# Anomalous Behaviour of Cryptographic Elliptic Curves over Finite Field

Radek Fujdiak[1], Petr Dzurenda[1], Petr Mlynek[1], Jiri Misurec[1], Milos Orgon[2], Bezzateev Sergey[3]
[1]*Department of Telecommunications, Brno University of Technology,*
*Technicka St. 12, 616 00 Brno, Czech Republic*
[2]*Department of Telecommunications, Slovak University of Technology in Bratislava,*
*Ilkovicova St. 3, 812 19 Bratislava, Slovak Republic*
[3]*Technologies of Information Security, Saint-Petersburg University of Aerospace Instrumentation,*
*Bolshaya Morskaya St. 67, 190 000 Saint Petersburg, Russia*
*fujdiak@feec.vutbr.cz*

***Abstract*—New wireless technologies and approaches enable to connect even the simplest sensors with limited computational power to the global network. The need for efficient and secure solutions is growing with the wider use of these devices. This paper provides a new method for speed optimization of Elliptic Curve Cryptography operations which are frequently used in the light-weight secure communication algorithms. This method is based on the anomalous behaviour of specific elliptic curves. We analyse more than 60 curves of various international standards. Further, our method is less complex, easy to deploy and comparable effective as ordinary, more complex methods. Last but not least, we show the importance of future research in the area of elliptic curve parameterization.**

***Index Terms*—Cryptography; data security; elliptic curves; information security.**

## I. INTRODUCTION

Elliptic curves have been studied as a mathematical concept since the second century A.C., while the name "elliptic" was given in the nineteenth century [1]. However, the concept of Elliptic Curve Cryptography (ECC) has only been known about in the last 30 years. The first use of elliptic curves in cryptography was by H. W. Lenstra for elliptic curve factorization which was used as the fastest algorithm to find factors of large integers [2]. However, N. Koblitz and V. Miller are considered as the founders of ECC. In 1985 N. Koblitz [3] and V. Miller [4] independently proposed the use of a group of points on an elliptic curve defined over a finite field. Over the past 30 years, ECC has become a key part of many current cryptosystems, cryptographic schemes and algorithms, e.g., Elliptic curve Diffie-Hellman (ECDH), Elliptic curve Integrated Encryption Scheme (ECIES), Elliptic curve Digital Signature Algorithm (ECDSA), Edwards-curve Digital Signature Algorithm (EdDSA), Elliptic curve Menezes-Qu-Vanstone (ECMQV), and Elliptic curve Qu-Vanstone (ECQV). ECC is also recommended in different standards, e.g., by Standards for Efficient Cryptography Group (SECG) in SEC1 [5], SEC2 [6] and SEC4 [7]; by the National Institute of Standards and Technology (NIST) in 800-57 [8] and FIPS 186-2 [9]; by Accredited Standards Committee X9 in ANSI X9.62 [10] and ANSI X9.63 [11]; by Institute of Electrical and Electronics Engineers in IEEE 1363-2000 [12]; and by Wireless Application Forum in WTLS [13]. Furthermore, many real implementations use and provide ECC primitives and algorithms, e.g., Bouncy castle [14], TinyECC [15] Crypto++ [16], OpenSSL [17], and FlexiProvider [18].

The main advantage of ECC is reaching the same level of security by using a smaller key compared to classical asymmetric cryptographic schemes based on factoring modulus or a discrete logarithm. This reduces such factors as memory storage requirements, key transmission time, arithmetic computation power costs and the bandwidth [19], [20]. This key advantage is the reason why ECC is favoured in internet-based applications and preferred for constrained devices with low computational power and low memory storage, smart cards and cryptographic tokens. These constrained devices are portable, small, and lightweight and have low processing power, parameter storage and memory [21]. These areas might be summarized as limited devices, which are defined mainly by their low computational power and low memory storage capacity. This paper deals with the research of new effective methods for speed and memory optimization of ECC used in limited devices. Many current methods deal with ECC optimization by finding new effective methods. These solutions are hard to deploy and often require main system changes. On the other hand, we try to provide a solution which is comparably effective as more complex methods, but very easy to deploy.

The rest of the paper is organized as follows. Section II summarizes related works. Section III introduces our experimental environment. Further, in Section IV the main experimental measurements of ECC are provided. The discussion and comparison with other works is provided in Section IV. Finally, Section V summarizes our conclusions.

Manuscript received 29 April, 2017; accepted 21 July, 2017.
Research described in this article was financed by the National Sustainability Program under grant LO1401. For the research, the infrastructure of the SIX Center was used.

## II. CURRENT AND RELATED WORK

Due to new wireless technologies and other worldwide technological changes and progresses made, the modern concept of The Internet of Things (IoT) is attracting more attention. IoT is based on connecting even the simplest sensors (limited devices) with each other as well as connecting these to the global network infrastructure [22]. These connected devices have a wide use over many areas and should provide heightened amounts of information which might be used to increase the life quality of the citizens, ensure higher level of process automation, discover new options for optimization, general security purposes and even for saving lives in disasters [23]. The wide use of connected devices and objects also gives rise to new challenges in the areas of architecture, availability, reliability, mobility, performance, management, scalability, interoperability, security and privacy [24]. Basic security issues might be solved by encryption, authentication and authorization algorithms. However, due to the usage of limited devices, the implementation of these algorithms is a difficult task. ECC is nowadays used among others to solve these difficulties and it also provides secured solutions for limited devices [25].

