# Software application based on ECC to compute parameters of security association protocols specified in IEEE 802.15.6 Standard

Gustavo Dávila*, Oscar Torres*, Jorge Eduardo Rivadeneira† and Pablo Hidalgo*
*Departamento de Electrónica Telecomunicaciones y Redes de Información,
Escuela Politécnica Nacional, Ladrón de Guevara, E11-253, Quito-Ecuador
Email: {gustavo.davila, oscar.torres01}@epn.edu.ec, phidalgo@ieee.org
†School of Electronics and Computer Science
University of Southampton, University Road, SO17 1BJ, Southampton, United Kingdom
Email: jerm1n14@southamptonalumni.ac.uk

***Abstract*—Wireless Body Area Network (WBAN) is a network of nodes which has emerged as a technology for medical and a non-medical application operating in the vicinity of, or inside a human body. The security services are necessary for preserving the confidentiality and integrity of sensitive information from users of WBAN. Standard 802.15.6 provides several association protocols within security services. In order to generate the parameters necessaries for setting the association protocols, this paper developed a software application that computes a $P_k$ based on elliptic curve method. The Elliptic Curve Cryptography (ECC) has been analysed in a mathematic form to proceed to the development of algorithms inside the software. The parameters defined in Std. 802.15.6 are probed in this software with the aim of having a tool to be used in every association protocol.**

***Index Terms*—Association Protocols, ECC, IEEE 802.15.6, Public Key, WBAN**

## I. Introduction

A new generation of nodes-based networks has been developed in IEEE 802.15.6 Standard. The main characteristic of this Standard has been the study of wireless communication for the network of nodes in the vicinity of, or inside a human body [1], [2], this networks are called Wireless Body Area Networks. WBANs have important implications in medical and sports applications.

Nowadays, there are no devices or nodes with the ability to perform medical measurements and communicate with the hub through the human body. The purpose of IEEE 802.15.6 is to provide the parameters of frameworks elements, MAC frame format, MAC functions, security services and communication bands for a short-range, low power and highly reliable wireless communications between devices in, on, or around the human body [1], [2], [3].

The communication between a node and hub in WBANs, IEEE 802.15.6 has defined three different security levels which are: unsecured, authentication unencrypted and authentication encrypted; the last two levels are used in secure communications. In a secure communication, the process that the nodes follow is ascertainment of authentication credentials, generation or activation of a Master Key(MK), Pairwise Temporal Key(PTK) creation, message security. This paper will be focused on the master key generation process. For establishing a new MK, the node and hub need to choose an authenticated or unauthenticated association process.

The Standard IEEE 802.15.6 has defined four security association protocols, in order to share the MK. The association protocols defined on the Standard are Unauthenticated Association, Public Key hidden Association, Password Authenticated Association and Display Authenticated Association[3], [4].This paper belongs to a bigger investigation which is focused in the Analysis of Vulnerabilities in Display Authenticated Association being necessary the process development of Association as one of its most important approaches, the implementation of generation of private key also called secret key ($S_k$) from public key ($P_k$) using Elliptic-curve cryptography.The association processes defined in the security services of IEEE 802.15.6 are based on Diffie-Hellman key exchange [5].

The $S_k$ will be chosen by the node according certain parameters, while the $P_k$ must be generated from the $S_k$ [2]. There are some methods for the generation of public keys based on private keys, however, IEEE 802.15.6 Standard allows only one, which is through an elliptic curve computation. Within the association protocols, there are some other algorithms to complete the link between the node and the hub, but the generation of the $P_k$ is one of the most important since without this calculation the following steps are non-viable. Display Authenticated Association requires a $P_k$ calculation based on the elliptic curve method for the subsequent generation of the MK. Additionally the algorithm described in this paper will be used in the calculation Diffie-Hellman Key used in Display Authenticated Association authentication process [1], [2].

