM32F103	2000.00	24.00	-	-	-
						96	STM32F103	3000.00	24.00	-	-	-
						128	STM32F103	6000.00	24.00	-	-	76230.00
			Verify	RFC6090	$\mathbb{F}_p$	80	STM32F103	4000.00	24.00	-	-	-
						96	STM32F103	6000.00	24.00	-	-	-
						128	STM32F103	12000.00	24.00	-	-	153840.00
2016	Liu *et al.*	[15]	kP	Montgomery	$\mathbb{F}_p$	80	ATmega128	-	8.00	5527000	-	-
						96	ATmega128	-	8.00	8837000	-	-
						112	ATmega128	-	8.00	13255000	-	-
						128	ATmega128	-	8.00	18893000	-	-
				GLV+TEC	$\mathbb{F}_p$	80	ATmega128	-	8.00	3135000	-	-
						96	ATmega128	-	8.00	4954000	-	-
						112	ATmega128	-	8.00	7475000	-	-
						128	ATmega128	-	8.00	10501000	-	-
2016	He *et al.*	[45]	Key est.	-	$\mathbb{F}_p$	160	Quad-core	13.44	2450000.00	-	-	-
2016	Kaur *et al.*	[47]	Auth.	B163	$\mathbb{F}_{2^m}$	80	-	192.00	-	-	-	-
2016	Liu *et al.*	[71]	kP	MoTE P159	$\mathbb{F}_p$	80	ATmega128	-	7.37	5468000	-	-
				MoTE P191	$\mathbb{F}_p$	96	ATmega128	-	7.37	8659000	-	-
				MoTE P223	$\mathbb{F}_p$	112	ATmega128	-	7.37	12879000	-	-
				MoTE P255	$\mathbb{F}_p$	128	ATmega128	-	7.37	18270000	-	-
				MoTE P159	$\mathbb{F}_p$	80	MSP430F1611	-	7.37	3476000	-	-
								-	7.37	4276000	-	-
				MoTE P191	$\mathbb{F}_p$	96	MSP430F1611	-	7.37	5381000	-	-
								-	7.37	6686000	-	-
				MoTE P223	$\mathbb{F}_p$	112	MSP430F1611	-	7.37	7151000	-	-
								-	7.37	9808000	-	-
				MoTE P255	$\mathbb{F}_p$	128	MSP430F1611	-	7.37	10952000	-	-
								-	7.37	13782000	-	-
2017	Win *et al.*	[49]	Enc. /Dec.	P160	$\mathbb{F}_p$	80	Galaxy Nexus	5000.00/ 1200.00	-	-	-	-
2017	Mathur *et al.*	[50]	ECDH	-	$\mathbb{F}_p$	-	openmote	682.60	16.00	10912000	-	21490.00
2017	Hasan *et al.*	[58]	BROSMAP	secp224k1	$\mathbb{F}_p$	112	Galaxy Note 5	66.87	-	-	-	-
2017	Sojka and Langendoerfer	[59]	kP	-	$\mathbb{F}_p$	16	MSP430F5438A	-	1.00	409120	-	-
						64	MSP430F5438A	-	1.00	651662120	-	-
						96	MSP430F5438A	-	1.00	2078311103	-	-
2018	Diro *et al.*	[62]	Enc./Dec. (256 bits)	-	$\mathbb{F}_p$	80	i7-6700HQ	3.59/1.24	2600	-	-	-
						128	i7-6700HQ	28.26/6.48	2600	-	-	-
						256	i7-6700HQ	541.79/222.33	2600	-	-	-
2018	Shen el al.	[64]	Auth.	-	$\mathbb{F}_q$	-	Pentium	41.367	3300	-	-	-
2018	Mood *et al.*	[65]	kP	P521	$\mathbb{F}_p$	256	Cortex-M3	1263	100	-	-	-
				Curve25519	$\mathbb{F}_p$	128	Cortex-M3	87	100	-	-	-

literature denominated lightweight. In this case we consider that all the works surveyed for FPGA implementation of ECC feature the *Architecture* characteristic.

3) HARDWARE CIRCUITS

Once an architecture has been conceptually designed and described, there is still an implementation step required to obtain the physical realization of the system. The implementation of hardware architectures can be approached from two main technologies which currently rule the market: using an FPGA or creating an ASIC. In both cases, a formal description of the architecture is used to generate a resistor-transistor level (RTL) description of the circuit. This new description is then mapped to a reconfigurable array of Look-Up-Tables, Flip-Flops, and other generics in the physical FPGA board. Or compiled using CMOS libraries into an array of transistors which is then manufactured into a silicon chip. The main advantages of the FPGA technology is the rapid development and implementation process which make it captivating for prototyping and testing. In the case of ASIC the main advantage lies on the technology level optimizations (*Circuit*) which can be applied to the chip in order to achieve greater efficiency.

The ASIC implementations of ECC found in the literature are summarized in Table 6. In this case we consider that all the ASIC proposals surveyed perform optimizations at circuit level and hence feature the *Circuit* characteristic according to the abstraction levels proposed. The benchmark for ASIC implementation reports the technology used, the

TABLE 5. Summary of the reviewed ECC implementations in FPGA denominated lightweight.

Year	Work	Ref.	Impl.	Curve	Field	Sec.	FPGA	FF	LUT	SLC/ ALM/LE	BRAM/ MEM	DSP	Fmax (MHz)	Cycles
2011	Varchola *et al.*	[84]	kP	P224	$\mathbb{F}_p$	112	XC2VP7-5FG456	-	-	773	3	1	210	1722088
								-	-	1158	3	4	210	765072
							A2F500M3G-STD	-	-	3690	12	-	109	1722088
								-	-	7832	12	-	109	765072
				P256	$\mathbb{F}_p$	128	XC2VP7-5FG456	-	-	774	3	1	210	2103941
								-	-	1158	3	4	210	949951
							A2F500M3G-STD	-	-	3690	12	-	109	2103941
								-	-	7832	12	-	109	949951
2012	Trujillo-Olaya *et al.*	[4]	kP	-	$\mathbb{F}_{2^m}$	≤ 20	EP3SE50F780C2	1704	2049	-	820	-	173	5576
								1551	2153	-	800	-	177	5699
					$\mathbb{F}_{2^m}$	≤ 26	EP3SE50F780C2	2093	2551	-	1060	-	161	9116
								1902	2451	-	1060	-	171	9328
					$\mathbb{F}_{2^m}$	≤ 30	EP3SE50F780C2	2363	2808	-	1220	-	152	11956
					$\mathbb{F}_{2^m}$	≤ 35	EP3SE50F780C2	2696	3158	-	1420	-	147	16046
					$\mathbb{F}_{2^m}$	≤ 39	EP3SE50F780C2	2958	3440	-	1580	-	134	19750
								2239	3129	-	1580	-	148	20224
					$\mathbb{F}_{2^m}$	≤ 44	EP3SE50F780C2	3289	3958	-	1780	-	115	24920
								2952	4258	-	1780	-	138	25365
					$\mathbb{F}_{2^m}$	≤ 48	EP3SE50F780C2	3547	4227	-	1940	-	114	29488
								3183	4705	-	1940	-	132	29876
					$\mathbb{F}_{2^m}$	≤ 51	EP3SE50F780C2	3750	4405	-	2060	-	110	33166
					$\mathbb{F}_{2^m}$	≤ 56	EP3SE50F780C2	4081	4766	-	2260	-	109	39776
								3650	5277	-	2260	-	130	40341
					$\mathbb{F}_{2^m}$	≤ 63	EP3SE50F780C2	4555	5226	-	2540	-	105	50038
					$\mathbb{F}_{2^m}$	≤ 80	EP3SE50F780C2	5725	6536	-	3260	-	96	80685
								5107	8203	-	3260	-	104	81826
2012	Driessen *et al.*	[85]	kP	secp160r1	$\mathbb{F}_p$	80	Spartan-3	507	1028	569	3	1	-	-
							Spartan-6	482	630	221	3	1	-	-
				P256	$\mathbb{F}_p$	128	Spartan-3	507	1028	569	3	1	-	-
							Spartan-6	482	630	221	3	1	-	-
2013	Schramm and Grzemba	[86]	kP	P192	$\mathbb{F}_p$	80	-	-	-	-	-	-	-	-
				P224	$\mathbb{F}_p$	112	-	-	-	-	-	-	-	-
				P256	$\mathbb{F}_p$	128	-	-	-	-	-	-	-	-
				P384	$\mathbb{F}_p$	192	-	-	-	-	-	-	-	-
				P521	$\mathbb{F}_p$	256	-	-	-	-	-	-	-	-
2013	Wenger *et al.*	[87]	kP	B163	$\mathbb{F}_{2^m}$	80	XC2VP7	-	-	-	-	-	-	-
2016	Roy *et al.*	[88]	kP	P256	$\mathbb{F}_p$	128	Virtex-5	35	212	81	24	8	172	1903650
							Spartan-6	35	193	72	24	8	156	1903650
2016	Yalçin	[89]	kP	-	$\mathbb{F}_p$	80	-	-	-	937	2	-	-	490000
						128	-	-	-	1036	2	-	-	480000
2017	Salman *et al.*	[90]	kP	-	$\mathbb{F}_p$	96-256	xc7z020clg484-3[a]	939	1118	318	2	-	164	27991781[b]
								4606	2765	1224	2	-	178	3392812[b]
							xc7a100tcsg324-3[a]	939	1135	312	2	-	187	27991781[b]
							xc7vx485tffg1761-3[a]	948	1166	325	2	-	155	27991781[b]
							xc7vx485tffg1761-3[a]	4606	3144	996	2	-	243	3392812[b]
							xc6vlx240tff1156-3[a]	4606	2840	1174	2	-	239	3392812[b]
							5SGXEA7K2F40C3[a]	647	961	588	20480	-	224	27991781[b]
								5084	4841	2629	20480	-	237	3392812[b]
							5CEBA4F23C7[a]	571	961	588	20912	-	119	27991781[b]
							EP4SE530H35C4[a]	952	933	899	20480	-	185	27991781[b]
								4617	4031	3681	20480	-	191	3392812[b]
							M2GL005S:1FG484[a]	1269	2079	2522	10	-	97	3392812[b]
								4926	5927	6839	10	-	93	3392812[b]

a Best results are retrieved.
b Average latency for the different field lengths supported.

implementation costs in GEs, the latency of the circuit, and the power and energy estimations when available.

CAT3 STRATEGIES AND REMARKS

Software implementations are intended for providing security services. As it can be noted from Table 4, only seven surveyed software implementations perform scalar multiplication routines. In most of the cases the curves used are defined, ranging from a curve-optimized solution to offering support for a family of curves. The security levels offered range from 80 to 128 bits in the majority of the cases. The preferred optimization goal identified was to improve performance, in the form of shorter runtimes.

The implementation platforms are not uniform in architectural type nor register size. Some of the surveyed papers developed software implementations for generic processors, which are usually found in environments where regular ECC can be used. Most of the implementations targeted processors are available in mobile systems—like smartphones. Whereas the processing power of these chips is not a problem, they can still benefit from ECLC traits such as low energy consumption and small bandwidth requirements. Lastly, only a handful of the implementations targeted the lower end of processors which are found in WSN motes and RFID tags. In these cases both the processor time and the energy footprint of the implementation ought to be observed closely.

As for evaluation of the proposals, some of the metrics reported are the runtime, the operational frequency achievable, the latency, and the energy consumption. These are in line with software implementations.

TABLE 6. Summary of the reviewed ECC implementations in ASIC denominated lightweight.

Year	Work	Ref.	Impl.	Curve	Field	Sec.	Tech.	Area (GE)	Freq (MHz)	Cycles	Power (μW)	Energy (μJ)
2012	Wenger	[79]	kP	Montgomery	$\mathbb{F}_p$	80	130nm	20980	-	1300000	-	-
2013	Wenger	[87]	kP	secp160r1	$\mathbb{F}_p$	80	130nm	17738	13.56	37168	561.00	1539.00
								22537	13.56	1298	1013.00	97.00
				secp192r1	$\mathbb{F}_p$	96	130nm	16877	13.56	55365	640.00	2615.00
								23825	13.56	1813	1084.00	145.00
				secp224r1	$\mathbb{F}_p$	112	130nm	17215	13.56	86058	663.00	4208.00
								25113	13.56	2469	1032.00	188.00
				secp256r1	$\mathbb{F}_p$	128	130nm	19506	13.56	130695	656.00	6320.00
								27244	13.56	3367	1031.00	256.00
2013	Wenger *et al.*	[91]	kP	B163	$\mathbb{F}_{2^m}$	80	130nm	-	-	-	-	-
2013	Wenger	[80]	kP	sect163r2	$\mathbb{F}_{2^m}$	80	130nm	11778	1.00	342724	93.80	32.10
								12662	1.00	174910	112.90	19.70
								13307	1.00	94882	152.40	14.50
								14552	1.00	54376	181.7	9.90
								9624	1.00	467370	66.10	30.90
								10405	1.00	303202	77.6	23.50
								11022	1.00	224222	73.6	16.50
								12270	1.00	182130	70.00	12.80
2014	Azarderakhsh *et al.*	[69]	kP	K163	$\mathbb{F}_{2^m}$	80	65nm	11571	0.10	106700	0.66	0.65
								11571	13.56	106700	77.20	0.61
2014	Höller *et al.*	[82]	kP	B163	$\mathbb{F}_{2^m}$	80	220nm	7510	13.56	12100000	-	-
								8920	13.56	9400000	-	-
								10550	13.56	7000000	-	-
								10290	13.56	5100000	-	-
								6180	13.56	2800000	-	-
2015	Roy *et al.*	[72]	kP	K283	$\mathbb{F}_{2^m}$	128	130nm	10204	16.00	1566000	6.11[a]	-
2015	Koziel *et al.*	[73]	kP	-	$\mathbb{F}_{2^m}$	80	65nm	11219	-	177707	-	-
						112	65nm	15177	-	351856	-	-
						128	65nm	19332	-	512555	-	-
2016	Bosmans *et al.*	[92]	kP	P256	$\mathbb{F}_p$	128	130nm	11727	16.00	6200000	-	-
2016	Yalçin *et al.*	[89]	kP	-	$\mathbb{F}_p$	80	180nm	11366[b]	-	490000	-	-
						128	180nm	12324[b]	-	840000	-	-
2016	Goyal and Sahula	[93]	Key est.	-	$\mathbb{F}_p$	-	90nm	$0.032mm^2$	-	22.4 ms	-	-
2018	Järvinen *et al.*	[75]	kP	K163	$\mathbb{F}_{2^m}$	80	130nm	75[c,d]	-	49700[d]	-	-
				K233	$\mathbb{F}_{2^m}$	112	130nm	75[c,d]	-	99686[d]	-	-
				K283	$\mathbb{F}_{2^m}$	128	130nm	75[c,d]	-	146118[d]	-	-

a Calculated at 1MHz.
b External memory is required.
c Cost of the datapath extension for a τ-adic addition.
d Only best results are retrieved.