Most of the current works that are focused on improving the efficiency of ECC algorithms are based on:

– developing new and more efficient algorithms i.e. for point multiplication [26]–[28],

– creating new and more efficient curves [29],

– using different unusual algorithms or more efficient mathematical fields for ECC [30], [31].

In general, these methods are appropriate and they bring significant and efficient results, but they are also often very hard to deploy. However, if we take a closer look at the curve speed in different kinds of algorithms, methods and implementations, we can see significant speed differences even for curves of similar types. In the work [32] we showed our lightweight ECDH implementation with various curves, where curves of the same size had up to 50 % speed difference. Our previous work [33] already showed the differences between the efficiency of prime field based and field of characteristic 2 based elliptic curve systems in limited devices. The prime field based curves show a higher efficiency on limited devices than curves with field of characteristic 2 (by tens of percent). Based on these results we believe the efficiency might be improved not only with new algorithms or new curves, but also by choosing the right elliptic curve from the current ones. Our solution will be easier to deploy with similar effectiveness. Additionally, the in-depth research of current elliptic curves and their behaviour on specific limited hardware might also bring significant results for new algorithms or curves in the future.

This paper provides a new look at the efficiency of various cryptographic elliptic curves. We provide a clear description of elliptic curve parameterization. On this basis, we explain the links between computation speed and elliptic curve domain parameters. This new approach might be used for creating faster and memory friendly curves. Further, the experimental measurements which are showing the discovered anomalous behaviour of some elliptic curves are provided. This anomaly might be used for simple speed and memory optimization without additional needs.

## III. EXPERIMENTAL BACKGROUND AND PRE-SETS

### A. Network Model

We are developing a secure solution for the smart grid. Our network model helps us to develop solutions which might be rapidly implemented into real applications. The smart grid often uses limited devices i.e. for measuring consumption (water, gas, heat take-offs), failure states indicators, power quality monitors and many others. Fig. 1 shows our smart grid network model for this measurement where we can see the part with limited resources (limited devices) and also the part with non-limited resources.

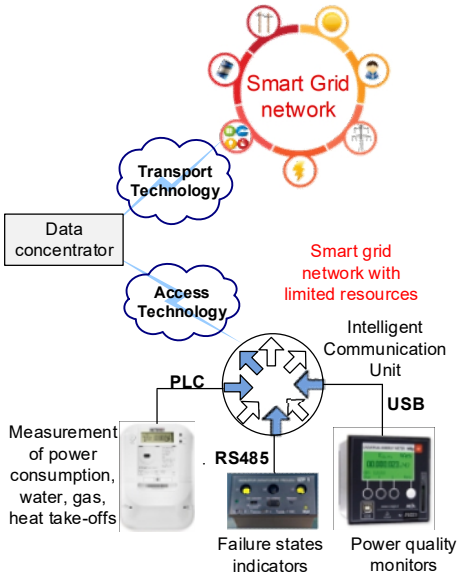


Fig. 1. Smart Grid network for remote data acquisition [34].

The Intelligent Communication Unit is a limited device which connects even the simplest meters, sensors and indicators via a specific interface (i.e. PLC, RS485, USB) and communicate over access technology (i.e. PLC, Wireless) with a data concentrator. In this experiment, we consider a microcontroller MSP430 and a computer unit, the Raspberry Pi 2, for the non-limited area as a core of the communication unit for the limited area.

### B. Experimental Measurements Pre-Sets

The MSP430 of the 5438A family is considered as a core for limited devices in our measurements. MSP430 is an ultra-low power micro-controller with power consumption in hundreds of µA for the active mode and units of µA for the standby/sleep mode. This microcontroller has 256 kB FLASH, 16 kB RAM, 32-bit multiplier, high/low frequency crystal (32 MHz/32 kHz) and allow 16-bit operations; more details about technical specification of MSP430 can be found in [35]. For MSP430, we used our lightweight implementation of elliptic curve primitives and ECDH algorithm.

The Raspberry Pi 2B is considered in our measurements as a core for non-limited devices. We included this unit in our measurement to obtain more data. The Raspberry Pi 2B is a simple computer unit based on the Broadcom BCM2836 processor. This device has a 900 MHz quad-core ARM

Cortex A7 processor and 1 GB LPDDR2 memory; more details about the technical specification of the Raspberry Pi 2B are in [36]. For the Raspberry Pi 2B, we used the OpenSSL library which implements elliptic curve primitives, curves of different standards and many different algorithms based on elliptic curves i.e. ECDH or ECDSA.

### C. Elliptic Curve Cryptography Implementations

Our light-weight solution is precisely described in [32] and it is available on [37]. We have implemented more than 60 elliptic curves of different kinds of standards together with a big-number representation and basic and modular arithmetic. We use classical representation of elliptic curves by domain parameters which are pre-generated and defined by a standard. For point multiplication, the non-adjective form of multiplication *w-NAF* is used (*Double-and-add* variation, D&A) and Montgomery modular algorithms are used for other modulo operations. The operations are optimized for the low-power microcontroller MSP430 and ECDH algorithm.

The OpenSSL is a non-lightweight ready-to-use library for a wide range of applications. This solution brings all the necessities for implementing a system based on ECC. This library also contains all the necessities for the ECDH and ECDSA algorithms. A detailed description of this library is in [38] and it is available on [39].