This paper faces the need to have a computational tool that allows the parameters calculation like the $P_k$ for the IEEE 802.15.6 association protocols. Section II describes a mathematical analysis of the functions and operations to generate the $P_k$, also includes the parameters described in the IEEE 802.15.6 and an example process will be performed for simple curves with the objective of verifying the correct operation of the application. Section III lists the algorithms,

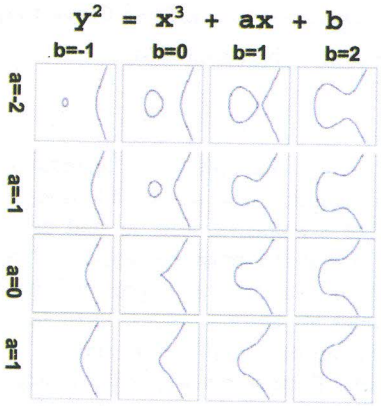


Fig. 1. Coefficients a, b and their respective graphs [7]

classes and methods for the design and implementation of $P_k$ generator software based on the elliptic curve method. The application assessment will be made in Section IV and the results obtained will be shown for both the curves entered manually and for the curve defined in the IEEE 802.15.6 Standard. Section V lists some conclusions and future works.

## II. Elliptic Curve

### A. Background

Elliptic curves (EC) were developed as algebraic functions, they have been studied for more than a century. The first time that elliptic curves were used in cryptography was in 1985. The first model of cryptography using elliptical curve was proposed by Neal Koblitz and Victor Miller [6], [7]. After studies and tests, the elliptic curve was standardised by multiple organisations around 1990, with its subsequent commercial appearance. The elliptic curve strength lies on the discrete logarithm problem ECDLP [7], [8], [9].

### B. Elliptic Curve Definition

The main characteristic of the family of elliptic curves is that they are non-singular curves [7], [9]. The elliptic curves geometrically do not have auto intersections. The set of points as solutions to the elliptic curve form an Abelian group [7], [9]. A group is Abelian if it fulfils the commutative property. The elliptic curve is defined on the basis of (1).

$$y^2 = x^3 + ax + b \tag{1}$$

The values of $a$ and $b$ will form the elliptic curve. Some possible values of $a$ and $b$ are described in Figure 1.

For cryptographic systems, it is not possible to use curves with double roots. A valid EC should ensure that the equation contains three different solutions [7], [9]. This condition is controlled by the discriminant condition shown in (2).

$$4a^3 + 27b^2 \neq 0 \tag{2}$$

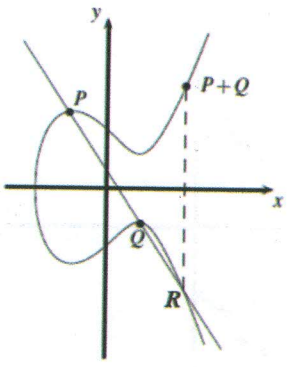


Fig. 2. Addition of two points in EC [7]

### C. Elliptic Curve Operations

There are two valid operations on elliptic curves: the addition of a point $P$ and a point $Q$, and the double of a point $P$.

Adding two points, where $P$ other than $-Q$ can be analysed geometrically and mathematically. Geometrically, the addition of two points is achieved by drawing a line through the points $P$ and $Q$. Due to the characteristics of the elliptic curve, this line drawn through $P$ and $Q$ must intersect at a point $R$ (also solution to the elliptic curve). The solution of the sum of points is the reflection on the x-axis of the point $R$. An example of addition of two points is given in Figure 2.

The mathematical analysis for the addition of two points is defined in (3), (4), (5).

$$x_R = \lambda^2 - x_1 - x_2 \tag{3}$$

$$y_R = -y_1 + \lambda(x_1 - x_R) \tag{4}$$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \tag{5}$$

Where $x_1$, $y_1$, $x_2$, $y_2$ are Cartesian coordinates of points $P$ and $Q$ respectively, and $\lambda$ is the slope of the line.

The second defined operation is the double of a point. Geometrically, doubling a point differs from the analysis done in the addition of two points because this involves drawing a tangent from the point to bend with the objective that intersects in a new solution point to the elliptical curve.The double point will be the reflection on the $x$-axis of that intersection [6]. An example of doubling the point process is given in Figure 3.

In order to find the double of a point analytically, we apply (6), (7), (8).

$$x_R = \lambda^2 - 2x_1 \tag{6}$$

$$y_R = -y_1 + \lambda(x_1 - x_R) \tag{7}$$

$$\lambda = \frac{3x_1^2 + a}{2y_1} \tag{8}$$

There is a point that satisfies (9) and (10).

$$P + O = P \tag{9}$$

$$2P = O \tag{10}$$

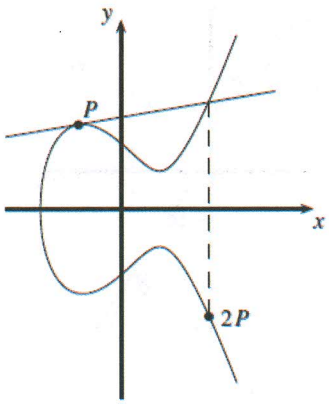


Fig. 3. Double of a point in EC [7]

The point $O$ is considered as a neutral element defined in an abstract way which can only be found in the infinity [7], [9]. If someone wishes to add a point with its reflection on the $x$-axis the resulting line would give solution or intersection only at infinity. In the same way, if we want to bend a point with coordinate $y = 0$, the tangent would be parallel to the $y$-axis, having a solution only at $O$.