All the architectural designs found in the literature focus on the computation of the scalar multiplication. Either as a processor, or a co-processor, or as an independent dedicated core. These works often define the curve or family of curves which is used, with security levels ranging from 80 to 128 bits in most of the cases. The FPGAs utilized are divided between Xilinx, Altera, and MicroSemi boards.

As implementation metrics the FPGA resources are reported for almost all of the cases. The operational frequency and latency cycles are also almost always reported. At difference of software implementations however, the main optimization goal is reducing the area or the resource usage.

Similarly as with the case of FPGA implementations, all of the ASIC related works propose designs to perform scalar multiplications. In these references, however, it is possible to note how the number of supported curves per proposal is reduced from an average of 3.7 to 1.85 curves per work. This is evidence of the main difference between FPGA and ASIC works: FPGAs allow rapid prototyping and some generality pre and post synthesis, ASIC designs are more optimized to take advantage of the domain parameters selected.

The libraries used in the surveyed works range from 180nm to 65nm. As implementation metrics, the works report area in GE, operational frequency, latency cycles, power dissipation, and energy consumption. As in the case of FPGA implementations, ASIC oriented works seem to favor area reducing optimizations.

The use of specific instructions which help to improve the performance of the implementation is also valid for software realizations. Some libraries have been made available with highly optimized routines which improve the processors efficiency. Some rely on exploiting the word size of the architecture in order to achieve instruction-level parallelism whereas others make use of mathematical cores available to the processor such as floating point units and DSPs.

To create lightweight architectures it is important to first analyze the algorithm and determine, considering that a hardware implementation is sought, modifications in order to simplify the design. Compared to software realizations, hardware solutions often have trouble dealing with the control structure required by the system. A simplified architecture implies a simplified controller. Complexity in the control of the circuit can generate resource overhead and increased latency.

The processing generality required in the implementation is what determines the hardware design approach. Complex algorithms which rely heavily on conditionals or that require a wide set of subroutines are easier to implement in an application specific processor. If the implementation already considers the use of a processing system, hardware acceleration can be achieved with a dedicated co-processor and

mapped memory. If the algorithm is simple enough and repetitive, then a hardware architecture is the best choice. All of these approaches can be used to solve a problem, the difference will be noticeable in the design complexity and the resource consumption of the final product.

Common approaches identified from the literature in the design of lightweight architectures for ECC involve:

- Determining if a specific module is really necessary or if the operation can be performed differently (e.g. squarings, inversions).
- Adjusting the word size of field multipliers to look for the best trade-off between resources and performance.
- Adjusting the width of the datapath to reduce the resources required to process the data.
- Modifying the way in which the data is stored in the system to reduce read/write times or to simplify the control.

In regards to circuits, the implementation might be improved in order to reduce the physical size, reduce the latency in processing, memory access times, or energy consumption. Selecting the most suitable implementation technology for the application (e.g. LUT-4 vs. LUT-6), selecting a specific type of memory (e.g. BRAM vs. distributed), preventing spurious computations, and applying clocking techniques (e.g. clock enable and multiclock domain), are all circuit specific optimizations which can be performed to almost any architectural design in order to improve the efficiency of the implementation.

V. ASSESSMENT OF THE SURVEYED WORKS

In the previous section we provided a classification based on objective data retrieved from the surveyed works. We now try to evaluate all the papers as a set, by using the characteristics described on Section III. Table 7 summarizes our observations.

This data is can be used for classifying the surveyed papers using clustering techniques. Other analysis can also be drawn from this information. These are provided in the next section.

VI. DATA ANALYSIS

The information retrieved from the surveyed papers can provide insights regarding the design of novel elliptic curve based cryptosystems that can be denominated lightweight. For this reason it is important to present a quantitative analysis of the related works in the literature. This section is dedicated to provide a modern analysis and trends of the surveyed papers.

A. STATISTICS

We now review some quantitative measurements for the different characteristics reported in Table 7.

1) LIGHTWEIGHTNESS

From each paper we tried to determine the main reason for the work to be denominated lightweight. In some of the cases the motivation was explicitly stated in the papers. In others it was implicit and a more complete analysis was required to assess this characteristic as accurately as possible. All these descriptions, as reported in Table 7, were classified in six main reasons. The frequency for each one of these main classes is illustrated in Fig. 12.

As it can be noted, the most predominant reasons to define a proposal as lightweight are by designing a lightweight protocol, by optimizing a lightweight implementation, and by proposing some lightweight variant of ECC.

Lightweight protocols rely on reducing the communications and memory overheads. Reducing the operations count, the operations diversity, the transmission overhead, the memory requirements, all are factors that should be covered in their design.

A good part of the publications surveyed cover the design and implementation of a complete cryptosystem. It is thus important to keep in mind that an ECLC solution starts with the elliptic curve: selecting the field, curve, representation, and algorithms. These are then translated into architectures and circuits for implementation. Each one of these steps must be aware of the application constraints.

Other reasons to define a paper as lightweight include the use of ECC over other PKC instances, the modification of a previously existing solution, and comparisons with the state of the art. These works showcase that the term ''lightweight'' is often used without the adequate substance to back the definition. Whereas it is not harmful to use the concept subjectively, it has the negative effect of misleading the researches which might be interested in this field of study.

2) ABSTRACTION GOALS

We have also analyzed to what extent the five abstractions levels for ECLC are observed. The results are presented in Fig. 13. As it can be noted, the quantity of papers which possess each characteristic has an inverse relation with the degree of specificity (from design to implementation). It is reasonable to expect that full systems which require longer development times (processors, co-processors, standalone cores) would be scarcer than protocols.

What we found surprising is that only 60% of the surveyed papers make use of the underlying system characteristics to create ECLC solutions. A possible explanation is that works without a clear problem to solve (technological challenge) seek to maintain generality and provide functionality for multiple systems. This contradicts the expectation that an ECC solution called ''lightweight'' is optimized for a specific technology. Even if the implementation scope is not clear, it would be convenient to at least observe the standards that govern the technology and derive optimizations from them.

3) DESIGN GOALS

The study of the design goals identified from the papers can provide insights into the differences between ECLC and other classes of lightweight cryptography. Fig. 14 shows how the design goals are represented in the surveyed papers. It can be noted that a primordial factor for ECLC systems

TABLE 7. Summary of the different reviewed works which have been called "lightweight" by the authors in the publication or related media (presentations). The reason for lightweightness (A), abstraction levels (B), goals (C), system (D), implementation technology (E), and security levels are all reported for each paper. The ✓ symbol indicates that an entry complies with the characteristic in that specific column. The different design goals specified in the works are summarized in priority order with 1st being the most important.

ID	Year	Work	Ref.	A	B. Abstraction					C. Goals					D	E. Technology			F. Security		
					B1	B2	B3	B4	B5	C1	C2	C3	C4	C5		E1	E2	E3	F1	F2	F3
A-01	2007	Kim *et al.*	[36]	A5	✓	✓	✓				1st	3rd	2nd		D2				✓		
A-02	2010	Kim *et al.*	[37]	A5	✓	✓				1st		2nd			D2						
A-03	2010	Sojka *et al.*	[68]	A3	✓	✓	✓			1st					D3				✓		
A-04	2011	Varchola *et al.*	[84]	A2				✓		2nd	1st						✓			✓	✓
A-05	2011	Sojka *et al.*	[78]	A3	✓	✓	✓			1st					D3	✓			✓		
A-06	2012	Ju	[38]	A1	✓	✓				1st	3rd		2nd		D3						
A-07	2012	Wenger and Großschädl	[79]	A2			✓	✓		1st	3rd	2nd			D1			✓	✓		
A-08	2012	Trujillo-Olaya *et al.*	[4]	A3	✓		✓			2nd	1st				D2		✓				✓
A-09	2012	Driessen *et al.*	[85]	A5	✓	✓		✓		2nd	1st			3rd	D6		✓		✓		✓
A-10	2013	Wenger	[87]	A2				✓	✓	1st	2nd				D2		✓	✓	✓	✓	✓
A-11	2013	Wenger *et al.*	[91]	A2				✓			2nd	1st			D2			✓	✓		
A-12	2013	Bakhache *et al.*	[39]	A1	✓	✓				2nd			1st		D7						
A-13	2013	Wenger	[80]	A2				✓	✓		2nd			1st	D3	✓		✓	✓		
A-14	2013	Namal *et al.*	[81]	A4						1st		2nd			D7	✓					
A-15	2013	Schramm and Grzemba	[86]	A2				✓			1st				D6		✓		✓	✓	✓
A-16	2014	Druml *et al.*	[40]	A1	✓	✓	✓			1st	2nd				D2	✓			✓		
A-17	2014	Yao *et al.*	[41]	A4						1st		3rd		2nd	D7						
A-18	2014	Azarderakhsh *et al.*	[69]	A2			✓	✓	✓		1st				D1			✓	✓		
A-19	2014	Höller *et al.*	[82]	A2			✓	✓	✓	1st	2nd				D2	✓		✓	✓		
A-20	2014	He *et al.*	[42]	A1	✓	✓				1st	3rd	2nd			D2						
A-21	2014	Liu *et al.*	[70]	A3	✓		✓			1st		2nd			D3						
A-22	2014	Liu *et al.*	[14]	A3	✓		✓			1st	2nd				D3	✓			✓	✓	✓
A-23	2015	Piñol-Piñol *et al.*	[83]	A2	✓		✓			1st	2nd				D1	✓			✓		✓
A-24	2015	Roy *et al.*	[72]	A3	✓		✓	✓	✓	3rd	2nd	1st			D1			✓			✓
A-25	2015	Koziel *et al.*	[73]	A3	✓		✓	✓	✓	2nd	1st				D1			✓	✓	✓	✓
A-26	2016	Bosmans *et al.*	[92]	A2				✓	✓		2nd	1st			D1			✓			✓
A-27	2016	Chaudhry *et al.*	[43]	A6	✓	✓				2nd		1st			D1				✓		
A-28	2016	Yalçin	[89]	A2				✓	✓		1st				D1		✓	✓	✓		✓
A-29	2016	Roy *et al.*	[88]	A2				✓			1st				D1		✓				✓
A-30	2016	Liu *et al.*	[15]	A3	✓		✓			1st		2nd			D3	✓			✓	✓	✓
A-31	2016	Reddy *et al.*	[44]	A1	✓		✓			2nd		1st			D7						
A-32	2016	He *et al.*	[45]	A1	✓	✓				1st		2nd			D5	✓					✓
A-33	2016	Lavanya and Natarajan	[46]	A4						1st	2nd				D1						
A-34	2016	Kaur *et al.*	[47]	A1	✓		✓			2nd		1st			D2	✓			✓		
A-35	2016	Goyal and Sahula	[93]	A4							2nd			1st	D1			✓	✓		
A-36	2016	Liu *et al.*	[71]	A3	✓		✓			1st	2nd				D1				✓	✓	✓
A-37	2017	Zhang *et al.*	[48]	A1	✓	✓				2nd		1st			D1						
A-38	2017	Win *et al.*	[49]	A6	✓	✓	✓			1st		2nd			D1	✓			✓		
A-39	2017	Salman *et al.*	[90]	A2				✓		1st	2nd	3rd			D6		✓		✓	✓	✓
A-40	2017	Mathur *et al.*	[50]	A5	✓	✓	✓			2nd		1st			D1	✓					
A-41	2017	Mahmood *et al.*	[51]	A1	✓	✓				1st		3rd	2nd		D5				✓		
A-42	2017	Diro *et al.*	[52]	A4						2nd		1st			D1				✓		
A-43	2017	Khleborodov	[74]	A3			✓			1st	3rd			2nd	D1						
A-44	2017	Badra and Zeadally	[54]	A6	✓	✓				3rd		2nd	1st		D5				✓		
A-45	2017	Diro *et al.*	[53]	A4						2nd		1st			D1				✓		
A-46	2017	Lavanya and Natarajan	[55]	A5	✓	✓				2nd				1st	D1						
A-47	2017	Meddah *et al.*	[56]	A6	✓	✓				1st					D4						
A-48	2017	Mohammedi *et al.*	[57]	A1	✓	✓				2nd			1st		D4						
A-49	2017	Hasan *et al.*	[58]	A4						1st		2nd			D1	✓				✓	
A-50	2017	Sojka *et al.*	[59]	A3	✓	✓	✓			1st					D3	✓			✓		
A-51	2018	Shen *et al.*	[60]	A1	✓	✓				2nd		3rd		1st	D4						
A-52	2018	Tewari *et al.*	[61]	A1	✓	✓					2nd	1st	3rd		D2					✓	
A-53	2018	Diro *et al.*	[62]	A4						2nd	3rd	1st			D1	✓			✓		✓
A-54	2018	Järvinen	[75]	A3	✓	✓	✓	✓	✓	3rd	2nd	1st			D1			✓	✓	✓	✓
A-55	2018	Vaniprabha and Poongodi	[63]	A4						3rd		1st	2nd		D4						
A-56	2018	Shen *et al.*	[64]	A1	✓	✓				2nd	3rd	1st			D4	✓					
A-57	2018	Mood and Nikooghadam	[65]	A1	✓	✓	✓			3rd		1st	2nd		D5	✓					✓
A-58	2018	Khleborodov	[76]	A3			✓			1st	3rd			2nd	D1						
A-59	2018	Almulhim and Zaman	[66]	A1	✓	✓						1st		2nd	D4						
A-60	2018	Mohammedi *et al.*	[67]	A6						2nd	3rd	1st			D4	✓			✓	✓	✓

is to achieve good performance, then security, and third to reduce the resource requirements of the system (hardware, bandwidth, energy). ECC generally is used to because of improved security, thus it is not surprising that security is a top goal in the surveyed papers. But it is also true that ECC systems suffer from lengthy latencies. Consequently, unlike other types of lightweight cryptography, ECLC will try first to address the performance constraints of the system and then observe the resource requirements of the application.