## IV. Impact of Elliptic Curve Domain Parameters on Speed Efficiency

Each elliptic curve is defined by the field and domain parameters. We will work with two different field types, field of characteristic 2 and prime field. The elliptic curves over finite field of characteristic 2 $(F_2m)$ have seven domain parameters (septuple)

$$T_{F_2m} = (m, f(x), a, b, G, n, h), \tag{1}$$

where $m$ is an integer specifying $F_2m$; $f(x)$ is irreducible binary polynomial of degree $m$ specifying the polynomial basis representation of $F_2m$; $a$, $b$ are two elements of $F_2m$ ($a$, $b \in F_2m$) specifying an elliptic curve E($F_2m$) defined by

$$E: y^2 + xy = x^3 + ax^2 + b \text{ in } F_2m, \tag{2}$$

where variable G is a base point $G = (x_G, y_G)$ on E($F_2m$); prime $n$ is order of G; and finally the integer $h$ is the cofactor

$$h = \#E(F_2m)/n. \tag{3}$$

The elliptic curves over finite prime field ($F_p$) have six domain parameters (sextuple)

$$T_{F_p} = (p, a, b, G, n, h), \tag{4}$$

where $p$ is an integer specifying $F_p$; $a$, $b$ are two elements of $F_p$ ($a$, $b \in F_p$) specifying an elliptic curve E($F_p$) defined by

$$E: y^2 \equiv x^3 + ax + b \pmod{p}, \tag{5}$$

where variable G is a base point $G = (x_G, y_G)$ on E($F_p$); prime $n$ is order of G; and finally the integer $h$ is the cofactor

$$h = \#E(F_p)/n. \tag{6}$$

There are three main mathematical operations with elliptic curves (or their points): point addition, point doubling and point (scalar) multiplication. These operations are the basis for all other mathematical algorithms for ECC i.e. the ECDH and ECDSA algorithms. The time consumption and computational difficulty of these basic operations with points of elliptic curves is obviously curve-size dependent. This means operations with curves of larger domain parameters should be slower than operations with curves of smaller domain parameters.

We independently measured more than 60 elliptic curves on two different software implementations (own and OpenSSL implementation) and two different hardware devices (the MSP430 limited-device and the Raspberry Pi 2B non-limited device). The size of domain parameters for each measured elliptic curve is in Appendix A (Table A-I, A-II). Table I and II summarize our main experimental results where *curve name* is the official elliptic curve name defined by the standard SEC (sect/secp elliptic curves), WTLS (wtls elliptic curves), ANSI (prime, c2pnb/tnb elliptic curves) or IPSec (ipsec elliptic curves). *Field* defines the $2^m$ of $F_2m$ (or $p$ of $F_p$). *No.* is the number of significantly smaller domain parameters than degree $m$ (i.e. curve *wtls1* with field $F_{2^{114}}$ has domain parameters $m_{114b}$, $a_{1b}$, $b_{1b}$, $x_{113b}$, $y_{112b}$, $n_{112b}$ the *No.* is 2, because of $a$, $b$). Parameter $t_G$ is time needed for curve ECDH key and ECDH parameters generation with the specific curve. The $\Delta t$ is defined as a time difference between the fastest curve and the concrete curve and it is given as a percent (value 0.00 mark the fastest curve), where $\Delta t_1$ is measurement on Raspberry Pi 2B and $\Delta t_2$ is on MSP430.

Table I shows the results for measured elliptic curves over the field of characteristic 2 on the Raspberry Pi 2B with OpenSSL implementation. The results are as expected and more samples or other measurements on different platforms are not necessary. The curves with a higher degree are generally slower. Further, the curves of the same degree with small domain parameters ($a$, $b$, $x$, $y$) are generally faster than curves with normal sized domain parameters (same size as curve degree). Some higher degree curves with small domain parameters (ipsec3, $F_{2^{156}}$) are even comparably as fast as lower degree curves with non-small domain parameters (sect131r2, $F_{2^{132}}$). Figure 2 shows an example of a different curve speed for $F_{2^{164}}$, $F_{2^{177}}$ and $F_{2^{186}}$ on the Raspberry Pi 2B with OpenSSL implementation of elliptic curves. The white colour is for curves with four, grey is for curves with two and black is for curves with zero small

domain parameters. As evident, the fastest curve is the curve with the smallest domain parameters.

TABLE I. EXPERIMENTAL RESULTS FOR MEASURED ELLIPTIC CURVES OVER FIELD OF CHARACTERISTIC 2 ON RASPBERRY.