### *D. Elliptic Curve Over Finite Fields*

In [7], [8] a prime number $p$ is considered, such that $p > 3$, the elliptic curve is defined on the field $Z_p$ in (11). The finite set $Z_p$ is defined in (12).

$$y^2 = x^3 + ax + b \pmod{p} \tag{11}$$

$$Z_p = 0, 1, 2, ..., p - 1 \tag{12}$$

An elliptic curve on a finite field as in the elliptic curve on the real numbers must satisfy the discriminant defined in (13).

$$4a^3 + 27b^2 \neq 0 \pmod{p} \tag{13}$$

Within an EC on $Z_p$, the solution elements also form a finite set. For a value in the $y$-coordinate to be a valid solution, it must be a quadratic residue (QR). A number is said to be a QR if and only if it has a square root within the field $Z_p$ [6]. Within a field $Z_p$, the number of quadratic residues is given by (14).

$$n = \frac{p-1}{2} \tag{14}$$

Where $n$ is the amount of QR that has the finite set $Z_p$. The number of quadratic residues does not determine which values are QR. A number is a quadratic residue if it complies with (15).

$$x^{\frac{p-1}{2}} = 1 \tag{15}$$

This implies that not all values within the set $Z_p$ can be solutions $(x, y)$ of the EC.

### *E. Public Key Generation with Elliptic Curve Cryptography*

The $P_k$ generation is based on (16).

$$S_k \times G = P_k \tag{16}$$

As mentioned before, the only valid operations for points belonging to the elliptic curve are the addition of two points and the double of a point. In order to find the $P_k$, a series of consecutive operations must be performed to reach the relationship in (16). It should be emphasized that the defined (3) - (8), are valid for elliptic curves on finite fields on the basis of modular arithmetic [9], [10] and on a $Z_p$. Within modular arithmetic there is not a division definition [9],[10], nevertheless, there is an operation of multiplying one value by the inverse of the other. The inverse of a number within finite set $p$ is defined on the basis of (17), valid as long as $p$ is a prime value greater than three.

$$\frac{1}{x} = x^{p-2} \quad over \ Z_p \tag{17}$$

The following section presents an example of how $P_k$, under the elliptic curve method, is obtained from a $S_k = 7$. The operations to arrive at this equivalent could be: $G, 2G, 4G, 5G, 6G, 7G$. or $G, 2G, 3G, 6G, 7G$. In this second option can be seen that the number of iterations to reach the relation, or in our case the $S_k$, was smaller. The number of iterations will have a direct impact on the processing cost and resources required for the generation of public keys. Any method used to arrive at that relationship gives exactly the same result and therefore the same $P_k$.

The solutions to the elliptic curve on a finite field $Z_p$ form an Abelian group. This implies that any operation performed on a defined point of the EC results in another point on the elliptic curve including the point of infinity.

The security provided by the elliptical curve is due to the fact that despite the knowledge of the $P_k$ and the starting point of the elliptical curve, finding $S_k$ is very complex. With an initial point of the elliptical curve, the possible ways to reach the $P_k$ are innumerable, this along with the large number of combinations provided by the number of bits of a $S_k$, offers a high level of security. For IEEE 802.15.6 Standard the number of bits of the $S_k$ are 256. The computation of the $P_k$ from a $S_k$ is relatively simple with the respective inverse process, having a system practically unbreakable.

### *F. Coefficients and Domain parameters for IEEE 802.15.6*

The parameters used in IEEE 802.15.6 Standard are specified for Curve P-256 which are standardized in FIPS Pub 186-3 [1],[11].