4) TARGET SYSTEM

Other aspect that is important to remark from the surveyed papers is the application scope for which they were designed. Lightweightness certainly is tied to the technology, identified in this work as the first abstraction level. Lightweight cryptography is the cryptography that has been tailored for the constrained systems. These systems, as it is shown in Fig. 15, evolve.

From our study, the IoT domain occupies the first place of ECLC application scope from the surveyed works.

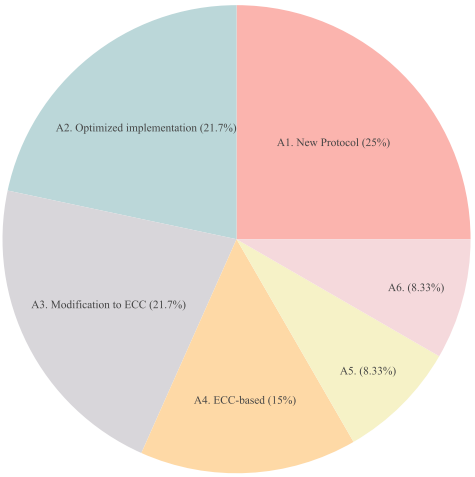

FIGURE 12. Distribution of the main reasons to denominate use the adjective lightweight in the surveyed papers.

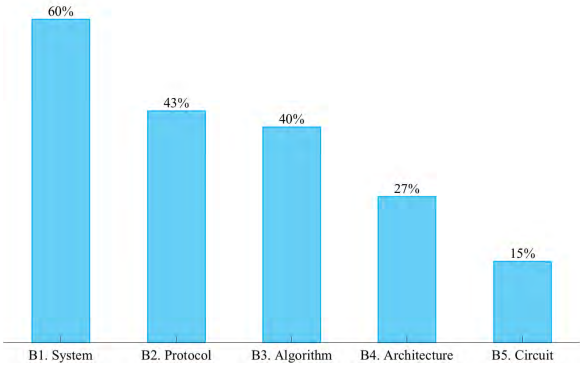

FIGURE 13. Distribution of the surveyed works for the different abstraction levels. Note that the total does not equal 100% since some works appear in multiple categories.

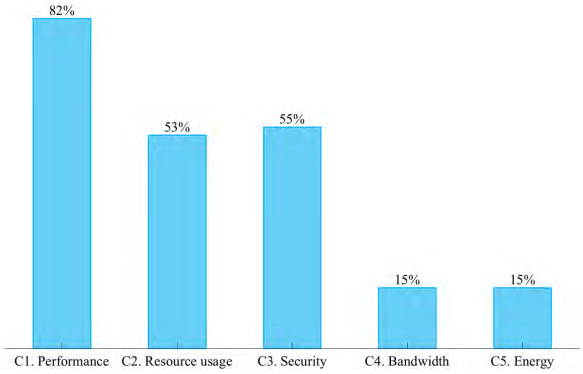

FIGURE 14. Percentage of publications associated with the different design goals presented in the survey. Note that the total does not equal 100% since some works appear in multiple categories.

This novel technology promises to bring changes to the very structure of society, hence its relevance. In second and third spots we can find RFID and WSN, which are often defined as the basis of IoT. Overall, these three applications cover ∼70% of the surveyed works. Notable mentions can

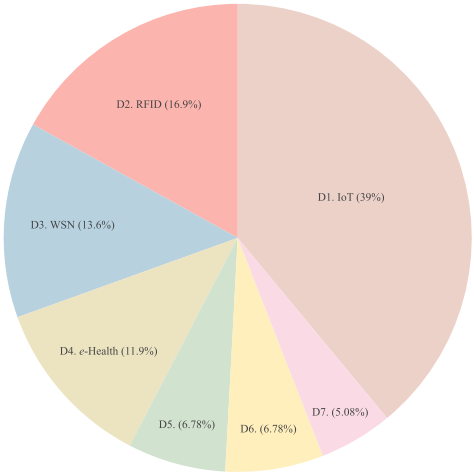

FIGURE 15. Distribution of the application scope reported for the different surveyed works.

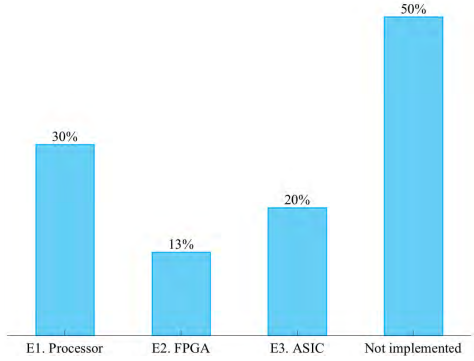

FIGURE 16. Implementation technology for the different proposals surveyed. Note that the total does not equal 100% since some works appear in multiple categories.

be made for e-Health and Smart Grids, which are emerging technologies.

5) IMPLEMENTATION TECHNOLOGY

This review revealed that only 50% of the surveyed papers were implemented. Dividing the implementation technologies between software and hardware, they come even with about 30% of the instances each. Specifically in hardware, 13% of the works were implemented in FPGA and 20% were implemented in ASIC. These statistics are presented in Fig. 16.

6) SECURITY

Fig. 17 illustrates the quantity of works which propose to use the different security levels. As it can be noted, more than half of the works surveyed propose to use security levels which are no longer recommended by NIST, and a third of them does not propose anything concrete.

B. QUALITATIVE ANALYSIS

An analysis which relies on the characteristics of the surveyed works if provided in the following.

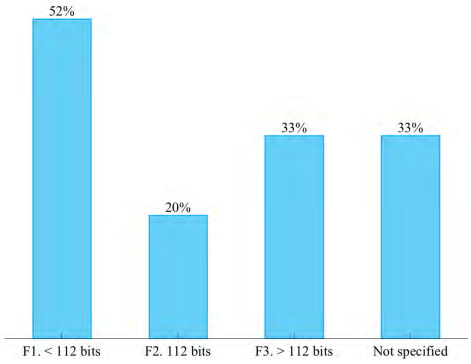

FIGURE 17. Security level recommended by the different proposals surveyed. Note that the total does not equal 100% since some works appear in multiple categories.

1) CLUSTERING

Using the data collected in Table 7 it is possible to classify the surveyed papers in different groups. For doing this we created an undirected graph where a matching characteristic between two papers is expressed as an edge. The created graph contains 60 nodes and 3921 edges[4] and is presented in Fig, 18.

In that illustration each node represents a paper and each edge represents a matching characteristic. The size of the nodes is determined by their centrality and the weight of the edges is given by the sum of all the edges between two nodes. The different colors represent different communities. Edges with weight lower than 3 have been removed for visualization; the topology is a Circular Layout and the image was created using Gephi.[5]

In the graph it is possible to appreciate the connectivity between the clusters by the coloring of the edges. Uniform colors indicate inter-cluster links, mixed-color edges represent links between clusters. By using the Modularity analysis of Gephi it was possible to divide the papers into four groups as detailed in Table 8. A resolution of 0.81 with randomization and weights was used for this analysis.

The first group with *Cluster 0* is dominated by works which are said to be lightweight by proposing an *Optimized implementation* (A2). The most predominant abstraction levels on this class are the *architecture* (B4) and the *circuit* (B5). The predominant optimization goals involve *hardware resources* (C2) in all the cases. In this cluster we can find almost all the FPGA and ASIC implementations.

The second set with *Cluster 1* includes several works which propose a *Modification to ECC* (A3). The predominant abstraction levels are *system* (B1) and *algorithm* (B3). The main goal for all but one instances is to improve performance, and almost all the implementations included target processors.

In the third group, *Cluster 3*, we can note that the main reason for calling the works lightweight is by designing a *New protocol* (A1). Almost all the instances observed the *system* (B1) and the *protocol* (B2) abstraction levels. In these instances the optimization goal is *performance* (C1) or *security* (C2).

This method resulted effective to divide the papers in possible sets of interest. However, when the number of surveyed papers is extensive, it becomes difficult to collect the data needed for the classification. This is more relevant in instances where it is sought to initially filter papers which might be related with the topic of interest and thus be attractive for the community. Modern data mining algorithms allow extracting metadata from text sources. These data can then be used to classify a wide set of papers.

2) EXTENDED CLUSTERING

Communities are groups which can be found in sets of items which share certain relationship. For this work these items are publications. The set of documents surveyed, their references, and their citations can be represented as a graph. Classification algorithms can then be applied over these graphs to detect communities based on the graph's topology.

Parting from all the documents surveyed we extracted all of their references and linked them. In this way, multiple documents in the initial set can make reference to the same publication and become connected. Next, using Google Scholar[6] we retrieved all the publications which reference the original set of documents and linked them as well. In this second step a new type of connections appeared in the original set. Some works became related through the papers that cite multiple elements of the original set. The resulting graph contained 1640 nodes and 2160 edges.[7] The weight of the edges between documents in the original set was increased by 2 and the weight of the citations was increased by 1. The resulting graph is shown in Fig. 19 with a representation obtained using the Circle Pack Layout in the Gephi[8] tool.

For the Modularity analysis we used a Resolution of 1.0 and allowed Randomization and Weight usage. With this analysis 19 clusters were identified. Details regarding the top six communities are presented in Table 9. Although the table only includes papers from the original set, the cardinality for each group is bigger.

The first group (*Cluster 1*) contains 211 elements (12.87%). Each one of the main papers included is said to be lightweight by proposing a *Modification to ECC* (A3), almost all have the *system* (B1) and *algorithm* (B2) abstraction levels assigned, and all seek to improve performance as their main goal. The set with *Cluster 2* contains papers of ASIC implementations for the most part, whereas in the set with *Cluster 3* are FPGA design proposals. Groups with *Cluster 4, 14, 16* include a mix of protocols and software implementations with varying differences between them. For example in *Cluster 4* we can note a predilection for performance as design goal and lower security specifications.

[4]The graph data is available at https://www.tamps.cinvestav.mx/~datasets/
[5]https://gephi.org/
[6]https://scholar.google.com
[7]The graph data is available at https://www.tamps.cinvestav.mx/~datasets/
[8]https://gephi.org/

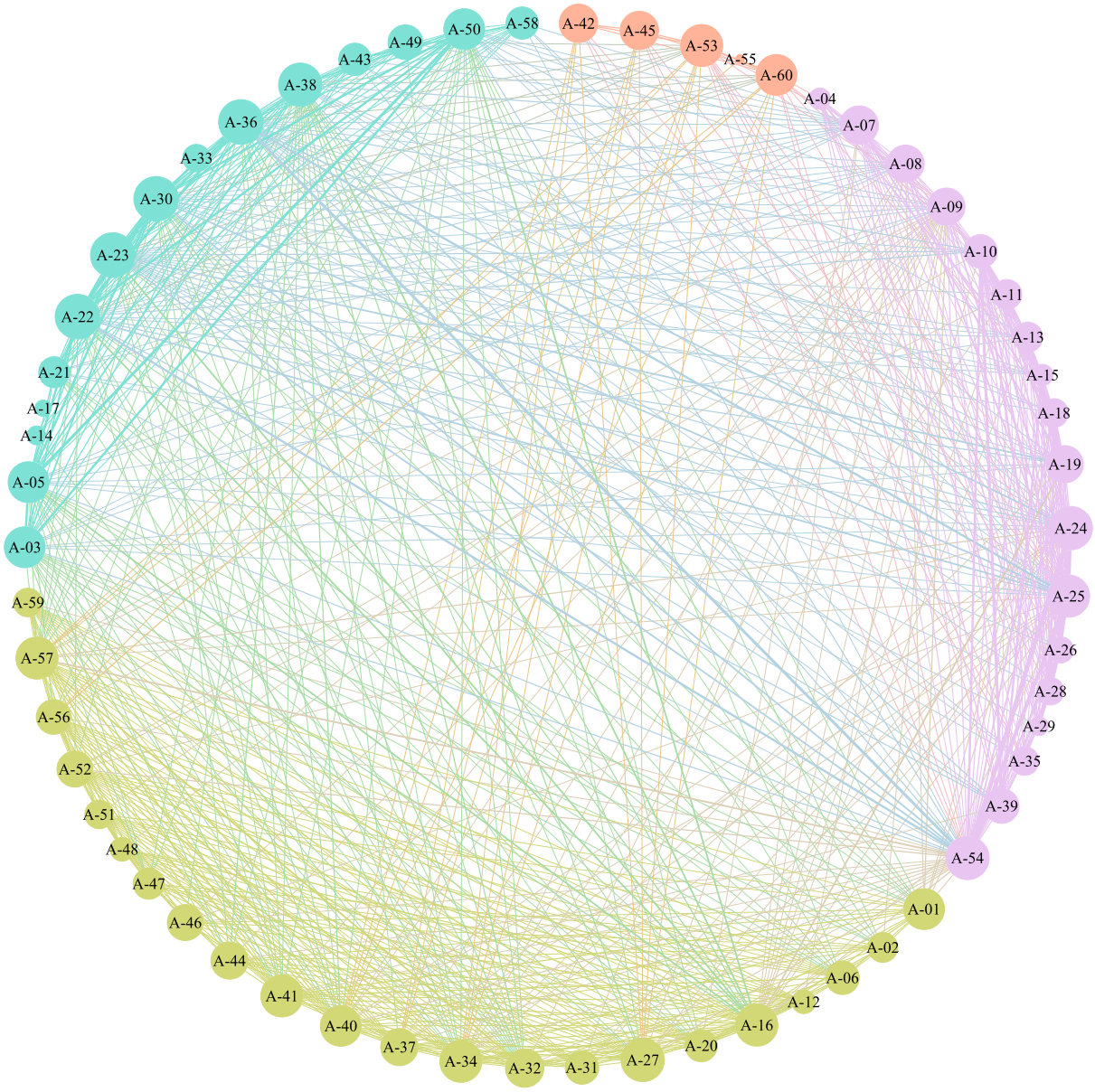

FIGURE 18. Analysis of the surveyed papers, as presented in Table 7, as a graph. The different colors represent different communities.