| Curve name | Field | No. [-] | $t_{G1}$ [µs] | $\Delta t_1$ [%] |
|---|---|---|---|---|
| **GF($2^{114}$)** | | | | |
| wtls1 | $2^{114}$ | 2 | 3,362 | 0.00 |
| wtls4 | $2^{114}$ | 0 | 3,496 | +3.99 |
| sect113r1 | $2^{114}$ | 0 | 3,500 | +4.10 |
| sect113r2 | $2^{114}$ | 0 | 3,528 | +4.94 |
| sect113k1 | $2^{114}$ | 0 | 3,487 | +3.72 |
| **GF($2^{132}$) and GF($2^{156}$)** | | | | |
| sect131r1 | $2^{132}$ | 0 | 6,235 | 0.00 |
| sect131r2 | $2^{132}$ | 0 | 6,464 | +3.67 |
| ipsec3 | $2^{156}$ | 4 | 6,417 | +2.92 |
| **GF($2^{164}$), ($2^{177}$) and GF($2^{186}$)** | | | | |
| wtls3 | $2^{164}$ | 2 | 8,174 | +2.14 |
| wtls5 | $2^{164}$ | 0 | 8,815 | +10.15 |
| sect163r1 | $2^{164}$ | 0 | 8,802 | +9.98 |
| sect163k1 | $2^{164}$ | 2 | 8,197 | +2.42 |
| c2pnb163v1 | $2^{164}$ | 0 | 8,822 | +10.23 |
| c2pnb163v2 | $2^{164}$ | 0 | 8,795 | +9.92 |
| c2pnb163v3 | $2^{164}$ | 0 | 8,795 | +9.90 |
| c2pnb176v1 | $2^{177}$ | 0 | 8,739 | +9.20 |
| ipsec4 | $2^{186}$ | 4 | 8,003 | 0.00 |
| **GF($2^{234}$) and GF($2^{240}$)** | | | | |
| wtls10 | $2^{234}$ | 2 | 15,865 | +0.23 |
| wtls11 | $2^{234}$ | 1 | 17,474 | +10.40 |
| sect233r1 | $2^{234}$ | 1 | 17,493 | +10.52 |
| sect233k1 | $2^{234}$ | 2 | 15,828 | 0.00 |
| sect239k1 | $2^{240}$ | 2 | 16,244 | +2.63 |
| c2tnb239v1 | $2^{240}$ | 0 | 17,890 | +13.03 |
| c2tnb239v2 | $2^{240}$ | 0 | 17,843 | +12.73 |
| c2tnb239v3 | $2^{240}$ | 0 | 17,784 | +12.36 |
| **GF($2^{284}$)** | | | | |
| sect283r1 | $2^{284}$ | 1 | 32,072 | +11.91 |
| sect283k1 | $2^{284}$ | 2 | 28,659 | 0.00 |
| **GF($2^{410}$)** | | | | |
| sect409r1 | $2^{410}$ | 2 | 76,160 | +14.35 |
| sect409k1 | $2^{410}$ | 1 | 66,605 | 0.00 |
| **GF($2^{572}$)** | | | | |
| sect571r1 | $2^{572}$ | 2 | 174,749 | +14.86 |
| sect571k1 | $2^{572}$ | 1 | 152,143 | 0.00 |

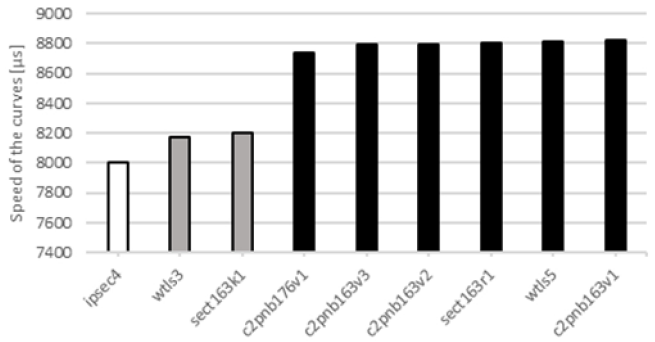


Fig. 2. Comparison of different curves over field of characteristic 2 – $F_{2}164$ , $F_{2}177$ and $F_{2}186$ on Raspberry with OpenSSL implementation.

The curves over the prime field shows a significant anomaly in behaviour. Table II shows the results for measured elliptic curves over the prime field on the Raspberry Pi 2B and also on the MSP430 (for a greater range of samples). The curves with a higher degree are generally slower. However, the curves of the same degree with small domain parameters ($a$, $b$, $x$, $y$) are not generally faster than curves with normal sized domain parameters (same size as curve degree). Conversely, the small domain parameters had no effect or made the curve even slower than the curves with the same sized domain parameters. Some higher degree curves with normal sized domain parameters (secp128r1, $F_{128}$) are even comparably as fast as lower degree curves with smaller domain parameters (wtls8, $F_{112}$). Figure 3 shows an example of a different curve speed for $F_{160}$ on the Raspberry Pi 2B with OpenSSL implementation of elliptic curves. The grey is for curves with two and black is for curves with zero small domain parameters. As we can see the slowest curve is the curve with the smallest domain parameters.

TABLE II. EXPERIMENTAL RESULTS FOR MEASURED ELLIPTIC CURVES OVER PRIME FIELD ON RASPBERRY PI 2B AND MSP430.