- $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$
- $r = 115792089210356248762697446949407573529996955224135760342422259061068512044369$
- $a = p - 3$
- $b = 5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b$
- $G_x = 6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296$

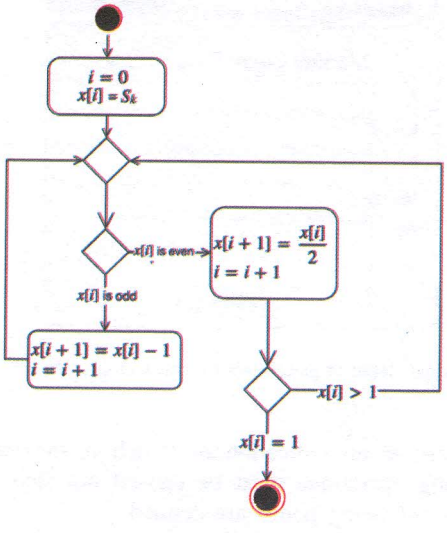


Fig. 4. UML Activity Diagram of the Iterations Algorithm

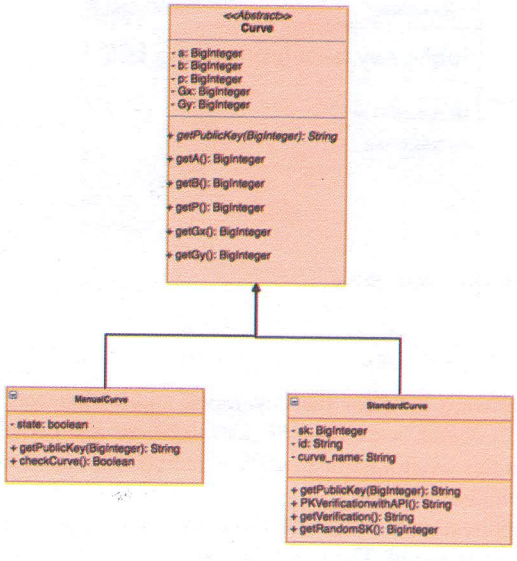


Fig. 5. UML Class Diagram of the $P_k$ Generator

- $G_y = 4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5$
- $1 < S_k < r - 1$

Where

- $p$: Prime number, value of finite field.
- $r$: Order of base point $G$.
- $a$ and $b$: Coefficients of curve P-256.
- $G_x$ and $G_y$: Initial coordinates of the curve

## III. Design and Implementation

The design and implementation phase of the asymmetric key generator through the evaluation of EC begins by defining the requirement based on the limiting in the calculation of points on the curve. As explained above the only allowed operations are double the point and the addition of one point with another. Therefore, it is necessary to define an optimized algorithm of iterations to the reach a required point. The implemented software is written entirely using Java. Since this software is part of another processes related to Display Authenticated Association, the algorithms already implemented using web interfaces, libraries or similar, are not convenient.

### A. Iterations Algorithm

The Iteration Algorithm aims to reach a required point $R = P_k$ from a point $G$ and minimize the number of steps until the desired value is obtained. As expressed in (16), the value of $P_k$ is equal to the result of the scalar multiplication between point $G$ and an integer value $S_k$ over a finite field $p$. The algorithm takes as an argument the value of the ($S_k$) and finds the fastest path starting from $S_k = 1$. Figure 4 illustrates using a UML activity diagram the corresponding algorithm.

Once all the values have been determined to reach the integer $S_k$, each point is calculated. This procedure involves modular computing. Using Java, *BigInteger* data type which contains a set of methods that work in classical arithmetic as well as modular arithmetic such as *mod ()*, *modInverse ()*, *multiply ()*, *pow ()*, etc., facilitates the implementation of a method of calculation.

### B. Classes Definition

The proposed generator emerges from the idea of entering the parameters of the curve manually to verify the operation of the generator, as well as generate automatically for the curve defined in IEEE 802.15.6 [12],[13]; therefore, three classes have been defined as shown in the class diagram shown in Figure 5.

The first class *(Curve)* is abstract and contains the curve parameter values as attributes along with five regular methods that will be inherited by the subclasses and the abstract method *getPublicKey()*.

The second class *(ManualCurve)* and third class *(StandardCurve)* inherit the attributes of the *(Curve)* class and the methods defined there. Both classes will be forced to implement the *getPublicKey ()* method, which will return the value of the $P_k$. The *ManualCurve* class, in addition to the inherited method, implements a *checkCurve ()* method, which determines whether the condition of the parameters of the equation of the elliptic curve defined in (13) is met. The *StandardCurve* class implements three more methods of *getPublicKey ()*, these are: *PKVerification ()*, *getVerification ()* and *getRandomSK ()*. The *getRandomSK ()* method generates a pseudorandom integer value $S_k$ acceptables in parameters defined by IEEE 802.15.6 Standard.