3) CENTRALITY

In the previous analysis we focused on the degree and quality of the connections of a node. However, in graphs there are other metrics which can provide useful information such as the eigenvector centrality. An analysis based on this metric can provide a different set of relevant papers for a researcher interested in the topic. In Fig. 20 we illustrate the reference tree derived from an analysis making emphasis on the centrality of certain nodes. The Fruchtermain-Reingold layout is used in this case.

Table 10 provides the list of the publications with centrality scores higher than 0.3 in the provided graph.

It is true that these analyses are not flawless, however, they provide insights regarding references that might be worth reading and related papers. Moreover, they represent a viable option when the number of papers is big.

4) KEYWORDS DETECTION

An easy way to review a wide set of documents is to perform keywords or topic extraction. These techniques use text from the papers such as the titles, the abstracts, or the whole document to extract words which are relevant to the document. These data can then be used to classify the documents. The main challenge is the processing power required for big sets of text.

A popular modern representation of the keywords of a set of texts are the word clouds. These diagrams, as the ones presented in Fig. 1 and Fig. 21, allow a visual representation

TABLE 8. Top three communities identified in the set of documents surveyed, classified using the data in Table 7. These three communities include ∼ 92% of the surveyed papers.

ID	Year	Work	Ref.	A	B. Abstraction					C. Goals					D	E. Technology			F. Security			Cluster
					B1	B2	B3	B4	B5	C1	C2	C3	C4	C5		E1	E2	E3	F1	F2	F3	
A-04	2011	Varchola *et al.*	[84]	A2				✓		2nd	1st						✓			✓	✓	0
A-07	2012	Wenger and Großschädl	[79]	A2			✓	✓		1st	3rd	2nd			D1			✓	✓			0
A-09	2012	Driessen *et al.*	[85]	A5	✓	✓		✓		2nd	1st			3rd	D6		✓		✓		✓	0
A-10	2013	Wenger	[87]	A2				✓	✓	1st	2nd				D2		✓	✓	✓	✓	✓	0
A-11	2013	Wenger *et al.*	[91]	A2				✓			2nd	1st			D2			✓	✓			0
A-13	2013	Wenger	[80]	A2				✓	✓		2nd			1st	D3	✓		✓	✓			0
A-15	2013	Schramm and Grzemba	[86]	A2				✓			1st				D6		✓		✓	✓	✓	0
A-18	2014	Azarderakhsh *et al.*	[69]	A2			✓	✓	✓		1st				D1			✓	✓			0
A-19	2014	Höller *et al.*	[82]	A2			✓	✓	✓	1st	2nd				D2	✓		✓	✓			0
A-24	2015	Roy *et al.*	[72]	A3	✓		✓	✓	✓	3rd	2nd	1st			D1			✓			✓	0
A-25	2015	Koziel *et al.*	[73]	A3	✓		✓	✓	✓	2nd	1st				D1			✓	✓	✓	✓	0
A-26	2016	Bosmans *et al.*	[92]	A2				✓	✓		2nd	1st			D1			✓			✓	0
A-28	2016	Yalçin	[89]	A2				✓	✓		1st				D1		✓	✓	✓		✓	0
A-29	2016	Roy *et al.*	[88]	A2				✓			1st				D1		✓				✓	0
A-35	2016	Goyal and Sahula	[93]	A4							2nd			1st	D1			✓	✓			0
A-39	2017	Salman *et al.*	[90]	A2				✓		1st	2nd	3rd			D6		✓		✓	✓	✓	0
A-54	2018	Järvinen	[75]	A3	✓	✓	✓	✓	✓	3rd	2nd	1st			D1			✓	✓	✓	✓	0
A-03	2010	Sojka *et al.*	[68]	A3	✓	✓	✓			1st					D3				✓			1
A-05	2011	Sojka *et al.*	[78]	A3	✓	✓	✓			1st					D3	✓			✓			1
A-08	2012	Trujillo-Olaya *et al.*	[4]	A3	✓		✓			2nd	1st				D2		✓				✓	1
A-14	2013	Namal *et al.*	[81]	A4						1st		2nd			D7	✓						1
A-17	2014	Yao *et al.*	[41]	A4						1st		3rd		2nd	D7							1
A-21	2014	Liu *et al.*	[70]	A3	✓		✓			1st		2nd			D3							1
A-22	2014	Liu *et al.*	[14]	A3	✓		✓			1st	2nd				D3	✓			✓	✓	✓	1
A-23	2015	Piñol-Piñol *et al.*	[83]	A2	✓		✓			1st	2nd				D1	✓			✓		✓	1
A-30	2016	Liu *et al.*	[15]	A3	✓		✓			1st		2nd			D3	✓			✓	✓	✓	1
A-33	2016	Lavanya and Natarajan	[46]	A4						1st	2nd				D1							1
A-36	2016	Liu *et al.*	[71]	A3	✓		✓			1st	2nd				D1				✓	✓	✓	1
A-38	2017	Win *et al.*	[49]	A6	✓	✓	✓			1st		2nd			D1	✓			✓			1
A-43	2017	Khleborodov	[74]	A3			✓			1st	3rd			2nd	D1							1
A-49	2017	Hasan *et al.*	[58]	A4						1st		2nd			D1	✓				✓		1
A-50	2017	Sojka *et al.*	[59]	A3	✓	✓	✓			1st					D3	✓			✓			1
A-58	2018	Khleborodov	[76]	A3			✓			1st	3rd			2nd	D1							1
A-01	2007	Kim *et al.*	[36]	A5	✓	✓	✓				1st	3rd	2nd		D2				✓			3
A-02	2010	Kim *et al.*	[37]	A5	✓	✓				1st		2nd			D2							3
A-06	2012	Ju	[38]	A1	✓	✓				1st	3rd		2nd		D3							3
A-12	2013	Bakhache *et al.*	[39]	A1	✓	✓				2nd			1st		D7							3
A-16	2014	Druml *et al.*	[40]	A1	✓	✓	✓			1st	2nd				D2	✓			✓			3
A-20	2014	He *et al.*	[42]	A1	✓	✓				1st	3rd	2nd			D2							3
A-27	2016	Chaudhry *et al.*	[43]	A6	✓	✓				2nd		1st			D1				✓			3
A-31	2016	Reddy *et al.*	[44]	A1	✓		✓			2nd		1st			D7							3
A-32	2016	He *et al.*	[45]	A1	✓	✓				1st		2nd			D5	✓					✓	3
A-34	2016	Kaur *et al.*	[47]	A1	✓		✓			2nd		1st			D2	✓			✓			3
A-37	2017	Zhang *et al.*	[48]	A1	✓	✓				2nd		1st			D1							3
A-40	2017	Mathur *et al.*	[50]	A5	✓	✓	✓			2nd		1st			D1	✓						3
A-41	2017	Mahmood *et al.*	[51]	A1	✓	✓				1st		3rd	2nd		D5				✓			3
A-44	2017	Badra and Zeadally	[54]	A6	✓	✓				3rd		2nd	1st		D5				✓			3
A-46	2017	Lavanya and Natarajan	[55]	A5	✓	✓				2nd				1st	D1							3
A-47	2017	Meddah *et al.*	[56]	A6	✓	✓				1st					D4							3
A-48	2017	Mohammedi *et al.*	[57]	A1	✓	✓				2nd			1st		D4							3
A-51	2018	Shen *et al.*	[60]	A1	✓	✓				2nd		3rd		1st	D4							3
A-52	2018	Tewari *et al.*	[61]	A1	✓	✓					2nd	1st	3rd		D2					✓		3
A-56	2018	Shen *et al.*	[64]	A1	✓	✓				2nd	3rd	1st			D4	✓						3
A-57	2018	Mood and Nikooghadam	[65]	A1	✓	✓	✓			3rd		1st	2nd		D5	✓					✓	3
A-59	2018	Almulhim and Zaman	[66]	A1	✓	✓						1st		2nd	D4							3

of the most relevant words using the font size as differentiator. For the word cloud illustrated we extracted the titles of the surveyed papers, their references, and the papers that cite them, and used the tool WordCloud.[9]

C. TRENDS

By incorporating the publication and citation dates of the surveyed works to the reference analysis it is possible to observe trends in regards to the number of documents published, and the number of citations.

Considering the set of documents surveyed, their references, and their citations, Fig. 22 illustrates the number of papers per year of publication. A simple analysis of the data shows that the tendency is exponential, as given by

$$f(t) = 1.83 \times 10^{-110} e^{0.1278t}. \quad (33)$$

If we focus on the surveyed papers only we can identify as similar behavior, as illustrated in Fig. 23. With limited data it is difficult to determine whether the growth tendency can be kept or if the number of related publications will become stagnant over the time. For now it appears that there is a growing interest surrounding ECLC and related fields. This is expected to go hand in hand with the development of IoT and its evolving security requirements. This behavior follows the distribution

$$f(t) = 4.471 \times 10^{-248} e^{0.2837t}. \quad (34)$$

A similar behavior is observed in the distribution of the citations to the surveyed papers per year. This is illustrated

[9]https://pypi.python.org/pypi/wordcloud

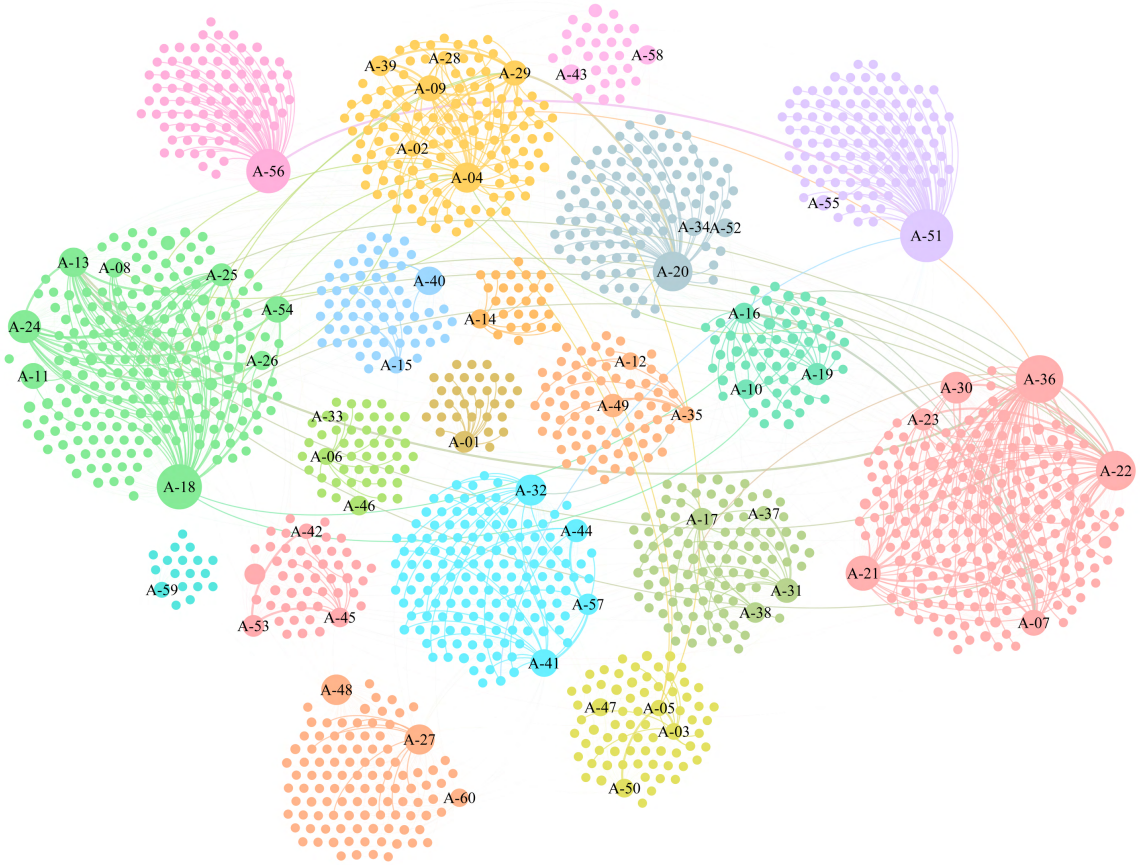

FIGURE 19. Analysis of the surveyed papers, their references, and citations as a graph. The varying colors represent different communities.

TABLE 9. Top six clusters identified in the set of documents surveyed.