| Curve name | Field | n [-] | $t_{G1}$ [µs] | $\Delta t_1$ [%] | $t_{G2}$ [µs] | $\Delta t_2$ [%] |
|---|---|---|---|---|---|---|
| **GF(112) and GF(128)** | | | | | | |
| wtls6 | 112 | 0 | 2,555 | 0.00 | 10,696 | +8.68 |
| wtls8 | 112 | 4 | 2,953 | +15.58 | 10,897 | +10.72 |
| secp112r1 | 112 | 0 | 2,570 | +0.59 | 9,842 | 0.00 |
| secp112r2 | 112 | 0 | 2,599 | +1.72 | 11,413 | +15,96 |
| secp128r1 | 128 | 0 | 2,825 | +10.57 | 12,092 | +22,86 |
| secp128r2 | 128 | 0 | 2,899 | +13.46 | 12,986 | +31,94 |
| **GF(160)** | | | | | | |
| wtls7 | 160 | 0 | 4,130 | 0.00 | 13,031 | 0.00 |
| wtls9 | 160 | 4 | 4,704 | +13.90 | 19,612 | +50.50 |
| secp160r1 | 160 | 0 | 4,143 | +0.31 | 13,050 | +0.15 |
| secp160r2 | 160 | 0 | 4,143 | +0.31 | 13,963 | +7.15 |
| secp160k1 | 160 | 2 | 4,557 | +10.34 | 15,080 | +15.72 |
| **GF(192)** | | | | | | |
| secp192k1 | 192 | 2 | 6,364 | +11.92 | 23,674 | 7.57 |
| prime192v1 | 192 | 0 | 5,716 | +0.53 | 25,400 | 13.04 |
| prime192v2 | 192 | 0 | 5,686 | 0.00 | 22,470 | 0.00 |
| prime192v3 | 192 | 0 | 5,701 | +0.26 | 24,171 | 5.36 |
| **GF(224)** | | | | | | |
| wtls12 | 224 | 0 | 7,483 | +0.09 | 26,631 | 0.00 |
| secp224r1 | 224 | 0 | 7,476 | 0.00 | 27,020 | 1.46 |
| secp224k1 | 224 | 2 | 8,388 | +12.20 | 30,771 | 15.55 |
| **GF(256)** | | | | | | |
| secp256k1 | 256 | 2 | 11,007 | +12.28 | - | - |
| prime256v1 | 256 | 0 | 9,803 | 0.00 | - | - |

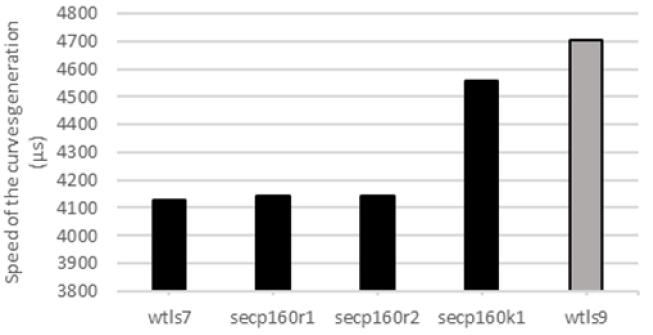


Fig. 3. Comparison of different curves over prime field $F_{160}$ on Raspberry with OpenSSL implementation.

Figure 4 shows the same example as in Fig. 3, but on the

MSP430 with our implementation of elliptic curves. The grey is for curves with two and black is for curves with zero small domain parameters. As we can see here also the slowest curve is the curve with the smallest domain parameters.

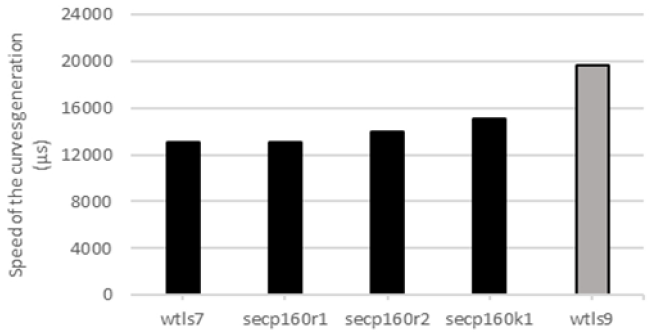


Fig. 4. Comparison of different curves over prime field $F_{160}$ on MSP430 with our implementation of elliptic curves.

The results show that on the field of characteristic 2 we can reduce up to 5 % to 15 % of the time consumption on smaller degree curves (114–156 b) and up to 15 % of the time consumption on the higher degree curves (164–572 b) only by choosing the right curve with small domain parameters. On the prime field we can reduce the time consumption by 10 % to 50 % only by choosing the right curve with normal not-smaller domain parameters. The measurements reveal the anomalous behaviour of the prime field curves. This means that it cannot be clearly concluded whether the smaller domain parameters will have a positive effect on speed of elliptic curve operations. They result in reduced memory requirements, but not necessarily a reduction in speed.

## V. DISCUSSION

From our measurements we expect significant results which should help in choosing the right elliptic curve for a real application from the point of view of time and memory requirement. Our results prove that the size of domain parameters has a significant impact on computational complexity and time consumption of algorithms which work with elliptic curves. Further, we show that the time consumption of an elliptic curve algorithm can be significantly reduced only by choosing the right elliptic curve. Table III shows the results of chosen current and relevant works dealing with ECC optimization, new methods and algorithms or new curves. As we can see, the maximum speed reduction is about 50 %, but very often much smaller.

TABLE III. COMPARISON OF SPEED REDUCTION WITH OTHER CURRENT AND RELEVANT WORKS.

| Short description | Speed reduction [%] | #Ref. |
|---|---|---|
| Work focused on the use of ECC in wireless sensor networks. They present a new technique to speed up the multiplication operation. | 46–48 % (D&A) 32–37 % (wNAF) | [26] |
| Work focused on double-and-add algorithm for point multiplication and application to the Fibonacci sequence. The work compares only chain size, where they achieved up to 18 % differences. But final speed tests are missing. | no data | [27] |
| Another work focused on new more efficient design of elliptic curve point multiplication based on Montgomery modular multiplication. | 4–38 % | [28] |
| The authors introduce the new curve 25519 with significant speed efficiency on hardware. A final comparison of this curve on specific hardware and with other curves is missing. | no data | [29] |
| The work focused on ECC in the Internet of Things. The authors implement their solution of ECDSA on a small processor and gain significant speed results compared to the other implementations by using FPGA acceleration. | 50 % | [30] |
| Another work providing a new algorithm based on modified traditional extended Euclidean Great Common Divisor (GCD). | >50 % | [31] |

Our solution shows very close results to the more complex optimization methods by simply only choosing the right curve. This fact might be used for easy speed and memory optimization without any further needs of algorithm change.