Having three classes defined with their respective attributes and methods, the results of the execution of the tool would not be visible to the end user without the instantiation of one of the classes and the display of information through the screen.The simplest case would allow the output of the result through the command shell. However, in order to achieve a much more user-friendly tool, we developed our own graphical interface.

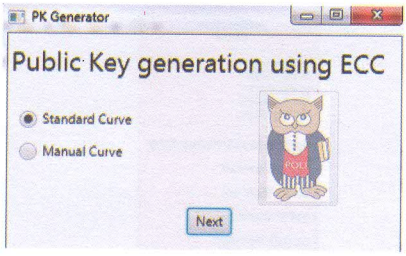


Fig. 6. Application Main Scene

### C. Guide User Interface

The software used for the developed application was *Netbeans 8.2)*, which uses *Java SE Development Kit 1.8.131.11 (JDK)*. In this environment *JavaFX* is included to facilitate the implementation of a GUI.

The application consists of three user interfaces that are deployed as required. The first GUI has a start menu where we can chose between a standardized curve or a curve to entered its parameters manually as shown in Figure 6.

If the option selected by the user was the Standard Curve, a new GUI with the necessary options for the calculation of the $P_k$ by this method will be deployed. Then the $S_k$ is entered manually or otherwise the program will generate a pseudo-random key, to finally calculate the $P_k$. For the case where the selected option is Manual Curve, the third interface is displayed. In this case, all the necessary parameters must be entered to carry out the calculation of $P_k$. These parameters to be entered will be: the coefficients of the equation of the elliptic curve $a$ and $b$, the value of the finite field $p$ on which to work, the generating point of the elliptic curve given by $G_x$ and $G_y$, and the $S_k$ All the above parameters are sent as input arguments to the methods of the respective class instances to perform the calculation of $P_k$ by this method.

For both cases the $P_k$ is shown in the form of coordinates, corresponding to a point on the elliptic curve. The coordinates of this point are displayed in a message box on-screen and saved on a file, which is updated each time a new computation is performed. The $S_k$ used to calculate $P_k$ is also stored in a separate text file.

## IV. Evaluations and Results

Once the design and implementation phase were completed, the tool was evaluated. Having two different scenarios requires separate testing in order to ensure the correct operation of the application.

### A. Manual Curve

For the testing of the scenario of the curve with parameters entered manually, the following analysis and exercise is proposed. Let be the EC defined in equation (18) for which we want to find the factor $i$ satisfying equation (19).

$$y^2 = x^3 + 7x + 4 \tag{18}$$

$$i(0,2) = (3,1) \tag{19}$$

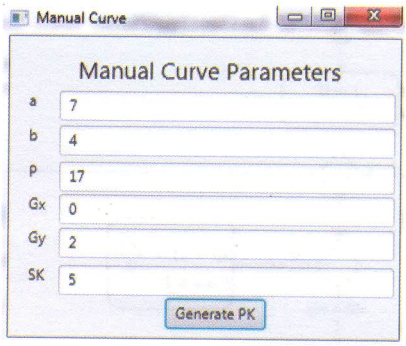


Fig. 7. Scenario 1: Input of parameters for calculation of $P_k$

Because there is no indication of which is the value, the corresponding operations must be carried out step by step. For this, the following points are defined.

$$G = (0,2)$$
$$P = (3,1)$$

The operations to find the factor $i$ will be based on equations (3), (4), (5), (7), (8) and (17).The first step is to find $2G$. For this process the value of $\lambda$, which is given in equation (5), must first be calculated. Once we determine $\lambda$ we proceed to the calculation of $(x_R, y_R)$, by (3) and (4). The resulting point of the operation is given in equation (20).

$$2G = (2,3) \tag{20}$$

The process will be similar for the other iterations, which are expressed in equations (21), (22), (23).

$$3G = 2G + G = (2,3) + (0,2) = (11,1) \tag{21}$$

$$4G = 3G + G = (11,1) + (0,2) = (15,4) \tag{22}$$

$$5G = 4G + G = (15,4) + (0,2) = (3,1) \tag{23}$$

Equation (23) gives the desired relation in equation (19), which would indicate that the wanted value of factor $i$ is five. The $P_k$ $(3,1)$ is derived from the $S_k$ equal to five. Figure 7 shows the input of parameters in the generation tool with a value of $S_k$ equal to five.