ID	Year	Work	Ref.	A	B. Abstraction					C. Goals					D	E. Technology			F. Security			Cluster
					B1	B2	B3	B4	B5	C1	C2	C3	C4	C5		E1	E2	E3	F1	F2	F3	
A-07	2012	Wenger and Großschädl	[79]	A2			✓	✓		1st	3rd	2nd			D1			✓	✓			1
A-21	2014	Liu *et al.*	[70]	A3	✓		✓			1st		2nd			D3							1
A-22	2014	Liu *et al.*	[14]	A3	✓		✓			1st	2nd				D3	✓			✓	✓	✓	1
A-23	2015	Piñol-Piñol *et al.*	[83]	A2	✓		✓			1st	2nd				D1	✓			✓		✓	1
A-30	2016	Liu *et al.*	[15]	A3	✓		✓			1st		2nd			D3	✓			✓	✓	✓	1
A-36	2016	Liu *et al.*	[71]	A3	✓		✓			1st	2nd				D1				✓	✓	✓	1
A-08	2012	Trujillo-Olaya *et al.*	[4]	A3	✓		✓			2nd	1st				D2		✓				✓	2
A-11	2013	Wenger *et al.*	[91]	A2				✓			2nd	1st			D2			✓	✓			2
A-13	2013	Wenger	[80]	A2				✓	✓		2nd			1st	D3	✓		✓	✓			2
A-18	2014	Azarderakhsh *et al.*	[69]	A2			✓	✓	✓		1st				D1			✓	✓			2
A-24	2015	Roy *et al.*	[72]	A3	✓		✓	✓	✓	3rd	2nd	1st			D1			✓			✓	2
A-25	2015	Koziel *et al.*	[73]	A3	✓		✓	✓	✓	2nd	1st				D1			✓	✓	✓	✓	2
A-26	2016	Bosmans *et al.*	[92]	A2				✓	✓		2nd	1st			D1			✓			✓	2
A-54	2018	Järvinen	[75]	A3	✓	✓	✓	✓	✓	3rd	2nd	1st			D1			✓	✓	✓	✓	2
A-02	2010	Kim *et al.*	[37]	A5	✓	✓				1st		2nd			D2							3
A-04	2011	Varchola *et al.*	[84]	A2				✓		2nd	1st						✓			✓	✓	3
A-09	2012	Driessen *et al.*	[85]	A5	✓	✓		✓		2nd	1st			3rd	D6		✓		✓		✓	3
A-28	2016	Yalçin	[89]	A2				✓	✓		1st				D1		✓	✓	✓		✓	3
A-29	2016	Roy *et al.*	[88]	A2				✓			1st				D1		✓				✓	3
A-39	2017	Salman *et al.*	[90]	A2				✓		1st	2nd	3rd			D6		✓		✓	✓	✓	3
A-03	2010	Sojka *et al.*	[68]	A3	✓	✓	✓			1st					D3				✓			4
A-05	2011	Sojka *et al.*	[78]	A3	✓	✓	✓			1st					D3	✓			✓			4
A-47	2017	Meddah *et al.*	[56]	A6	✓	✓				1st					D4							4
A-50	2017	Sojka *et al.*	[59]	A3	✓	✓	✓			1st					D3	✓			✓			4
A-32	2016	He *et al.*	[45]	A1	✓	✓				1st		2nd			D5	✓					✓	14
A-41	2017	Mahmood *et al.*	[51]	A1	✓	✓				1st		3rd	2nd		D5				✓			14
A-44	2017	Badra and Zeadally	[54]	A6	✓	✓				3rd		2nd	1st		D5				✓			14
A-57	2018	Mood and Nikooghadam	[65]	A1	✓	✓	✓			3rd		1st	2nd		D5	✓					✓	14
A-17	2014	Yao *et al.*	[41]	A4						1st		3rd		2nd	D7							16
A-31	2016	Reddy *et al.*	[44]	A1	✓		✓			2nd		1st			D7							16
A-37	2017	Zhang *et al.*	[48]	A1	✓	✓				2nd		1st			D1							16
A-38	2017	Win *et al.*	[49]	A6	✓	✓	✓			1st		2nd			D1	✓			✓			16

in Fig. 24. This plot reveals that the interest in ECLC has sparkled in the last five years. The rapid growth can be associated to the consolidation of IoT and related technologies. The projection shown in Fig. 24 is represented by the model

$$f(t) = e^{0.6474t - 1301}. \quad (35)$$

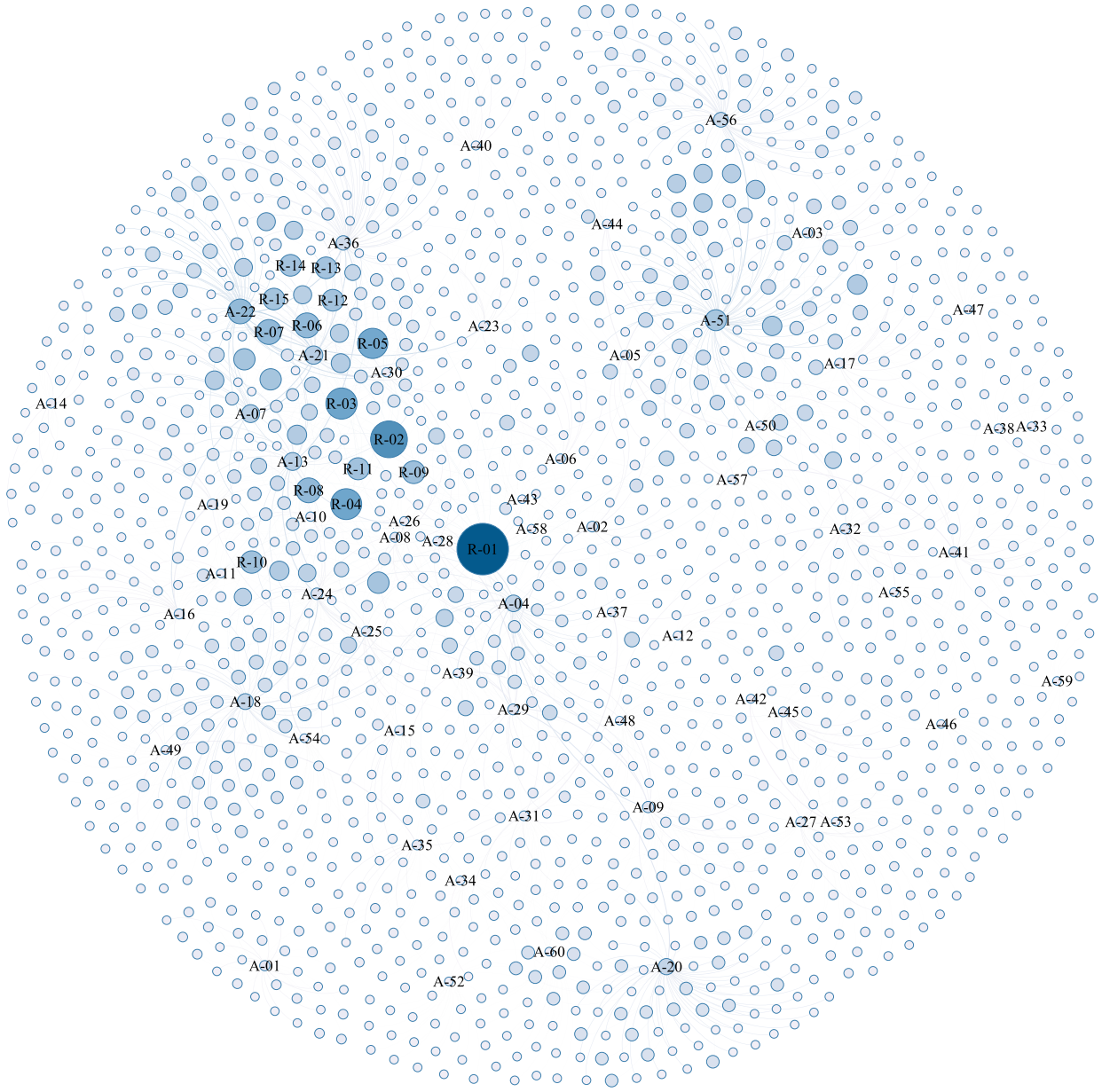

FIGURE 20. Analysis of the surveyed papers, their references, and citations as a graph. The size and color of the nodes is determined by their centrality.

If we study the citations to the surveyed papers in a case by case basis, it is possible to notice that there is no apparent relation in the number of citations with the year of publication. This analysis is presented in Fig. 25. A color code was included in the plot to showcase the number of references made by a paper. As it can be noted, it seems that there exists a relation between the number of publications references and the number of citations received.

VII. ECLC TAXONOMY AND APPLICATIONS

The advantages of ECLC over traditional asymmetric cryptography alternatives are clear: smaller key sizes, improved security, and flexibility, all inherited from ECC. When these are coupled with resource awareness and implementation efficiency, attractive solutions can be obtained. Being able to provide critical security services in constrained networked environments, without the drawbacks of conventional symmetric solutions, is the most important feature of ECLC.

The limitations steam from the novelty of the area. The main problems are the complexity in the design of ECLC solutions, the lack of interoperability, and the possibility of new vulnerabilities being discovered. All these can be addressed as long as there is a community interested on researching the field and new elliptic curves are proposed.

Fig. 26 illustrates ECLC as function of the different characteristics retrieved from the papers. In this diagram we have included the three main reasons for defining a proposal as lightweight, the five abstraction levels, and the most

TABLE 10. Documents with the higher Eigenvector Centrality (> 0.3) in the dataset of surveyed documents, their references, and the papers that cite them.

ID	Year	Work	Ref	Centrality
R-01	2004	Hankerson *et al.*	[12]	1.000000
R-02	2004	Gura *et al.*	[94]	0.654960
R-03	2008	Szczechowiak *et al.*	[95]	0.524719
R-04	1987	Montgomery	[96]	0.518480
R-05	2008	Liu and Ning	[97]	0.502248
R-06	2012	Großschädl *et al.*	[98]	0.379274
R-07	2006	Wang and Li	[99]	0.379274
R-08	1988	Itoh and Tsujii	[100]	0.377328
A-22	2014	Liu *et al.*	[14]	0.375931
R-09	2012	Cohen *et al.*	[10]	0.328163
R-10	2009	Hein *et al.*	[101]	0.321392
R-11	1985	Montgomery	[102]	0.312420
R-12	2003	Großschädl *et al.*	[103]	0.304679
R-13	2009	Lederer *et al.*	[11]	0.304679
R-14	2009	Ugus *et al.*	[104]	0.304679
R-15	2008	Hisil *et al.*	[105]	0.303745

important design goal for each work. In total 57% of the surveyed works are included in this representation.

From the surveyed data it can be appreciated how it is usually the case that ECLC solutions first try to meet the performance and security requirements of the application. Reducing resource and bandwidth usages are often secondary goals. From our experience, achieving high performance and security are critical for any security system. However, reducing resource usage (which impacts production costs) and energy consumption (which impacts devices lifetime) are key factors in fomenting the adoption of emerging technologies. Carefully constructed trade-offs between performance, security, hardware resources, and energy are undoubtedly important. In the next section we provide strategies for creating ECLC systems.

From Tables 2 through 6 we can identify some state of the art uses of ECLC. Providing confidentiality, integrity, authentication, and key establishment, through means of encryption, signatures, key agreement, and authentication protocols are common use cases. The technologies where these are found

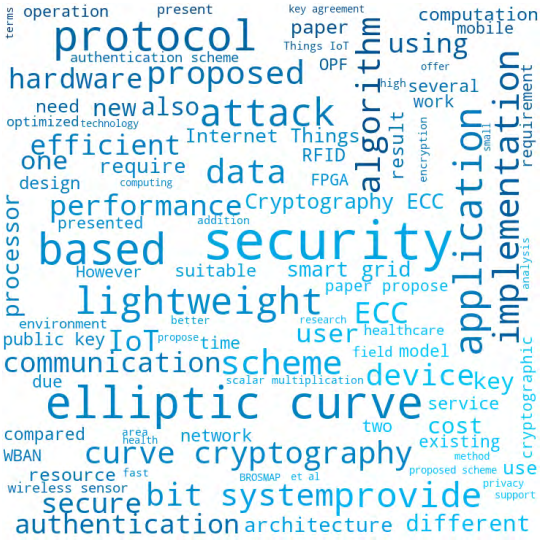

FIGURE 21. Word cloud generated with the abstracts of the surveyed papers. The size of each word is determined by its frequency in the source text.

include:

- IoT [43], [46], [48]–[50], [52], [55], [58]
- WSN [38], [48], [59]
- RFID [36], [37], [40], [42], [47], [61]
- *e*-Health [57], [60], [63], [66], [67]
- Smart Grid [45], [51], [54], [65]

These instances exemplify the potential for ECLC for several technologies which are perceived to become dominant in the near future [106]. Ensuring that our information is safe under these constrained applications is the purpose of ECLC.

VIII. STRATEGIES TO CONSTRUCT ECLC-BASED SOLUTIONS

We propose that the steps to follow for constructing an ECLC design and implementation process should match the abstraction levels presented in this survey. For the *System* level we have defined steps VIII-A to VIII-C. The *Protocol* level is reflected in step VIII-D. The steps VIII-E trough VIII-G are

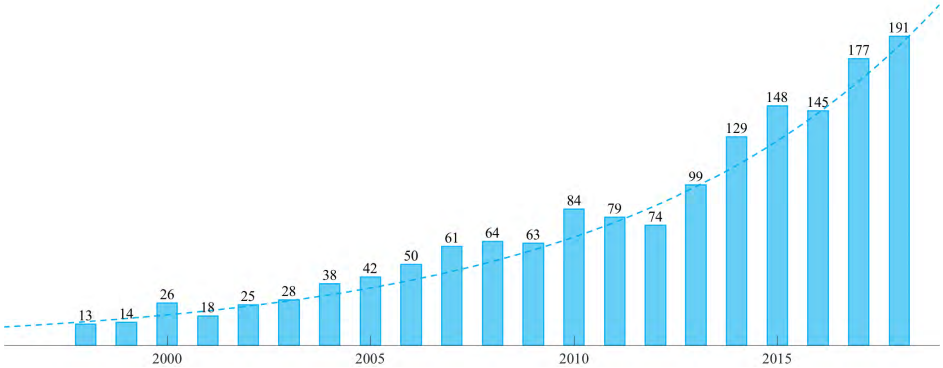

FIGURE 22. Publications per year in the last 20 years. Includes the papers surveyed, their references, and their citations.

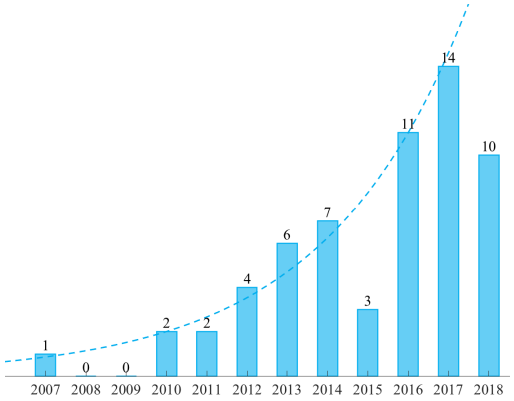

FIGURE 23. Publications per year of the surveyed papers.