## VI. CONCLUSIONS

The Elliptic Curve Cryptography (ECC) is nowadays already a frequently used method for lightweight secured communication solution and it is spread over wide areas. We might see it also as a solution for future LPWAN, MANET, VANET and many others. We showed as an example the Smart Grid network (Fig. 1), in which the low-power devices are used to gather the sensitive customers or network data from sensors as well as crucial management data. For securing this communication, these devices with limited physical resources need a sufficiently secure and resource effective cryptographic solution. For this purpose, the AES cipher and ECC are often used together, the ECC being always more resources demanding.

Many current works focused on improving the efficiency of ECC are dealing with developing new efficient algorithms, creating new efficient curves or trying to implement or develop new primitives. However, this paper shows an efficient and easy deployable solution for the speed optimization of ECC algorithms without the need of changing current algorithms, primitives or protocols. Our solution is based on choosing the right elliptic curve with right domain parameters. We provide measurements of more than 60 elliptic curves (Section IV) on two different hardware devices (limited and not-limited) with two different software implementations (our software and OpenSSL). These measurements show a significant impact of the domain parameters on the speed of basic elliptic curve operations. Compared with other works (focused on new algorithms, curves or more efficient operations), we achieved comparable results by only choosing the right elliptic curve and with a deeper look at its domain parameters (up to 50 % time reduction). Further, section V shows that many other current works achieved same or smaller time reduction by using much more complex methods. Last but not least, these facts demonstrate that in-depth research of elliptic curve parameterization might bring valuable results.

Future work should investigate the relationship between the domain parameters and the common computational algorithms in greater detail (i.e. point multiplication algorithms). Mathematical analysis is required for a deeper understanding of the speed dependency. We believe this research will bring very significant and valuable information for future cryptosystems and new elliptic curves.

## APPENDIX A

TABLE A-I. LENGTH OF DOMAIN PARAMETERS FOR MEASURED ELLIPTIC CURVES OVER FIELD OF CHARACTERISTIC 2.

| Curve name | m [b] | a [b] | b [b] | x [b] | y [b] | n [b] |
|---|---|---|---|---|---|---|
| **GF($2^{114}$)** | | | | | | |
| wtls1 | 114 | 1 | 1 | 113 | 112 | 112 |
| wtls4 | 114 | 110 | 112 | 112 | 112 | 113 |
| sect113r1 | 114 | 110 | 112 | 112 | 112 | 113 |
| sect113r2 | 114 | 111 | 112 | 113 | 112 | 113 |
| sect113k1 | 114 | 110 | 112 | 112 | 112 | 113 |
| **GF($2^{132}$) and GF($2^{156}$)** | | | | | | |
| sect131r1 | 132 | 131 | 130 | 128 | 131 | 131 |
| sect131r2 | 132 | 130 | 131 | 130 | 131 | 131 |
| ipsec3 | 156 | 0 | 19 | 7 | 9 | 154 |
| **GF($2^{164}$) and GF($2^{186}$)** | | | | | | |
| wtls3 | 164 | 1 | 1 | 162 | 162 | 163 |
| wtls5 | 164 | 163 | 160 | 163 | 161 | 163 |
| sect163r1 | 164 | 163 | 163 | 162 | 159 | 162 |
| sect163k1 | 164 | 1 | 1 | 162 | 162 | 163 |
| c2pnb163v1 | 164 | 163 | 160 | 163 | 161 | 163 |
| c2pnb163v2 | 164 | 161 | 163 | 158 | 163 | 162 |
| c2pnb163v3 | 164 | 163 | 162 | 162 | 163 | 162 |
| c2pnb176v1 | 177 | 176 | 175 | 176 | 175 | 161 |
| ipsec4 | 186 | 0 | 13 | 5 | 4 | 184 |
| **GF($2^{234}$) and GF($2^{240}$)** | | | | | | |
| wtls10 | 234 | 0 | 1 | 233 | 233 | 232 |
| wtls11 | 234 | 1 | 231 | 232 | 233 | 233 |
| sect233r1 | 234 | 1 | 231 | 232 | 233 | 233 |
| sect233k1 | 234 | 0 | 1 | 233 | 233 | 232 |
| sect239k1 | 240 | 0 | 1 | 238 | 239 | 238 |
| c2tnb239v1 | 240 | 238 | 239 | 239 | 239 | 238 |
| c2tnb239v2 | 240 | 239 | 239 | 238 | 239 | 237 |
| c2tnb239v3 | 240 | 233 | 239 | 239 | 238 | 236 |
| **GF($2^{284}$)** | | | | | | |
| sect283r1 | 284 | 0 | 1 | 283 | 281 | 281 |
| sect283k1 | 284 | 281 | 282 | 283 | 282 | 282 |
| **GF($2^{410}$)** | | | | | | |
| sect409r1 | 410 | 0 | 1 | 407 | 409 | 407 |
| sect409k1 | 410 | 1 | 406 | 409 | 407 | 409 |
| **GF($2^{572}$)** | | | | | | |
| sect571r1 | 572 | 0 | 1 | 570 | 570 | 570 |
| sect571k1 | 572 | 1 | 570 | 570 | 570 | 570 |