Figure 8 shows the generated $P_k$, thus verifying the correct operation of the tool.

One advantage about the using of manual curve is the capability of inserting new values of initial point in the curve $G_x$, $G_y$, this advantage may be used when is necessary making operations on a $P_k$ already generated. Unlike other implemented applications, this allows the modification of the initial points values and as mentioned before, this is part of a larger project where is mandatory to modify the initial points and find another valid point of the elliptic curve, as its evident in Display Authenticated Association process.

### B. IEEE 802.15.6 Standard Curve

For the case of the curve with defined parameters will be used the one defined in the IEEE 802.15.6 Standard, the curve

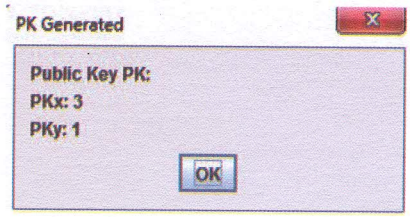


Fig. 8. Scenario 1: Generation of the $P_k$

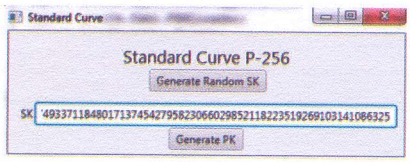


Fig. 9. Scenario 2: Generation of the $S_k$


Fig. 10. Public Key Generated

P-256, which generates a $P_k$ of 256 bits. The generation of the $S_k$ will be done pseudo-randomly, fulfilling the parameters described in section II. Figure 9 shows the generation of $S_k$ while the result of generating the $P_k$ can be seen in Figure 10.

The $P_k$ obtained using the P-256 curve was verified using the Bouncy Castle Provider API [4], which includes, within its classes, methods capable of evaluating different types of standardised curves. The result of the test was successful, thus verifying the validity of this developed tool and the algorithm implemented.

In both scenarios, the functionality of the tool has been checked without neglecting the handling of exceptions, either by inadequate input of information by the user, or at the time of computing the point of infinity $O$.

### C. Comparison with other applications

The application developed and shown in this article in comparison with other ones such as those available in [14], has some advantages for our research. One of the main advantages is this allows to select new values of $G_x$ and $G_y$ as mentioned.

Another advantage is the capacity to obtain the $P_{kx}$ and $P_{ky}$ points separately in both manual and standard curves. The frame fields of the association protocols defined in IEEE Standard 802.15.6 requires PK separated by coordinates [2]. In addition this application allows to obtain in hexadecimal format both the secret keys and the public keys, a necessary part for structuring of the management frames in IEEE 802.15.6.

## V. Conclusions and Futures Works

The developed application successfully fulfils its objectives, which not only manage the proper handling of exceptions generated by the misuse of the tool in the execution process; also, the maximum optimization of the algorithm to decrease the processing requirements and the delivery of proven results.

This tool has demonstrated that it is not limited to the use of the P-256 curve used in the IEEE 802.15.6 Standard, since it could works for other standard curves parameters like: secp192r1, secp224r1, secp384r1or non standard ones. For this reason, it allows the calculation of public keys to use in many others security processes.

The resultants point that has been calculated, called $P_k$ coordinate, is not used directly in the frames of association between a node and a hub. However, this point is used directly on the calculus of parameters to exchange between the entities.

This application has its own methods for computing a $P_k$ with any parameters. The results are assessed and verified with the results obtained using the Bouncy Castle Provider API [4]; Which is limited to a defined number of curves, not being as generic as our solution.

The algorithm implemented is useful to start an association process defined in IEEE 802.15.6, Display Authenticated Association. With this tool we can get the parameter $P_k$ and calculate Diffie-Hellman key to continue with the association process which is being implemented. Since this work, it is possible to create an application which compute a CMAC process to reach the complete process association.

Other tools that perform the calculation of public keys with ECC hardly fit the necessary requirements for the development of a vulnerability analysis in the Display Authenticated Assosiation process limiting its use within this research. However, it does not mean that this type of tools can be used in other environments.

## References

[1] K. S. Kwak, S. Ullah, and N. Ullah, "An overview of ieee 802.15. 6 standard," in *Applied Sciences in Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on*. IEEE, 2010, pp. 1–6.