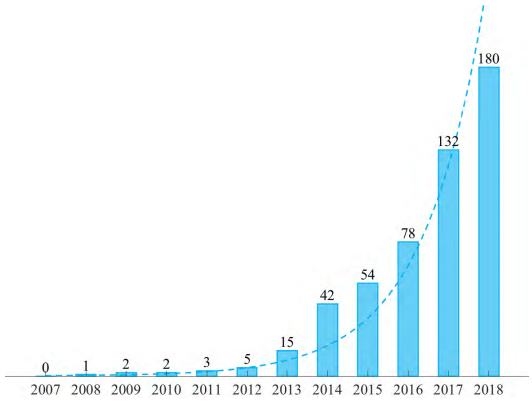

FIGURE 24. Citations per year to the surveyed papers.

associated with the *Algorithm* abstraction level. Finally the levels *Architecture* and *Circuit* are grouped in step VIII-H.

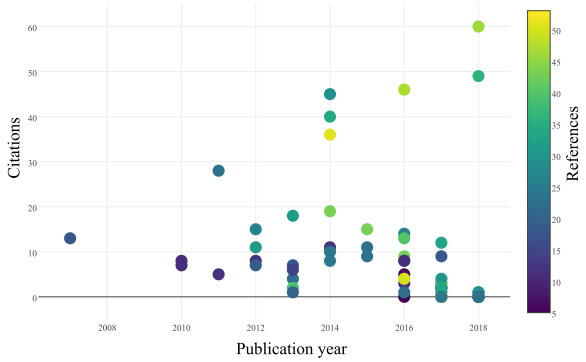

FIGURE 25. Citations to each surveyed paper. The color code indicates the number of references made by the paper.

A. IDENTIFY THE PROBLEM

The first step requires to clearly defining the problem and security requirements. Based on this definition a set of security services can be proposed to provide protection for the system. Some of the most popular services from asymmetric cryptography are authentication, key establishment, and signatures.

Some works which clearly define the problem are those that target specifically RFID [42], WSN [38], healthcare [57], among others. By knowing the problem, the required security services are defined.

B. DETERMINE THE IMPLEMENTATION PLATFORM

It is also important to identify the target implementation platform (the device) and identify the application constraints. It should first be defined if the implementation will be in hardware or in software. For the former, select the implementation technology: FPGA, ASIC or a different

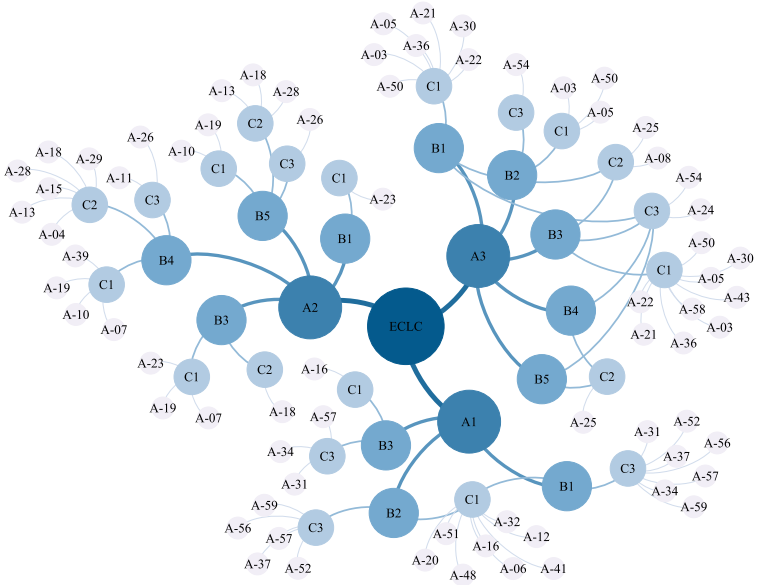

FIGURE 26. Taxonomy of ECLC. Only the most representative characteristics are included for visualization purposes.

system and study the hardware resources available in each case. For the latter, select a processor and study its architecture, its register width, the operations supported, and the memory availability.

From the literature review, some works that target WSN motes are [50] and [71]. These proposals have identified their implementation platform and designed their solution in such a way that it was useful for such platform. In the case of hardware, works like [88] and [89] have designed their solutions according to the features of the selected platform.

C. IDENTIFY THE SYSTEM CONSTRAINTS

Depending on the specifications and features of the devices some operational constraints can be drawn. Others come from standards and norms if the system is to be compliant with any regulations such as IEEE 802, ISO/IEC 14443, FIPS PUB 200, ISO/IEC 29182, among others. In general terms all limited devices share similar constraints, but the application may determine that some are more critical than others.

Take as example the proposals in Trujillo-Olaya *et al.* [4], Sojka-Piotrowska and Langendoerfer [59], and Sojka *et al.* [68], [78] take into account the application scope in order to define the security features of the solution. This idea should not be limited to the security aspects of the proposal, however, but also to the technical requirements of the system as in [40] and [50].

D. SELECT A PROTOCOL TO SOLVE THE PROBLEM

For most of the problems pertinent to networked environments general solutions are available in the literature. It is recommended, however, to construct an ad hoc protocol for the application scope. In this step the goal is to choose the protocol algorithm in general terms. This selection should account for the security services to be provided, which steam from the problem to be solved. Some of the most popular protocols include variants of ECDH, ECDSA, EC-ElGamal, ECIES, COAP and IKE.

E. SELECT THE DOMAIN PARAMETERS OF THE ELLIPTIC CURVE

Determine the ECC to be used with the protocol specified to provide security services for the system. Select the tuple of domain parameters $D = \{\mathbb{F}_q, E, P, n\}$ as specified in Section II-D.

The most common fields $\mathbb{F}_q$ to be utilized are the prime field or the binary field. The former being more efficient in software whereas the latter are preferred in hardware implementations. Current standards (NIST, IEEE, SEC, ...) recommend specific fields for security reasons. In the case of the prime field, it is defined by a prime p. In the case of the binary field, an irreducible polynomial $F(x)$ defines it. The length in bits of p or the degree of $F(x)$ must be compliant with the standards to meet a specific security level.

In the prime fields there are different elliptic curve families: random, Koblitz, Montgomery, Edwards, Twisted-Edwards, and most recently MoTE. For binary fields there are some other families: binary random, binary Koblitz, binary Edwards, and Hessian. Each one of these is a set of constructions denominated family, which contains curves for fields of different length. These were discussed in more detail in Section II-B.

The coefficients for each curve model are also defined for each instance. The generator or base point for an elliptic curve is provided with the specification, and some have multiple generators.

These values should be selected in a way that allows to achieve advantages in the implementation of the cryptosystem for the target platform.

The contribution in [71] provides a good example of a clear description of domain parameters, designed for constrained environments, which are said to come from a "family of lightweight elliptic curves" (the MoTE curves). We shall use the MoTE curve P159 as case study:

- Field. The selected field is the prime field $\mathbb{F}_p$ where

$$p = 2^{159} - 91. \tag{36}$$

  The length of p is 159 bits, therefore the expected security of this curve is 80-bit.
- Curve. The curve is defined through the Montgomery model

$$E_M : By^2 = x^3 + Ax^2 + x \tag{37}$$

  where

$$A = 3191566 \quad \text{and } B = -3191568. \tag{38}$$

  Since this is a MoTE curve, it is birationally equivalent to the Edwards curve given by

$$E_T : -x^2 + y^2 = 1 + dx^2y^2 \tag{39}$$

  where

$$d = 837225916393474870456 \\ /088348941705219765626634 92. \tag{40}$$

- Generator. Since the elliptic curve group is cyclic, any point in the curve can be a generator of the curve.
- Order. The order reported for the curve is $4l < p$ where l is a prime smaller than 2^{157}.
- Co-factor The curve has a co-factor of 4.

The first two parameters enumerated are mandatory in order to establish a system which is based on the curve specified. The generator is required in order to establish security protocols with additional parties, the difference with the first two parameters is that the generator can vary across instances of the security system. The last two parameters are related with the security of the system; these are not required for the system to work however it is important to report them for security auditing.

F. DEFINE THE POINT REPRESENTATION

As presented in Section II-B, points on an elliptic curve group $E(\mathbb{F}_q)$ can have multiple representations which have different characteristics. Derivations from the affine representation such as the projective coordinates, the w coordinates, the λ coordinates, and combinations of these can represent interesting alternatives for an elliptic curve system. For ECLC systems the mixed coordinate systems offer attractive advantages in reducing some of the operations required, such as inversions, and if paired with strategies like Co-Z, it is possible to reduce the storage requirements. Selecting different coordinate representations usually implies modifications to the group operations and a direct impact on the number and type of $\mathbb{F}_q$ operations.

The work in [73] proposes to use w coordinates to represent the elliptic curve points. In that work, w coordinates are first used to reduce the number of field operations required in the group operations. A mixed system with w and projective-w was implemented in order to reduce the number of field inversions required. The Co-Z strategy is also exploited in order to further reduce the operations count and the storage requirements of their solution.

G. CHOOSE THE ARITHMETIC

Once the elliptic curve to be used has been defined and the implementation technology is known, it is necessary to do the actual data processing. ECC relies on the group and field operations described in Section II-B. It is important to determine the most suitable alternatives to perform the group and field operations. This suitability is determined by the elliptic curve, the point representation, and finite field selected.

As mentioned, at group level the most important operation is the scalar multiplication. Some of the most popular alternatives to perform this calculation include the Double and Add method, Comb methods, NAF method, and the Montgomery Powering Ladder [12]. Whereas some of these focus on efficiency by performing the minimal number of operations required, others such as the Montgomery Ladder seek regularity in the processing of the data to prevent information leakage.

The scalar multiplication relies on point addition and point doubling. Through adequate selection of the point representation and for specific fields and curves, these operations can be optimized. Reducing the number of field operations, discarding complex field operations from the processing, reducing the diversity of field operations required, and mitigating duplicated calculations are all possible goals for these optimizations.

The field algorithms required by the group operations should be determined by the implementation system and the field selection. If the application constraints demand it, then performance should be prioritized in the field realizations used. For software systems, the availability depends on the instructions supported by the processor. In the case of hardware the limitations are physical since the resources tend to be restricted.

In some works like [86], the scalar multiplication would be performed using naïve approaches such as the Double and Add method. This procedure, however, has the disadvantage that it takes decisional branches which depend on the data utilized. This information leakage can be used to retrieve the secret key and compromise the system. More recent proposals, like [15], rely on the Montgomery ladder since it performs on constant time and mitigates the information leakage. This scalar multiplication method has also been used to construct more efficient group operations like the differential addition and doubling proposed in [73].

H. DEFINE THE IMPLEMENTATION STRATEGIES

By implementation we refer to taking the algorithmic description of the solution and mapping it to a physical realization. This can be achieved through the use of a processor, a reconfigurable device, or an integrated circuit. For each of these options the implementation strategies are different. Hence it is important to identify the underlying implementation technology.

According to the target platform each algorithm can be implemented in multiple ways. For constrained environments though, it is mandatory to address the restrictions of the system. Power availability, clocking frequency, maximum delay, and memory are generic constraints that can be observed for any technology. Specifically in processors, the stack size, the operations supported, the register width, and the RAM are related constraints. For hardware, physical size and generics availability are important factors. The software or hardware specification of the solution ought to be designed respecting these physical constraints.

Furthermore, the design goals can also influence the implementation. Once the ECLC design is suitable for the target platform, additional considerations can be taken into account in order to align the design with specific goals. A performance oriented implementation will be substantially different from an area optimized core, even if both perform the same task. The implementation strategies are determined first by the constraints of the system, and then by the design goals.

Papers like [15], [88], and [89] provide detailed explanations of their implementation strategies which may help the reader to further understand the design principles in ECLC.

I. SECTION REMARKS

The application of this procedure can ease the need to research the different steps that must be taken in the design of an ECLC solution. By using this method, a researcher would instead focus on a specific set of steps in order to achieve an ad hoc solution. Having broad knowledge of the problem, the target technology, and the system constraints can be considered as background data that should be specified when a project is started. Addressing the protocol, domain parameters, representation, and arithmetic can be considered sufficient to obtain a complete specification of the solution,

in a theoretical sense. The implementation strategies are what shall consolidate the idea.

IX. OPEN PROBLEMS AND FUTURE TRENDS

There are multiple challenges that must be overcome in order to make use of ECLC in constrained devices. First and foremost, even though the performance and implementation sizes of ECLC systems outperforms other asymmetric techniques, they still fall behind symmetric solutions. Lightweight symmetric ciphers and hash functions can achieve implementation sizes and latencies at least one order of magnitude smaller than those of ECLC. This restrains the use of ECLC systems from providing security services such as bulk encryption and authentication. Instead, ECLC solutions are pivotal in key establishment and the use of digital signatures. But these applications must observe the lengthy latencies and the hardware/processing overhead. In order to improve the quality of the services provided by ECLC, further research must be conducted to reduce the latency and implementation size of these solutions.

Another important challenge is associated with information security. As the development of new elliptic curves and processing techniques progresses, the attack models are also improved. Mathematical, cybernetic, and physical attacks are a threat to any security system. However, the challenge is greater if we consider the application scope of ECLC. Providing additional security measures generally represents significant security overheads. It is often the case that constrained environments cannot afford additional security protections. From the mathematical point of view, new algorithms can reduce the complexity of certain instances of ECDLP [107]. This leads to requiring increased key sizes or different families of curves. In that case the efficiency of the solution can be reduced. Cybernetic attacks are those that reach their targets over the internet. These protections are easier to account for as usually network-wide protections are put in place by a gateway. Finally, attacks which have direct access to the network are the most challenging to deal with. Constrained systems often are deployed in unsupervised environments and in high density. Multiple attacks can be performed with different goals and it is practically impossible to provide protections for every possible scenario. ECLC implementations must then consider that a device can be captured so it should not rely on nonvolatile information.

Standardization also hurts ECLC. Current standards are outdated in regards to the application scope. When the original suites were proposed, lightweight cryptography was not yet consolidated. Thus NIST and Sec standards only include general application curves. Novel efficient elliptic curves have been proposed but there is not a suite which includes them. This lack of standardization limits their usage. If multiple systems do not support these new curves, interoperability problems might arise. This problem also brings security risks. Since there is not any suite using modern elliptic curves, these will not get exposed to enough public scrutiny. More involvement of the community could help identifying risks and optimization opportunities.

In recent years, the idea that quantum computers will be a reality in the next decades has gained support. As NIST points out, many scientists now believe that the creation of practical quantum computers is merely a significant engineering challenge [108]. Some authors go as far as to state that within the next 20 years or so quantum computers, sufficiently large to present a threat to modern cryptography, will be built [109]. In 1994 Peter Shor discovered an algorithm capable of factoring numbers in polynomial time on a quantum computer, along with another to compute discrete logarithms. In a practical approach, this implies that traditional cryptographic primitives that rely on such problems can be broken by a quantum computer which is large enough. The possibility of the creation of a such quantum computer capable of running Shor's algorithm would represent the demise of any PKC system which relies on IF, DLP, and ECDLP as we know it.

In a Post-Quantum setting, Elliptic Curve based cryptography would be unable to rely on the hardness of ECDLP. However, the isogenies in super-singular curves [110] has been proposed as a different NP-hard problem which enables cryptographic constructions based on elliptic curves. The isogenies problem on super-singular curves has no quantum attack known, but there has not been enough analysis on their security [108]. This new variant of ECC would need to be adapted for constrained environments thus preserving the essence of ECLC.

X. CONCLUSIONS AND FINAL REMARKS

In this survey we have determined the criteria that make an ECC-based solution lightweight and viable for use in practical constrained applications. Representative works were systematically revised to determine the key aspects in an ECC design that lead to lightweight realizations. As a result, this paper provided for the first time the concept and requirements for Elliptic Curve Lightweight Cryptography (ECLC). We designed and described a methodology to create ECLC systems. We also discussed the open challenges that must be addressed by these systems. From the surveyed data we can answer the questions raised in the problem statement.

If there are proposals of elliptic curves designed to fit the needs of constrained devices, is it possible to call such curves as lightweight? Or is this adjective more associated with implementations of these systems?

In our study we found out that 48% of the surveyed papers were not implemented. This implies that it is possible to denominate a proposal as ''lightweight'' even when there is not implementation data to back this statement. Nonetheless, from our proposed methodology, we suggest that such works must observe the higher abstraction levels of *System*, *Protocol*, or *Algorithm*, in order to create a solution which can be useful for constrained environments.

If the elliptic curves are not standardized, will there be any traction on implementing them?

Multiple surveyed works were developed for nonstandardized elliptic curves. This leads us to believe that there is certain interest in the use of constructions which prove to be advantageous for the application. However, if modern, efficient curves were included in a standard for lightweight cryptography it would definitely help improving the auditing and optimization process for these systems. This would also contribute to spread their use.

What are the guidelines to determine if an ECC-based design or solution is lightweight?

The abstraction levels outlined in this survey can serve as a guideline for this purpose. This method relies on identifying if the solution was designed with modifications in regards to the implementation system, at the level of protocols, algorithms, architectures, or circuits, in aims to tailor it for constrained environments. It is desirable that these decisions can be backed up with implementation data, but as mentioned before, it is not a requirement. We found a recurring mistake found in the identification of why each one of the surveyed works is denominated lightweight. This was the misconception that replacing traditional PKC (RSA) with generic ECC can lead to lightweight implementations. It is indeed true that such works might be more suitable for constrained environments than the state of the art in PKC. However, true ECLC solutions are those that satisfy the definition provide in this work: first to select ECC and then to considerer several aspect for its efficient implementation in constrained devices.

What can be denominated ECLC? Is this concept utilized in the literature? Or is the word "lightweight" ever associated with ECC proposals/implementations?

We came to define ECLC as the set of elliptic curve protocols, domain parameters, algorithms, and implementation techniques, designed to provide security in constrained environments. All the surveyed works were denominated "lightweight" in the paper or related media. However, no one analyzes or supports the use of that adjective. We identified that the main difference with traditional lightweight cryptography is that ECLC first tries to address the performance constraints of the system to then pursue other objectives.

There are multiple threats that can compromise the security of ECLC systems. However, this technology offers great opportunities for the development of new systems such as IoT. The development of new networked environments will require strong security primitives which are efficient and represent small overheads for the device. This is a role that ECLC, with any of its different variants, can fulfill.

A. FUTURE WORK

Some directions that we would like to explore in the near future include: studying the implications of quantum attacks on the security of constrained devices; exploring the state of the art for other PKC solutions in the context of lightweight cryptography; construct benchmarks using commercial devices in order to compare the multiple solutions available under fair conditions.

REFERENCES

[1] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR 8114, Mar. 2017, doi: 10.6028/NIST.IR.8114.

[2] *Information Technology—Security Techniques—Lightweight Cryptography—Part 2: Block Ciphers*, document ISO/IEC 29192-2:2012, Jan. 2012.

[3] E. Barker, "Part 1. Revision 4. Recommendation for key management," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-57, 2016. Accessed: Oct. 10, 2018. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

[4] V. Trujillo-Olaya, T. Sherwood, and Ç. K. Koç, "Analysis of performance versus security in hardware realizations of small elliptic curves for lightweight applications," *J. Cryptograph. Eng.*, vol. 2, no. 3, pp. 179–188, 2012.

[5] B. Rashidi. (2017). "A survey on hardware implementations of elliptic curve cryptosystems." [Online]. Available: https://arxiv.org/pdf/1710.08336.pdf

[6] G. M. de Dormale and J.-J. Quisquater, "High-speed hardware implementations of elliptic curve cryptography: A survey," *J. Syst. Archit.*, vol. 53, nos. 2–3, pp. 72–84, 2007.

[7] S. Kalra and S. K. Sood, "Elliptic curve cryptography: Survey and its security applications," in *Proc. Int. Conf. Adv. Comput. Artif. Intell. (ACAI)*, New York, NY, USA, 2011, pp. 102–106.

[8] S. M. Sakharkar, R. S. Mangrulkar, and M. Atique, "A survey: A secure routing method for detecting false reports and gray-hole attacks along with Elliptic Curve Cryptography in wireless sensor networks," in *Proc. IEEE Students' Conf. Elect., Electron. Comput. Sci.*, Mar. 2014, pp. 1–5.

[9] R. Harkanson and Y. Kim, "Applications of elliptic curve cryptography: A light introduction to elliptic curves and a survey of their applications," in *Proc. 12th Annu. Conf. Cyber Inf. Secur. Res. (CISRC)*, 2017, pp. 6:1–6:7.

[10] H. Cohen *et al.*, Eds., *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, 2nd ed. London, U.K.: Chapman & Hall, 2012.

[11] C. Lederer, R. Mader, M. Koschuch, J. Großschädl, A. Szekely, and S. Tillich, "Energy-efficient implementation of ECDH key exchange for wireless sensor networks," in *Proc. 3rd IFIP WG 11.2 Int. Workshop Inf. Secur. Theory Pract. Smart Devices, Pervasive Syst., Ubiquitous Netw. (WISTP)*, Berlin, Germany: Springer-Verlag, 2009, pp. 112–127.

[12] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Secaucus, NJ, USA: Springer-Verlag, 2003.

[13] D. J. Bernstein, T. Lange, and R. R. Farashahi, "Binary edwards curves," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 5154. Berlin, Germany: Springer, 2008, pp. 244–265.

[14] Z. Liu, E. Wenger, and J. Großschädl, "MoTE-ECC: Energy-scalable elliptic curve cryptography for wireless sensor networks," in *Applied Cryptography and Network Security*. Cham, Switzerland: Springer, 2014, pp. 361–379.

[15] Z. Liu, J. Weng, Z. Hu, and H. Seo, "Efficient elliptic curve cryptography for embedded devices," *ACM Trans. Embedded Comput. Syst.*, vol. 16, pp. 53:1–53:18, Dec. 2016.

[16] H. C. A. van Tilborg and S. Jajodia, Eds., *Encyclopedia of Cryptography and Security*, 2nd ed. Boston, MA, USA: Springer, 2011.

[17] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[18] A. J. Menezes, S. A. Vanstone, and P. C. van Oorschot, *Handbook of Applied Cryptography*, 1st ed. Boca Raton, FL, USA: CRC Press, 1996.

[19] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[20] D. Beaver, S. Micali, and P. Rogaway, "The round complexity of secure protocols," in *Proc. 22nd Annu. ACM Symp. Theory Comput. (STOC)*, 1990, pp. 503–513.

[21] M. Bellare, V. T. Hoang, and P. Rogaway, "Foundations of garbled circuits," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2012, pp. 784–796.

[22] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Found. Secure Comput.*, vol. 4, no. 11, pp. 169–179, 1978.

[23] A. Lopez-Alt, E. Tromer, and V. Vaikuntanathan. (2013). *On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption*. [Online]. Available: https://eprint.iacr.org/2013/094

[24] J. Alperin-Sheriff and C. Peikert. (2014). *Faster Bootstrapping With Polynomial Error*. [Online]. Available: https://eprint.iacr.org/2014/094
[25] C. Peikert, *A Decade of Lattice Cryptography*. Boston, MA, USA: Now, 2016.
[26] C. Peikert, "Lattice cryptography for the Internet," in *Post-Quantum Cryptography*, M. Mosca, ed. Cham, Switzerland: Springer, 2014, pp. 197–219.
[27] R. Overbeck and N. Sendrier, "Code-based cryptography," in *Post-Quantum Cryptography*. Berlin, Germany: Springer, 2009, pp. 95–145.
[28] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer, 1985, pp. 47–53.
[29] S. Chatterjee and P. Sarkar, *Identity-Based Encryption*, 1st ed. Boston, MA, USA: Springer, 2011.
[30] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.
[31] X. Chen, K. Choi, and K. Chae, "A secure and efficient key authentication using bilinear pairing for NFC mobile payment service," *Wireless Pers. Commun.*, vol. 97, pp. 1–17, Nov. 2017.
[32] F. Guo, Y. Mu, W. Susilo, H. Hsing, D. S. Wong, and V. Varadharajan, "Optimized identity-based encryption from bilinear pairing for lightweight devices," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 2, pp. 211–220, Mar. 2017.
[33] K. T. Nguyen, N. Oualha, and M. Laurent, "Securely outsourcing the ciphertext-policy attribute-based encryption," *World Wide Web*, vol. 21, pp. 169–183, Jan. 2018.
[34] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 49, pp. 104–112, Aug. 2015.
[35] J. Fan, O. Reparaz, V. Rozic, and I. Verbauwhede, "Low-energy encryption for medical devices: Security adds an extra design dimension," in *Proc. 50th ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, May 2013, pp. 1–6.
[36] S. Kim, Y. Kim, and S. Park, "RFID security protocol by lightweight ECC algorithm," in *Proc. 6th Int. Conf. Adv. Lang. Process. Web Inf. Technol. (ALPIT)*, Aug. 2007, pp. 323–328.
[37] C.-J. Kim, S.-Y. Yun, and S.-C. Park, "A lightweight ECC algorithm for mobile RFID service," in *Proc. 5th Int. Conf. Ubiquitous Inf. Technol. Appl.*, Dec. 2010, pp. 1–6.
[38] S. Ju, "A lightweight key establishment in wireless sensor network based on elliptic curve cryptography," in *Proc. IEEE Int. Conf. Intell. Control, Autom. Detection High-End Equip.*, Jul. 2012, pp. 138–141.
[39] B. Bakhache, E. El-Hamawi, and H. Houssain, "Fast and secure key agreement protocol for the security of low power wireless networks," in *Proc. IEEE Faible Tension Faible Consommation (FTFC)*, Jun. 2013, pp. 1–4.
[40] N. Druml *et al.*, "A Flexible and Lightweight ECC-Based Authentication Solution for Resource Constrained Systems," in *Proc. 17th Euromicro Conf. Digit. Syst. Design (DSD)*, Aug. 2014, pp. 372–378.
[41] X. Yao, X. Han, and X. Du, "A light-weight certificate-less public key cryptography scheme based on ECC," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2014, pp. 1–8.
[42] D. He, N. Kumar, N. Chilamkurti, and J.-H. Lee, "Lightweight ECC based RFID authentication integrated with an ID verifier transfer protocol," *J. Med. Syst.*, vol. 38, no. 10, p. 116, 2014.
[43] S. A. Chaudhry, H. Naqvi, K. Mahmood, H. F. Ahmad, and M. K. Khan, "An improved remote user authentication scheme using elliptic curve cryptography," *Wireless Pers. Commun.*, vol. 96, pp. 5355–5373, Oct. 2017.
[44] A. G. Reddy, E.-J. Yoon, A. K. Das, and K.-Y. Yoo, "Lightweight authentication with key-agreement protocol for mobile network environment using smart cards," *IET Inf. Secur.*, vol. 10, no. 5, pp. 272–282, Mar. 2016.
[45] D. He, H. Wang, M. K. Khan, and L. Wang, "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography," *IET Commun.*, vol. 10, no. 14, pp. 1795–1802, 2016.
[46] M. Lavanya and V. Natarajan, "Lightweight authentication for COAP based IOT," in *Proc. 6th Int. Conf. Internet Things (IoT)*, New York, NY, USA, 2016, pp. 167–168.
[47] K. Kaur, N. Kumar, M. Singh, and M. S. Obaidat, "Lightweight authentication protocol for RFID-enabled systems based on ECC," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[48] W. Zhang, D. Lin, H. Zhang, C. Chen, and X. Zhou, "A lightweight anonymous mutual authentication with key agreement protocol on ECC," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 170–176.
[49] E. K. Win, T. Yoshihisa, Y. Ishi, T. Kawakami, Y. Teranishi, and S. Shimojo, "A lightweight multi-receiver encryption scheme with mutual authentication," in *Proc. IEEE 41st Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jul. 2017, pp. 491–497.
[50] A. Mathur, T. Newe, W. Elgenaidi, M. Rao, G. Dooly, and D. Toal, "A secure end-to-end IoT solution," *Sens. Actuators A, Phys.*, vol. 263, pp. 291–299, Aug. 2017.
[51] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Gener. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2017.
[52] A. A. Diro, N. Chilamkurti, and P. Veeraraghavan, "Elliptic curve based cybersecurity schemes for publish-subscribe Internet of Things," in *Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Cham, Switzerland: Springer, 2017, pp. 258–268.
[53] A. A. Diro, N. Chilamkurti, and N. Kumar, "Lightweight cybersecurity schemes using elliptic curve cryptography in publish-subscribe fog computing," *Mobile Netw. Appl.*, vol. 22, pp. 848–858, Oct. 2017.
[54] M. Badra and S. Zeadally, "Lightweight and efficient privacy-preserving data aggregation approach for the Smart Grid," *Ad Hoc Netw.*, vol. 64, pp. 32–40, Sep. 2017.
[55] M. Lavanya and V. Natarajan, "Lightweight key agreement protocol for IoT based on IKEv2," *Comput. Elect. Eng.*, vol. 64, pp. 580–594, Nov. 2017.
[56] N. Meddah, A. Jebrane, and A. Toumanari, "Scalable lightweight ABAC scheme for secure sharing PHR in cloud computing," in *Advanced Information Technology, Services and Systems*. Cham, Switzerland: Springer, 2018, pp. 333–346.
[57] M. Mohammedi, M. Omar, and A. Bouabdallah, "Secure and lightweight remote patient authentication scheme with biometric inputs for mobile healthcare environments," *J. Ambient Intell. Humanized Comput.*, vol. 9, pp. 1527–1539, Sep. 2017.
[58] H. Hasan *et al.*, "Secure lightweight ECC-based protocol for multi-agent IoT systems," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2017, pp. 1–8.
[59] A. Sojka-Piotrowska and P. Langendoerfer, "Shortening the security parameters in lightweight WSN applications for IoT—Lessons learned," in *Proc. 2nd IEEE PERCOM Workshop Secur. Privacy Trust Internet Things*, Mar. 2017, pp. 636–641.
[60] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Gener. Comput. Syst.*, vol. 78, no. 3, pp. 956–963, 2018.
[61] A. Tewari and B. B. Gupta, "A robust anonymity preserving authentication protocol for IoT devices," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–5.
[62] A. A. Diro, N. Chilamkurti, and Y. Nam, "Analysis of lightweight encryption scheme for fog-to-things communication," *IEEE Access*, vol. 6, pp. 26820–26830, 2018.
[63] A. Vaniprabha and P. Poongodi, "Augmented lightweight security scheme with access control model for wireless medical sensor networks," *Cluster Comput.*, vol. 21, pp. 1–11, Jan. 2018.
[64] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *J. Netw. Comput. Appl.*, vol. 106, pp. 117–123, Mar. 2018.
[65] D. Abbasinezhad-Mood and M. Nikooghadam, "Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications," *Future Gener. Comput. Syst.*, vol. 84, pp. 47–57, Jul. 2018.
[66] M. Almulhim and N. Zaman, "Proposing secure and lightweight authentication scheme for IoT based E-health applications," in *Proc. 20th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2018, pp. 481–487.
[67] M. Mohammedi, M. Omar, W. Aitabdelmalek, A. Mansouri, and A. Bouabdallah, "Secure and lightweight biometric-based remote patient authentication scheme for home healthcare systems," in *Proc. Int. Symp. Program. Syst. (ISPS)*, Apr. 2018, pp. 1–6, doi: 10.1109/ISPS.2018.8379017.
[68] A. Sojka, K. Piotrowski, and P. Langendoerfer, "Short ECC: A lightweight security approach for Wireless Sensor Networks," in *Proc. Int. Conf. Secur. Cryptogr. (SECRYPT)*, Jul. 2010, pp. 1–5.

[69] R. Azarderakhsh, K. U. Jarvinen, and M. Mozaffari-Kermani, "Efficient algorithm and architecture for elliptic curve cryptography for extremely constrained secure applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1144–1155, Apr. 2014.

[70] Z. Liu, J. Großschädl, and D. S. Wong, "Low-weight primes for lightweight elliptic curve cryptography on 8-bit AVR processors," in *Information Security and Cryptology*. Cham, Switzerland: Springer, 2014, pp. 217–235.

[71] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure Internet of Things: ECC comes of age," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 3, pp. 237–248, May 2017.

[72] S. S. Roy, K. Järvinen, and I. Verbauwhede, "Lightweight coprocessor for Koblitz curves: 283-bit ECC including scalar conversion with only 4300 gates," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2015, pp. 102–122.

[73] B. Koziel, R. Azarderakhsh, and M. Mozaffari-Kermani, "Low-resource and fast binary edwards curves cryptography," in *Progress in Cryptology*. Cham, Switzerland: Springer, 2015, pp. 347–369.

[74] D. Khleborodov, "Fast elliptic curve point multiplication based on binary and binary non-adjacent scalar form methods," *Adv. Comput. Math.*, vol. 44, pp. 1275–1293, Aug. 2018.

[75] K. Jarvinen, S. S. Roy, and I. Verbauwhede, "Arithmetic of τ-adic expansions for lightweight koblitz curve cryptography," *J. Cryptograph. Eng.*, vol. 8, no. 4, pp. 285–300, Nov. 2018.

[76] D. Khleborodov, "Fast elliptic curve point multiplication based on window Non-Adjacent Form method," *Appl. Math. Comput.*, vol. 334, pp. 41–59, Oct. 2018.

[77] N. Meloni, "New point addition formulae for ECC applications," in *Arithmetic of Finite Fields*, C. Carlet and B. Sunar, Eds. Berlin, Germany: Springer, 2007, pp. 189–201.

[78] A. Sojka, K. Piotrowski, and P. Langendoerfer, "Symbiosis of a lightweight ecc security and distributed shared memory middleware in wireless sensor networks," in *Proc. IEEE 30th Symp. Reliable Distrib. Syst. Workshops*, Oct. 2011, pp. 36–41.

[79] E. Wenger and J. Grossschadl, "An 8-bit AVR-based elliptic curve cryptographic RISC processor for the Internet of Things," in *Proc. 45th Annu. IEEE/ACM Int. Symp. Microarchitecture Workshops (MICROW)*, Dec. 2012, pp. 39–46.

[80] E. Wenger, "Hardware architectures for MSP430-based wireless sensor nodes performing elliptic curve cryptography," in *Applied Cryptography and Network Security*, M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, Eds. Berlin, Germany: Springer, 2013, pp. 290–306.

[81] S. Namal, K. Georgantas, and A. Gurtov, "Lightweight authentication and key management on 802.11 with Elliptic Curve Cryptography," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 1830–1835.

[82] A. Höller, N. Druml, C. Kreiner, C. Steger, and T. Felicijan, "Hardware/software co-design of elliptic-curve cryptography for resource-constrained applications," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[83] O. P. Piñol, S. Raza, J. Eriksson, and T. Voigt, "BSD-based elliptic curve cryptography for the open Internet of Things," in *Proc. 7th Int. Conf. New Technol., Mobility Secur. (NTMS)*, Jul. 2015, pp. 1–5.

[84] M. Varchola, T. Guneysu, and O. Mischke, "MicroECC: A lightweight reconfigurable elliptic curve crypto-processor," in *Proc. Int. Conf. Reconfigurable Comput. (FPGAs)*, Nov. 2011, pp. 204–210.

[85] B. Driessen, T. Güneysu, E. B. Kavun, O. Mischke, C. Paar, and T. Pöppelmann, "IPSecco: A lightweight and reconfigurable IPSec core," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig)*, Dec. 2012, pp. 1–7.

[86] M. Schramm and A. Grzemba, "On the implementation of a lightweight generic FPGA ECC crypto-core over GF(p)," in *Proc. Int. Conf. Appl. Electron.*, Sep. 2013, pp. 1–4.

[87] E. Wenger, "A lightweight ATmega-based application-specific instruction-set processor for elliptic curve cryptography," in *Lightweight Cryptography for Security and Privacy*. Berlin, Germany: Springer, 2013, pp. 1–15.

[88] D. B. Roy, P. Das, and D. Mukhopadhyay, "ECC on your fingertips: A single instruction approach for lightweight ECC design in GF(p)," in *Selected Areas in Cryptography*. Cham, Switzerland: Springer, 2016, pp. 161–177.

[89] T. Yalçin, "Compact ECDSA engine for IoT applications," *Electron. Lett.*, vol. 52, no. 15, pp. 1310–1312, 2016.

[90] A. Salman, A. Ferozpuri, E. Homsirikamol, P. Yalla, J. Kaps, and K. Gaj, "A scalable ECC processor implementation for high-speed and lightweight with side-channel countermeasures," in *Proc. Int. Conf. ReConFigurable Comput. FPGAs (ReConFig)*, Dec. 2017, pp. 1–8.

[91] E. Wenger, T. Korak, and M. Kirschbaum, "Analyzing side-channel leakage of RFID-suitable lightweight ECC hardware," in *Radio Frequency Identification*. Berlin, Germany: Springer, 2013, pp. 128–144.

[92] J. Bosmans, S. S. Roy, K. Jarvinen, and I. Verbauwhede, "A tiny coprocessor for elliptic curve cryptography over the 256-bit NIST prime field," in *Proc. 29th Int. Conf. VLSI Design 15th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2016, pp. 523–528.

[93] T. K. Goyal and V. Sahula, "Lightweight security algorithm for low power IoT devices," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2016, pp. 1725–1729.

[94] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2004, pp. 119–132.

[95] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, "NanoECC: Testing the limits of elliptic curve cryptography in sensor networks," in *Wireless Sensor Networks*. Berlin, Germany: Springer, 2008, pp. 305–320.

[96] P. L. Montgomery, "Speeding the Pollard and elliptic curve methods of factorization," *Math. Comput.*, vol. 48, pp. 243–264, Jan. 1987.

[97] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. 7th Int. Conf. Inf. Process. Sensor Netw.*, Washington, DC, USA, 2008, pp. 245–256.

[98] J. Großschädl, M. Hudler, M. Koschuch, M. Krüger, and A. Szekely, "Smart elliptic curve cryptography for smart dust," in *Quality, Reliability, Security and Robustness in Heterogeneous Networks*. Berlin, Germany: Springer, 2012, pp. 623–634.

[99] H. Wang and Q. Li, "Efficient implementation of public key cryptosystems on mote sensors (short paper)," in *Information and Communications Security*, P. Ning, S. Qing, and N. Li, Eds. Berlin, Germany: Springer, 2006, pp. 519–528.

[100] T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverses in GF (2^m) using normal bases," *Inf. Comput.*, vol. 78, pp. 171–177, Sep. 1988.

[101] D. Hein, J. Wolkerstorfer, and N. Felber, "ECC is ready for RFID—A proof in silicon," in *Selected Areas in Cryptography*, R. M. Avanzi, L. Keliher, and F. Sica, Eds. Berlin, Germany: Springer, 2009, pp. 401–413.

[102] P. L. Montgomery, "Modular multiplication without trial division," *Math. Comput.*, vol. 44, no. 170, pp. 519–521, Apr. 1985.

[103] J. Großschädl and G.-A. Kamendje, "Architectural enhancements for Montgomery multiplication on embedded RISC processors," in *Applied Cryptography and Network Security*, J. Zhou, M. Yung, and Y. Han, Eds. Berlin, Germany: Springer, 2003, pp. 418–434.

[104] O. Ugus, D. Westhoff, R. Laue, A. Shoufan, and S. A. Huss. (2009). "Optimized implementation of elliptic curve based additive homomorphic encryption for wireless sensor networks." [Online]. Available: https://arxiv.org/abs/0903.3900

[105] H. Hisil, K. K.-H. Wong, G. Carter, and E. Dawson, "Twisted Edwards curves revisited," in *Advances in Cryptology—ASIACRYPT*, J. Pieprzyk, ed. Berlin, Germany: Springer, 2008, pp. 326–343.

[106] Sri International Menlo Park CA. (2008). *Six Technologies with Potential Impacts on US Interests out to 2025*. Accessed: Oct. 16, 2018. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a519715.pdf

[107] T. Kim and R. Barbulescu, "Extended tower number field sieve: A new complexity for the medium prime case," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 2016, pp. 543–571.

[108] L. Chen *et al. Report on Post-Quantum Cryptography*. Accessed: Dec. 10, 2018. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf

[109] M. Mosca. *Cybersecurity in an Era With Quantum Computers: Will we be Ready?* Accessed: Oct. 10, 2018. [Online]. Available: http://eprint.iacr.org/2015/1075.pdf

[110] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *Post-Quantum Cryptography*. Berlin, Germany: Springer, 2011, pp. 19–34s.

CARLOS ANDRES LARA-NINO received the master's degree in computer science from CINVESTAV Tamaulipas, Mexico, in 2016, where he is currently pursuing the Ph.D. degree. His academic interests include digital systems, robotics, FPGAs, and security. His current research focuses on the implementation of cryptographic algorithms optimized for constrained environments.

ARTURO DIAZ-PEREZ received the Ph.D. degree in electrical engineering from CINVESTAV, Mexico, in 1998. He is currently a full-time Professor with CINVESTAV Guadalajara. He has co-authored the book *Cryptographic Algorithms on Reconfigurable Hardware*. His research interests include information security and algorithms for high-performance computing, hardware security in constrained devices, and security schemes for big-data storage and processing.

MIGUEL MORALES-SANDOVAL received the Ph.D. degree from the National Institute for Astrophysics, Optics, and Electronics, Mexico, in 2008. He is currently a researcher in computer science with special interests on data security, cryptography, and embedded systems. He is currently focused on the development of hardware/software security schemes for networked embedded systems and for the cloud.