TABLE A-II. LENGTH OF DOMAIN PARAMETERS FOR MEASURED ELLIPTIC CURVES OVER PRIME FIELD.

| Curve name | p [b] | a [b] | b [b] | x [b] | y [b] | n [b] |
|---|---|---|---|---|---|---|
| **GF(112) and GF(128)** | | | | | | |
| wtls6 | 112 | 112 | 111 | 108 | 112 | 112 |
| wtls8 | 112 | 0 | 2 | 1 | 2 | 113 |
| secp112r1 | 112 | 112 | 111 | 108 | 112 | 112 |
| secp112r2 | 112 | 111 | 111 | 111 | 112 | 110 |
| secp128r1 | 128 | 128 | 128 | 125 | 128 | 128 |
| secp128r2 | 128 | 128 | 127 | 127 | 126 | 126 |
| **GF(160)** | | | | | | |
| wtls7 | 160 | 160 | 160 | 159 | 160 | 161 |
| wtls9 | 160 | 0 | 2 | 1 | 2 | 161 |
| secp160r1 | 160 | 160 | 157 | 159 | 158 | 161 |
| secp160r2 | 160 | 160 | 160 | 159 | 160 | 161 |
| secp160k1 | 160 | 0 | 3 | 158 | 160 | 161 |
| **GF(192)** | | | | | | |
| secp192k1 | 192 | 0 | 2 | 192 | 192 | 192 |
| prime192v1 | 192 | 192 | 191 | 189 | 187 | 192 |
| prime192v2 | 192 | 192 | 192 | 192 | 191 | 192 |
| prime192v3 | 192 | 192 | 190 | 191 | 190 | 192 |
| **GF(224)** | | | | | | |
| wtls12 | 224 | 224 | 224 | 224 | 224 | 224 |
| secp224r1 | 224 | 224 | 224 | 224 | 224 | 224 |
| secp224k1 | 224 | 0 | 3 | 224 | 223 | 225 |
| **GF(256)** | | | | | | |
| secp256k1 | 256 | 0 | 3 | 255 | 255 | 256 |
| prime256v1 | 256 | 256 | 255 | 255 | 255 | 256 |

## REFERENCES


[1] M. W. Barsagade, S. Meshram, "Overview of history of elliptic curves and its use in cryptography", *International Journal of Scientific & Engineering Research*, vol. 5, no. 4, pp. 467–471, 2014.

[2] H. W. Lenstra Jr., "Factoring integers with elliptic curves", *Annals of Mathematics*, pp. 649–673, vol. 126, no. 3, 1987. [Online]. Available: https://doi.org/10.2307/1971363

[3] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation*, pp. 203–209, vol. 48, no. 177, 1987. [Online]. Available: https://doi.org/10.1090/S0025-5718-1987-0866109-5

[4] V. Miller, "Uses of elliptic curves in cryptography", in *Advances in Cryptology (CRYPTO 1985)*, pp. 417–426, 1986. [Online]. Available: https://doi.org/10.1007/3-540-39799-X_31

[5] *SEC 1: Elliptic Curve Cryptography (version 2.0),* SECG - Standards for Efficient Cryptography (Certicom Research), May 2009.

[6] *SEC 2: Recommended Elliptic Curve Domain Parameters (version 2.0),* SECG - Standards for Efficient Cryptography (Certicom Research), January 2010.

[7] *SEC 4: Elliptic Curve Qu-Vanstone Implicit Certificate (version 1.0),* SECG - Standards for Efficient Cryptography (Certicom Research), January 2013.

[8] E. Barker, W. Barker, W. Burr, W. Polk, M. Smid, "Recommendation for Key Management – Part 1: General (Revision 3)", *Computer Security NIST Special Publication 800-57*, 2012.

[9] *FIPS PUB 186-2: Digital Signature Standard (DSS)*, U.S. Department of Commerce (National Institute of Standards and Technology). FIPS standard, 2000.

[10] *ANSI X9.62: Public Key Cryptography for the Financial Services Industry - Elliptic Curve Digital Signature Algorithm (ECDSA)*, X9 Standard, 1999.

[11] *ANSI X9.63: Public Key Cryptography for the Financial Services Industry: Elliptic Curve Key Agreement and Key Transport Protocols, working draft*, X9 Standard, October 2000.

[12] *IEEE 1363-2000: Standard Specifications for Public Key Cryptography*. IEEE Standard, 2000.

[13] *WAP-199-WTLS: Wireless Application Protocol,* Wireless Transport Layer Security Specification Standard, February 2000.

[14] The Legion of Bouncy Castle. Java and C# Crypto Libraries, Collection of APIs used in cryptography, 2013.

[15] Cyber Defense Laboratory. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks, 2011.

[16] W. Dai. Crypto++™ Library: a Free C++ Class Library of Cryptographic Schemes, 2016. [Online]. Available: http://cryptopp.com

[17] OpenSSL. OpenSSL Library: a Project of Open Source SSL and TLS protocols implementation, 2016.

[18] Theoretical Computer Science Research Group. Flexiprovider: a powerful toolkit for the Java Cryptography Architecture, 2012.

[19] Y. Hitchcock, E. Dawson, A. Clark, P. Montague, "Implementing an efficient elliptic curve cryptosystem over GF(p) on a smart card", *ANYIAM Journal*, pp. 354–377, vol. 44, 2003. [Online]. Available: https://doi.org/10.21914/anziamj.v44i0.686

[20] K. Lauter, "The advantages of elliptic curve cryptography for wireless security", *IEEE Wireless Communications*, pp. 62–67, vol. 11, no. 1, 2004. [Online]. Available: https://doi.org/10.1109/MWC.2004.

1269719
[21] M. Bafandehkar, S. M. Yasin, R. Mahmod, Z. M. Hanapi, "Comparison of ECC and RSA algorithm in resource constrained devices", in *IEEE Int. Conf. IT Convergence and Security,* 2013. [Online]. Available: http://dx.doi.org/10.1109/ICITCS.2013.6717816
[22] A. Botta, W. Donato, V. Persico, A. Pescape, "Intergration of cloud computing and internet of things: a survey", *Future Generation Computer Systems*, pp. 684–700, vol. 56, 2016. [Online]. Available: https://doi.org/10.1016/j.future.2015.09.021
[23] H. Arasteh *et al*. "IoT-based Smart Cities: a Survey", in *16th IEEE Int. Conf. Environment and Electrical Engineering (EEEIC)*, 2016. [Online]. Available: http://dx.doi.org/10.1109/EEEIC.2016.7555867
[24] S. H. Shah, I. Yaqoob, "A Survey: Internet of Things (IOT) technologies, applications and challenges", in *4th IEEE Int. Conf. Smart Energy Grid Engineering (SEGE)*, 2016. [Online]. Available: http://dx.doi.org/10.1109/SEGE.2016.7589556
[25] H. Hayouni, M. Hamdi, T. Kim, "A survey on encryption schemes in wireless sensor networks", in *7th IEEE Int. Conf. Advanced Software Engineering & Its Applications*, 2014. [Online]. Available: http://dx.doi.org/10.1109/ASEA.2014.14
[26] R. K. Kodali, S. Karanam, K. Patel, "Fast elliptic curve point multiplication for WSNs", in *IEEE TENCON Spring Conf.*, pp. 194–198, 2013. [Online]. Available: http://dx.doi.org/10.1109/TENCONSpring.2013.6584439
[27] S. Liu, G. Qi, X. A. Wang, "Fast and secure elliptic curve scalar multiplication algorithm based on a kind of deformed Fibonacci-type series", in *10th IEEE Int. Conf. P2P. Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2015, pp. 398–402. [Online]. Available: http://dx.doi.org/10.1109/3PGCIC.2015.21
[28] M. Mohammdi, A. S. Molahosseini, "Efficient design of Elliptic curve point multiplication based on fast Montgomery modular multiplication", in *3th IEEE Int. eConf. Computer and Knowledge Engineering (ICCKE)*, 2013. [Online]. Available: http://dx.doi.org/10.1109/ICCKE.2013.6682865
[29] D. J. Bernstein, "Curve25519: New Diffie-Hellman Speed Records", in *Public Key Cryptography (PKC 2006), Lecture Notes in Computer Science,* vol. 3958, 2006. [Online]. Available: http://dx.doi.org/10.1007/11745853_14
[30] Z. Liu, J. Groszschaedl, Z. Hu, K. Jarvinen, H. Wang, I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things", *IEEE Trans. Computers*, vol. 99, pp. 1–14, 2016. [Online]. Available: http://dx.doi.org/10.1109/TC.2016.2623609
[31] K. Liao, X. Cui, N. Liao, T. Wang, X. Zhang, Y. Huang, D. Yu, "High-speed constant-time division module for Elliptic Curve Cryptography based on GF($2^m$)", in *IEEE Int. Symposium on Circuits and Systems (ISCAS 2014)*, 2014, pp. 818–821. [Online]. Available: http://dx.doi.org/10.1109/ISCAS.2014.6865261
[32] R. Fujdiak, P. Masek, J. Hosek, P. Mlynek, J. Misurec, "Efficiency evaluation of different types of cryptography curves on low-power devices", in *7th IEEE Int. Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2015, pp. 269–274. [Online]. Available: http://dx.doi.org/10.1109/ICUMT.2015.7382441
[33] R. Fujdiak, J. Misurec, P. Mlynek, L. Janer, "Cryptograph key distribution with elliptic curve Diffie-Hellman algorithm in low-power devices for power grids", *Revue Roumaine des Sciences Techniques*, pp. 84–88, vol. 61, no. 1, 2016.
[34] P. Mlynek, J. Misurec, M. Koutny, P. Silhavy, "Two-port network transfer function for power line topology modeling", *Radioengineering*, vol. 21, no. 1, pp. 356–363, 2012.
[35] Texas Instruments. Mixed Signal Microcontroller: MSP430F5438A-EP. Technical documentation (SLAS967A), January 2014 (Revised January 2014).
[36] RS Components, Raspberry Pi 2B: Model B. Technical documentation for Raspberry Pi 2B (832-6274). [Online]. Available: http://uk.rs-online.com/webdocs/1392/0900766b8139232d.pdf
[37] R. Fujdiak, J. Misurec, P. Mlynek, O. Raso, Library for using elliptic curves in low-power devices. Software (version 1.60), 2015. [Online]. Available: http://www.utko.feec.vutbr.cz/~fujdiak/EC/cz.html
[38] OpenSSL, User Guide for the OpenSSL FIPS Object Module v2.0. Special Documentation, May 2016 [Online]. Available: https://www.openssl.org/docs/fips/UserGuide-2.0.pdf
[39] OpenSSL, Cryptography and SSL/TLS Toolkit, Software (version Openssl-1.1.0c), November 2016. [Online]. Available: https://www.openssl.org/source/