[2] "Ieee standard for local and metropolitan area networks - part 15.6: Wireless body area networks," *IEEE Std 802.15.6-2012*, pp. 1–271, Feb 2012.

[3] X. Huang, D. Liu, and J. Zhang, "An improved ieee 802.15. 6 password authenticated association protocol," in *Communications in China (ICCC), 2015 IEEE/CIC International Conference on*. IEEE, 2015, pp. 1–5.

[4] D. Hook, *Beginning cryptography with Java*. John Wiley & Sons, 2005.

[5] W. Stallings, *Fundamentos de seguridad en redes: aplicaciones y estándares*. Pearson Educación, 2004.

[6] S. Madakam and H. Date, "Security mechanisms for connectivity of smart devices in the internet of things," in *Connectivity Frameworks for Smart Devices*. Springer, 2016, pp. 23–41.

[7] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[8] V. G. Martinez and L. H. Encinas, "Implementing ecc with java standard edition 7," *International Journal of Computer Science and Artificial Intelligence*, vol. 3, no. 4, p. 134, 2013.

[9] N. Koblitz, *A course in number theory and cryptography*. Springer Science & Business Media, 1994, vol. 114.

[10] T. Güneysu and C. Paar, "Modular integer arithmetic for public key cryptography," in *Secure Integrated Circuits and Systems*. Springer, 2010, pp. 3–26.

[11] S. Ullah, M. Mohaisen, and M. A. Alnuem, "A review of ieee 802.15. 6 mac, phy, and security specifications," *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, p. 950704, 2013.

[12] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 157–175.

[13] A. Abidi, B. Bouallegue, and F. Kahri, "Implementation of elliptic curve digital signature algorithm (ecdsa)," in *Computer & Information Technology (GSCIT), 2014 Global Summit on*. IEEE, 2014, pp. 1–61.
[14] Cryptomathic. Ecc calculator. [Online]. Available: http://extranet.cryptomathic.com/ecc/index

**Gustavo Dávila** He was born on July 6th, 1994. He finished his career in Electronics and Telecommunications Engineering at Escuela Politécnica Nacional, and he is in process of obtain his Degree. He works at Huawei Technologies Co, Ltd., as Project Control Manager, since September 2017. He is passionate about Site Reliability Engineering (SRE) in order to improve process and daily operation developing software for Data Analysis. Junior SRE certified in Queretaro, Mexico. His research interests include WBAN, IoT, 5G, and Data Science.

**Oscar Torres** He was born on November 22, 1994 in Quito. He finished his secondary studies at Unidad Educativa Hermano Miguel "La Salle". He is currently finishing his bachelor degree in Electronics and Telecommunications Engineering at National Polytechnic School. He has been granted several times the Academic Excellence Scholarship for high Academic Performance Indicator. What has allowed him to participate in several activities on behalf of the career. He works at Uniplex S.A as Support Engineer in SAP ASE Database and related products. His areas of interest are: Cellular Networks, Unified Communications, Networking, WBAN, IoT, Information Security, Big Data, Data Managment.

**Jorge Eduardo Rivadeneira** He was born in Quito in September 1989. He obtained his Bachelor's degree in Electronics and Information Networks from the National Polytechnic School in 2013. He worked as lecturer of Electronics Devices, Electronic Circuits, Linux Operating Systems, and Object Oriented Programming during the years 2014, 2015 and 2017 respectively, in addition to being Head of the General Electronics Laboratory. Master in Cyber Security by the University of Southampton in the United Kingdom. Currently he co-directs a degree project at the National Polytechnic School. Within its areas of interest and research are: Information Security, Electronic Security, Cryptography, Machine Learning, IoT and Personal Area Networks.

**Pablo Hidalgo** He obtained the degree of Electronics and Telecommunications Engineer in the Escuela Poitecnica Nacional (1985) being declared the best graduate of his promotion. Scholarship by the German Government and sponsored by the E.P.N. He completed a postgraduate study in Telecommunications at Deutsche Bundespost (1988 - 1990). He obtained the title of Master in Connectivity and Telecommunications Networks at E.P.N. (November 2014). He was promoter and Coordinator of the Engineering Career in Electronics and Information Networks at E.P.N. (2000 - 2007) (2013 - 2014) (2017- present). He has directed more than 150 theses and degree projects. His current areas of interest are: Information Networks, Wireless Communications and Data Transmission. He is a member of the Association for Computing Machinery (ACM) and a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE).